# Applications of Gaussian Naïve Bayes in Protein Secondary Prediction

**Rahul Alapati**[1, *, +], **Vineet Nayak**[1, +], **Anuj Gupta**[1, +], **Sritika Chakladar**[1, +] **and Austin Ream**[2, +]

[1]Graduate Student, Computer Science and Software Engineering, Auburn University, 36830, USA [2] Undergraduate Student, Computer Science and Software Engineering, Auburn University, 36830, USA

[*] rza0037@auburn.edu

[+]these authors contributed equally to this work

## ABSTRACT

Although the knowledge of the secondary structure alone is not as useful as a full three-dimensional model, secondary structure predictions provide important constraints for fold-recognition techniques as well as homology modelling, ab initio and constraint-based tertiary structure prediction methods. Therefore, it is vital to accurately predict the secondary structure of a protein. Naïve Bayes is a simple technique for constructing classifiers, that can be used to assign class labels to problem instances. This report lays out the implementation details of a Naïve Bayes classifier namely: Gaussian Naïve Bayes. The Gaussian Naïve Bayes classifier can be used to predict protein secondary structure based on the position specific scoring matrices generated by PSI-BLAST. Q3 Accuracy is calculated to determine the accuracy of the Gaussian Naïve Bayes classifier.

## 1. Introduction

Protein secondary structure is the three-dimensional form of local segments of proteins. The two most common secondary structure elements are alpha helices and beta sheets. Secondary structure elements typically form as an intermediate before the protein folds into its three-dimensional tertiary structure.

Secondary structure is formally defined by the pattern of hydrogen bonds between the amino hydrogen and the carboxyl oxygen atoms in the peptide backbone.

The Dictionary of Protein Secondary Structure (DSSP), is commonly used to describe the protein secondary structure with single letter codes. The secondary structure is assigned based on hydrogen bonding patterns. There are eight types of secondary structures that DSSP defines, but we consider only three of them

namely:

- H = 4-turn helix (alpha helix). Minimum length 4 residues.
- E = extended strand in parallel and/or anti-parallel beta-sheet conformation. Minimum length 2 residues.
- C = coil (residues which are not in any of the above conformations)

Predicting protein tertiary structure from only its amino acid sequence is a very challenging problem but using the secondary structure definitions is more tractable. Not only can successful secondary structure predictions provide a starting point for direct tertiary structure modelling, but they can also significantly improve sequence analysis and sequence-structure threading for aiding in structure and function determination.

Naïve Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes theorem with strong independence assumptions between the features.

Naïve Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values. All naïve Bayes classifiers assume that the value of a feature is independent of the value of any other feature, given the class variable.

It is one of the most efficient and effective inductive learning algorithms for machine learning and data mining. Its competitive performance in classification is surprising, because of the conditional independence assumption on which it is based. It is based on Bayes theorem and it predicts conditional probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class.

In this project, we explore the probabilistic nature for protein secondary structure using Naïve Bayes algorithm. Our approach models the Gaussian Naïve Bayes learning algorithm to train and model a classifier that predicts the 3 class labels for secondary structure. We use this model to predict the class labels on any given test dataset by selecting the most probable class assignment.

## 2. Methods

The protein sequences in Multi-FASTA format, the true 3-class secondary structure labels (‘H’, ‘E‘, ‘C’) of the proteins and the position specific scoring matrices have been used to train and test the gaussian naïve Bayes classifier.

The input protein sequences and their respective SS classes have been curated into non-overlapping sets of Training (75%) and Test (25%) datasets using the simple random sampling without replacement.

Now, PSIBLAST and the nr database have been used to generate train and test PSSMs for protein sequences in train and test datasets, respectively.

BLAST (Basic Local Alignment Search Tool) is a sequence similarity search method, in which a query protein is compared to protein sequences in a target database to identify regions of local alignment and report those alignments that score above a given score threshold. Position-Specific Iterative (PSI)-BLAST is a protein sequence profile search method that builds off the alignments generated by a run of the BLASTp program.

A sample PSSM for a sequence is shown below in figure 1:

>sequence
TIKVLFVDDHEMVRIGISSYLSTQSDIEVVGEGASGKEAIAKAHELKPDLILMDLLMEDMDGVEATT
QIKKDLPQIKVLMLTSFIEDKEVYRALDAGVDSYILKTTSAKDIADAVRKTSRGESVFEPEVLVKMR
NRMKKRAELYEMLTEREMEILLLIAKGYSNQEIASASHITIKTVKTHVSNILSKLEVQDRTQAVIYAF
QHNLIQ

| | | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | T | -1 | -3 | -2 | -3 | -3 | -2 | -2 | -3 | -3 | -3 | -3 | -2 | -3 | -4 | -3 | 3 | 6 | -4 | -3 | -2 |
| 2 | I | -3 | -5 | -6 | -6 | -3 | -5 | -5 | -6 | -6 | 7 | 0 | -5 | 2 | -2 | -5 | -5 | -2 | -5 | -4 | 2 |
| 3 | K | -3 | 7 | -2 | -3 | -5 | 0 | -2 | -4 | -3 | -5 | -5 | 4 | -4 | -5 | -4 | -1 | 1 | -5 | -4 | -4 |
| 4 | V | -2 | -5 | -5 | -6 | -3 | -5 | -5 | -6 | -6 | 3 | 1 | -5 | -1 | -3 | -5 | -4 | -2 | -5 | -4 | 6 |
| 5 | L | -2 | -5 | -6 | -6 | -3 | -4 | -5 | -6 | -5 | 1 | 5 | -5 | 3 | -1 | -5 | -4 | -3 | -4 | -3 | 2 |
| 6 | F | -3 | -5 | -6 | -6 | -3 | -5 | -5 | -6 | -5 | 5 | 4 | -5 | -1 | 1 | -5 | -5 | -3 | -4 | -3 | 3 |
| 7 | V | 4 | -4 | -5 | -5 | 1 | -4 | -4 | -4 | -5 | 0 | -2 | -4 | -2 | -4 | -4 | -3 | -2 | -5 | -4 | 6 |
| 8 | D | -4 | -4 | -1 | 8 | -6 | -2 | 0 | -4 | -3 | -6 | -6 | -3 | -6 | -6 | -4 | -2 | -3 | -7 | -6 | -6 |
| 9 | D | -4 | -4 | -1 | 8 | -6 | -3 | -1 | -4 | -3 | -6 | -6 | -3 | -6 | -6 | -4 | -2 | -3 | -7 | -6 | -6 |
| 10 | H | -3 | -2 | -2 | -3 | -5 | 6 | -1 | -4 | 9 | -5 | -5 | -2 | -3 | -4 | -4 | -3 | -3 | -5 | -1 | -5 |
| 11 | E | 3 | -1 | -2 | 0 | -4 | 1 | 4 | -2 | 3 | -4 | -3 | -2 | 0 | -4 | 2 | 0 | -1 | -5 | -4 | -2 |
| 12 | M | -3 | -4 | -5 | -6 | -4 | -4 | -5 | -6 | -5 | 2 | 4 | -4 | 7 | -2 | -5 | -4 | -3 | -4 | -3 | 3 |
| 13 | V | -2 | -5 | -5 | -6 | -3 | -5 | -5 | -6 | -5 | 2 | 1 | -5 | 1 | 0 | -5 | -4 | -3 | -5 | -3 | 6 |
| 14 | R | -4 | 8 | -3 | -4 | -6 | -1 | -2 | -5 | -3 | -5 | -3 | 0 | -4 | -5 | -5 | -3 | -3 | -5 | -4 | -5 |
| 15 | I | 1 | 2 | -3 | -1 | -4 | 3 | 0 | -2 | -3 | 1 | 0 | 1 | 3 | -4 | -4 | -1 | 1 | -5 | -4 | -2 |
| 16 | G | -1 | -5 | -3 | -4 | -5 | -4 | -4 | 7 | -4 | -6 | -6 | -4 | -5 | -6 | -4 | -2 | -4 | -5 | -5 | -6 |
| 17 | I | -4 | -5 | -6 | -6 | -4 | -5 | -5 | -6 | -5 | 1 | 4 | -5 | 0 | 6 | -6 | -5 | -3 | -3 | -1 | 1 |
| 18 | S | 1 | 5 | -2 | -4 | -3 | 0 | -1 | -1 | -3 | -2 | -3 | 2 | -2 | -5 | -4 | 2 | -1 | -5 | -4 | 0 |
| 19 | S | 3 | -4 | -3 | -3 | -3 | -2 | -3 | -1 | -4 | -1 | 0 | -3 | 5 | 1 | -4 | 2 | 1 | -4 | -1 | 0 |

**Figure 1. Sample PSSM for a sequence.**

The PSSM contains 20 values for each residue in a protein sequence. For a protein sequence of N residues, there will be N * 20 PSSM Values.

Now, for feature generation a sliding window of 5 around central residue is used, as shown in Figure 2.

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | T | -1 | -3 | -2 | -3 | -3 | -2 | -2 | -3 | -3 | -3 | -3 | -2 | -3 | -4 | -3 | 3 | 6 | -4 | -3 | -2 |
| 2 | I | -3 | -5 | -6 | -6 | -3 | -5 | -5 | -6 | -6 | 7 | 0 | -5 | 2 | -2 | -5 | -5 | -2 | -5 | -4 | 2 |
| 3 | K | -3 | 7 | -2 | -3 | -5 | 0 | -2 | -4 | -3 | -5 | -5 | 4 | -4 | -5 | -4 | -1 | 1 | -5 | -4 | -4 |
| 4 | V | -2 | -5 | -5 | -6 | -3 | -5 | -5 | -6 | -6 | 3 | 1 | -5 | -1 | -3 | -5 | -4 | -2 | -5 | -4 | 6 |
| 5 | L | -2 | -5 | -6 | -6 | -3 | -4 | -5 | -6 | -5 | 1 | 5 | -5 | 3 | -1 | -5 | -4 | -3 | -4 | -3 | 2 |
| 6 | F | -3 | -5 | -6 | -6 | -3 | -5 | -5 | -6 | -5 | 5 | 4 | -5 | -1 | 1 | -5 | -5 | -3 | -4 | -3 | 3 |
| 7 | V | 4 | -4 | -5 | -5 | 1 | -4 | -4 | -4 | -5 | 0 | -2 | -4 | -2 | -4 | -4 | -3 | -2 | -5 | -4 | 6 |
| 8 | D | -4 | -4 | -1 | 8 | -6 | -2 | 0 | -4 | -3 | -6 | -6 | -3 | -6 | -6 | -4 | -2 | -3 | -7 | -6 | -6 |
| 9 | D | -4 | -4 | -1 | 8 | -6 | -3 | -1 | -4 | -3 | -6 | -6 | -3 | -6 | -6 | -4 | -2 | -3 | -7 | -6 | -6 |
| 10 | H | -3 | -2 | -2 | -3 | -5 | 6 | -1 | -4 | 9 | -5 | -5 | -2 | -3 | -4 | -4 | -3 | -3 | -5 | -1 | -5 |

**Figure 2. Sliding window of 5 around the central residue 'L'.**

For the residue L, we consider the PSSM values of K & V above it and the PSSM values of F & V below it to generate a feature vector. Therefore, there will be 20 * 5 = 100 PSSM values for each non-terminal residue like L. For terminal residues like T that do not have one or more neighbors on either side, rows containing 20 values of -1 are used for feature generation.

The N * 20 PSSM values are now converted into N * 100 Feature vectors, as shown below in Figure 3.

20 PSSM values for residue 'L':

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | L | -2 | -5 | -6 | -6 | -3 | -4 | -5 | -6 | -5 | 1 | 5 | -5 | 3 | -1 | -5 | -4 | -3 | -4 | -3 | 2 |

100 Feature vector for residue 'L':

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | L | -2 | -5 | -6 | -6 | -3 | -4 | -5 | -6 | -5 | 1 | 5 | -5 | 3 | -1 | -5 | -4 | -3 | -4 | -3 | 2 |
| | | -2 | -5 | -5 | -6 | -3 | -5 | -5 | -6 | -6 | 3 | 1 | -5 | -1 | -3 | -5 | -4 | -2 | -5 | -4 | 6 |
| | | -3 | 7 | -2 | -3 | -5 | 0 | -2 | -4 | -3 | -5 | -5 | 4 | -4 | -5 | -4 | -1 | 1 | -5 | -4 | -4 |
| | | -3 | -5 | -6 | -6 | -3 | -5 | -5 | -6 | -5 | 5 | 4 | -5 | -1 | 1 | -5 | -5 | -3 | -4 | -3 | 3 |
| | | 4 | -4 | -5 | -5 | 1 | -4 | -4 | -4 | -5 | 0 | -2 | -4 | -2 | -4 | -4 | -3 | -2 | -5 | -4 | 6 |

**Figure 3. Feature generation from N * 20 PSSM Values.**

Now, the central residue's true Secondary Structure types from .ss have been used for class label assignment as H, E or C.

The class label for the residue 'L' is determined as 'E' from the .ss file.

### 2.1. Gaussian Naïve Bayes Learning on Training Set:

The fundamental Naïve Bayes assumption made here is that each feature makes an independent and equal contribution to the outcome. The assumptions made by Naïve Bayes are not generally correct in real world situations, but often works well in practice.

Now, we trained a classifier model on the class conditional means, variances and SS class priors of the training dataset, assuming each P (Xi | Y = yk) (probability of feature Xi given a class yk) to follow Gaussian Distribution.

The class conditional means and variances are calculated for all the values in the 100 features based on the true SS Class (H, E or C) of the residue.

The SS class priors are calculated as the probabilities of H, E and C in the training dataset.

A sample model with class conditional means, variances and priors is shown below in the figure 4:

| Class | Means | | | | | Variances | | | | | Priors |
|---|---|---|---|---|---|---|---|---|---|---|---|
| H | -0.5988 | -1.037 | -2.179 | -2.265 | ...100 Means | 7.59827 | 11.394 | 7.0605 | 8.7505 | ...100 Variances | 0.379836 |
| E | -2.381 | -2.56 | -2.619 | -3.417 | ...100 Means | 4.188209 | 10.175 | 10.188 | 10.362 | ...100 Variances | 0.231913 |
| C | -1.4228 | -1.577 | -0.8211 | -0.854 | ...100 Means | 4.081433 | 6.7969 | 8.2119 | 13.442 | ...100 Variances | 0.388252 |

**Figure 4. Gaussian Model with means, variances and priors.**

The priors of H, E and C add up to 1. (0.379836 + 0.231913 + 0.388252)

### 2.2. Gaussian Naïve Bayes Learning on Test Set:

For classification, the probability of given set of inputs for all possible values of the class y are determined. The class with the highest probability is assigned as a label to the residue. This is also known as Maximum A Posteriori (MAP).

The MAP can be expressed mathematically as follows:

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

where P(y) is the class probability or the prior and P(xi | y) is the conditional probability.

The P(xi | y) can be calculated using the following equation:

$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

where $\mu_k$ is the mean of the values in x associated with class Ck and $\sigma_k^2$ the variance of the values in x associated with class Ck.

Using MAP, the priors P(y) learned from the training dataset are multiplied with the class conditional probabilities P(xi | y) and then the most probable class is assigned as a label to the amino acid residue.

**Q3 Accuracy:**

Now, Q3 or the three-state accuracy of secondary structure of proteins is calculated to evaluate the classification performance on the test dataset.

The Q3 accuracy is defined as the “number of correctly predicted secondary confirmations of amino acid residues” upon “the total number of residues belonging to that particular confirmation (H, E or C)”.

## 3. Results

The following is a sample output showing the predicted class labels for a single FASTA sequence:

FASTA Sequence:

```
>sequence
AMAPPFFDLKPVSVDLALGESGTFKCHVTGTAPIKITWAKDNREIRPGGNYKMTLVENTATLTVLKVTKGDAGQYTCYASNVA
GKDSCSAQLGVQEPPRFIKKLEPSRIVKQDEHTRYECKIGGSPEIKVLWYKDETEIQESSKFRMSFVESVAVLEMYNLSVEDS
GDYTCEAHNAAGSASSSTSLKVKEPPVFRKKPHPVETLKGADVHLECELQGTPPFQVSWHKDKRELRSGKKYKIMSENFLTSI
HILNVDSADIGEYQCKASNDVGSYTCVGSITLKA
```

The output will be generated as follows:

```
>sequence
HHCCCHHHHCCCHHHHHCCCHHHECEHECCCCCHHEEEEECCCHCCCCCCHHHHCCCCCCCEEHHHHHCCCCHEEEEHECHHCC
EEEHECHHCCCHCCCHHEHHHCCCHHHCCCCHHHHHCCCHHHECHHHCCCCCHHEEEEECCCHHCCCCHHEHCCCCCCHHEHHH
```

The following is the Q3 Accuracy of our Gaussian Naïve Bayes Classifier:

| Class | Q3 Accuracy |
|---|---|
| Total | 0.637666643 |
| H | 0.585636225 |
| E | 0.610674837 |
| C | 0.704300106 |

We achieved a total accuracy of 63.7 %, accuracy of 58.5 % in case of helices (H), accuracy of 61.06 % in case of beta sheets (E) and accuracy of 70.4 % in case of coils (C).

**Experiment 1**: While calculating the MAP estimates, the priors or class probabilities for H, E and C were not considered. The following is the Q3 Accuracy of our Gaussian Naïve Bayes Classifier when priors are not considered:

| Class | Q3 Accuracy |
|---|---|
| Total | 0.633182704 |
| H | 0.567018064 |
| E | 0.639087318 |
| C | 0.692933966 |

The total accuracy is reduced to 63.3 % from 63.7 %, to 56.7 % from 58.5 % in case of helices (H) and to 69.2 % from 70.4 % in case of coils (C).

The accuracy is increased to 63.9 % from 61.06 % in case of beta sheets (E).

**Experiment 2:** In feature generation, only the 20 PSSM values for each residue are considered and the concept of sliding window is not used in this case. The following is the Q3 Accuracy of our Gaussian Naïve Bayes Classifier when sliding window is not used:

| Class | Q3 Accuracy |
|---|---|
| Total | 0.497530922 |
| H | 0.344195668 |
| E | 0.49107474 |
| C | 0.648510885 |

The total accuracy is reduced to 49.75 % from 63.7 %, to 34.4 % from 58.5 % in case of helices (H), to 49.1 % from 61.06 % in case of beta sheets (E) and to 64.85 % from 70.4 % in case of coils (C).

**Experiment 3:** In feature generation, we use a sliding window of 3 instead of 5. The following is the Q3 Accuracy of our Gaussian Naïve Bayes Classifier when sliding window of 3 is used:

| Class | Q3 Accuracy |
|---|---|
| Total | 0.604105037 |
| H | 0.523304044 |
| E | 0.584971378 |
| C | 0.693434677 |

The total accuracy is reduced to 60.41 % from 63.7 %, to 52.33 % from 58.5 % in case of helices (H), to 58.49 % from 61.06 % in case of beta sheets (E) and to 69.34 % from 70.4 % in case of coils (C).

## 4. Discussion

We performed different experiments as discussed above to improve the Q3 accuracy of our Gaussian Naïve Bayes Classifier. Our accuracy was the lowest (49.75 %) when the sliding window was not used for feature generation. We achieved the highest accuracy of 63.7 % when we used a sliding window of 5 and used MAP estimates to determine the class assignment. These experiments outlay the importance of the knowledge about neighbors and the priors or class probabilities in Secondary Structure classification.

Also, by observing the accuracy of 63.7 % we can conclude that, the assumptions (i.e. features are independent of each other given a class) made while building Gaussian Naïve Bayes classifier works well in practice, though they are not generally correct in real world situations.

## 5. References


1. Medha Bhagwat and L. Aravind. Chapter 10 PSI-BLAST Tutorial, Comparative Genomics: Volumes 1 and 2.
2. Arun Kumar Chinnasamy, Wing-Kin Sung, and Ankush Mittal. Protein Structure and Fold Prediction using Tree-Augmented Naive Bayesian Classifier. J. Bioinform. Comput. Biol. 03, 803 (2005).

3. Ross D. King  Michael J.E. Sternberg. Identification and application of the concepts important for accurate and reliable protein secondary structure prediction. Volume 5, Issue 11, November 1996 Proteins.
4. David T Jones. Protein secondary structure prediction based on position-specific scoring matrices. Volume 292, Issue 2, 17 September 1999, Pages 195-202 JMB.
5. Robles, V., Larrañaga, P., Peña, J. M., Menasalvas, E., Pérez, M. S., Herves, V., & Wasilewska, A. (2004). Bayesian network multi-classifiers for protein secondary structure prediction. Artificial Intelligence in Medicine, 31(2), 117-136.
6. Zhang, H. (2004). The optimality of naive Bayes. AA, 1(2), 3.
7. https://en.wikipedia.org/