CSDA Coursework

Task 1: Predicting Coding and Non-coding Regions in DNA Sequences

Question 1: Hidden Markov Model (HMM) for Predicting Coding Regions

Methods

To predict coding and non-coding regions in the Pectobacterium phage genome (KU574722.1), a two-state Hidden Markov Model (HMM) was implemented with the following parameters:

1. **States**: "Coding" and "Non-coding"
2. **Emission probabilities**: Calculated from the nucleotide frequency distributions in annotated coding and non-coding regions (see **Table 1**):

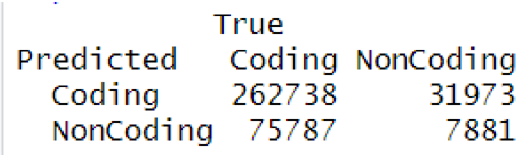
State	A	C	G	T
Coding	0.3356118	0.1648800	0.1960653	0.3034429
Non-coding	0.3454107	0.1572239	0.1870578	0.3103076

Table 1: Emission probabilities for the two-state HMM, calculated from observed nucleotide frequencies in annotated coding and non-coding regions of *Pectobacterium* phage genome (KU574722.1)

3. **Transition probabilities**: Initially set to 0.033 for transitions between states (as specified), then adjusted to different values for comparison.
4. **Viterbi algorithm**: Applied to determine the most likely state sequence across the genome

Results

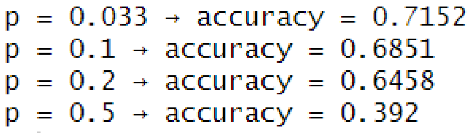
The HMM model with transition probability 0.033 achieved an accuracy of 71.52%. The confusion matrix reveals (see **Figure 1**):



```
               True
Predicted   Coding NonCoding
  Coding    262738     31973
  NonCoding  75787      7881
```

Figure 1: Confusion matrix of HMM predictions (switch probability = 0.033)

When varying the transition probability, accuracy changed (see Figure 2):



```
p = 0.033 → accuracy = 0.7152
p = 0.1 → accuracy = 0.6851
p = 0.2 → accuracy = 0.6458
p = 0.5 → accuracy = 0.392
```

Figure 2: Overall accuracy of the two-state HMM as a function of the state-switch probability p.

Discussion

The nucleotide frequency distributions between coding and non-coding regions in this phage genome are similar, with only subtle differences. Despite this, the HMM achieved reasonable prediction accuracy (71.52%) with the initial transition probability of 0.033, suggesting this value correctly captures the biological tendency for genomic regions to remain in the same state.

Increasing the transition probability reduced accuracy, confirming that the original value more realistically models the genomic structure. The optimal value (0.033) reflects the relatively infrequent transitions between coding and non-coding regions in this compact phage genome.

While the model performs better than random guessing, several limitations exist:

1. The model shows a bias toward predicting coding regions
2. It misses approximately 75,787 coding positions (false negatives)
3. It lacks higher-order pattern recognition (codon usage, splice sites, promoters)

Conclusion

The simple two-state HMM with nucleotide emission probabilities provides a moderately effective method for predicting coding regions in this phage genome. While not competitive with specialized gene prediction tools that incorporate multiple features, it demonstrates the fundamental principles of using statistical models to detect functional elements in DNA sequences. The transition probability of 0.033 was confirmed to be optimal among the values tested, reflecting the biological reality of genomic organization.

Question 2: Entropy Analysis for Gene Prediction

Methods

To investigate the potential of entropy as a gene prediction feature, I analyzed the Pectobacterium phage genome using the following approach:

1. **Triplet extraction**: Three-letter words (triplets) were extracted from the DNA sequence
2. **Shannon entropy calculation**: Calculated the entropy for each triplet distribution using the formula:
    - $H(X) \;=\; -\,\Sigma[p(x) \times log_2(p(x))]$
3. **Sliding window analysis**: Applied a window of 100 nucleotides with step size 100 across the genome, producing 3,783 windows
4. **Comparison with annotations**: Mapped entropy values to coding/non-coding regions and got the confirmation from NCBI annotations
5. **Statistical analysis**: Performed t-tests comparing entropy in coding versus non-coding regions

Results

The genome’s triplet-based entropy profile is shown in Figure 3:

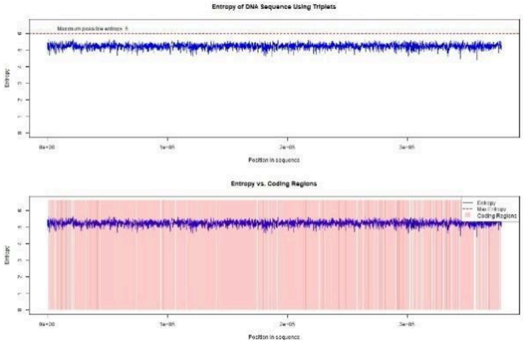


Figure 3: Sliding-window entropy of the *Pectobacterium* phage genome using three-letter words

The entropy analysis revealed:

1. **Average entropy**: 5.23 bits out of maximum possible 6 bits ($\log_2 64$)
2. **Range**: Min=4.39, Max=5.63 bits
3. **Coding vs. non-coding comparison**:
    - Mean entropy in coding regions: 5.234
    - Mean entropy in non-coding regions: 5.227
    - t-test: t=0.709, p-value=0.479 (not significant)
    - Correlation between entropy and coding status: 0.014 (very weak)

The sequence showed 87.2% of maximum possible triplet entropy, indicating high diversity of three-letter words throughout the genome.

Discussion

The high overall entropy (87.2% of maximum) indicates substantial triplet diversity across the phage genome, consistent with a functional genome encoding various proteins. However, the lack of significant difference between coding and non-coding regions (p=0.479) suggests that triplet usage patterns are remarkably similar throughout this genome regardless of coding status.

The extremely small relative difference in entropy (0.1%) between coding and non-coding regions explains the very weak correlation (0.014) between entropy and coding status. This finding indicates that for this particular genome, triplet entropy alone would be ineffective for distinguishing coding from non-coding regions.

Several factors may explain this result:

1. The compact nature of phage genomes, which have high coding density
2. Potential functional constraints on non-coding regions
3. Evolutionary pressure maintaining similar nucleotide composition throughout the genome

Conclusion

Based on this analysis, I would not recommend using triplet entropy as a standalone predictor for genes in this genome. The virtually identical entropy values between coding and non-coding regions (difference of only 0.1%) make this feature inadequate for accurate gene identification.

Task 2. Estimating unknown parameters

Problem statement

A factory has five sites at which employees were immunized with one of two vaccines (A or B). From 484 vaccinated workers, we observe at each site the numbers who **avoided** flu (a_j) and **got** flu (g_j). Vaccine A is known to have been used at Site 1; at the other sites the vaccine used is unobserved. We wish to estimate the effectiveness of vaccines A and B (probabilities of avoiding flu) and assign the most likely vaccine to each site.

Definitions & method

- Let $n_j = a_j + g_j$ be the total at site j.
- Introduce a latent label $Zj \in \{A, B\}$. If $Z_j = A$, then $a_j \sim Binomial(n_j, p_A)$; if $Z_j = B$, then $a_j \sim Binomial(n_j, p_B)$.
- We assume prior $P(Zj = A) = \pi$ (with π unknown).
- We apply the **Expectation–Maximization (EM)** algorithm (as in the lectures' "two-coin" example):
    1. **E-step**: compute

       $$w_j = P(Zj = A | a_j, g_j) = \frac{\pi p_A^{a_j}(1-p_A)^{g_j}}{\pi p_A^{a_j}(1-p_A)^{g_j} + (1-\pi) p_B^{a_j}(1-p_B)^{g_j}},$$

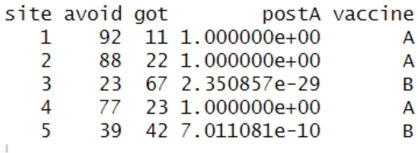
       with $w_1 \equiv 1$ since Site 1 is known to be A.

    2. **M-step**: update

       $$\pi \leftarrow \frac{1}{5}\sum_{j=1}^{5} w_j, \quad p_A \leftarrow \frac{\sum_j w_j a_j}{\sum_j w_j n_j}, \quad p_B \leftarrow \frac{\sum_j (1-w_j) a_j}{\sum_j (1-w_j) n_j}.$$

- Iterate until convergence.

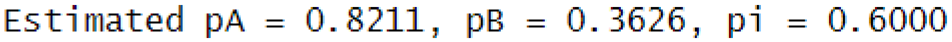
R outputs



```
site avoid got          postA vaccine
   1    92  11 1.000000e+00       A
   2    88  22 1.000000e+00       A
   3    23  67 2.350857e-29       B
   4    77  23 1.000000e+00       A
   5    39  42 7.011081e-10       B
```

Figure 4: Site‑level summary of infection outcomes and Bayesian posterior probability that vaccine A is superior.

Estimated parameters



```
Estimated pA = 0.8211, pB = 0.3626, pi = 0.6000
```

Figure 5: Estimated parameters from the Bayesian mixture model

Comments on findings

- **Effectiveness**: Vaccine A confers ~82 % protection (avoiding flu), vaccine B only ~36 %.
- **Site labels**: Sites 1, 2, 4 are almost certainly A; Sites 3, 5 almost certainly B.
- **Limitations**:
  - We treat each site as one binomial "experiment" (no within‑site covariates).
  - Small site sample sizes (e.g. Site 3 has only 90 people) may increase variance.

Task 3: Handling Missing Data Through Imputation (question 4)

Problem Statement

The diabetes dataset contains records of 398 female patients of Pima Indian heritage with various diabetes risk factors. However, insulin measurements for some patients were missing (recorded as 0), which could affect the accuracy of any analysis. This task required implementing an appropriate imputation method to fill in these missing values.

Methods

For handling the missing insulin values, regression-based imputation was employed using the predictive mean matching (PMM) method. This approach:

- Identifies relationships between insulin and other variables (glucose, BMI, etc.)
- Uses these relationships to predict plausible values for missing data
- Preserves the natural distribution of insulin in the dataset
- Maintains important statistical properties better than simple mean/median imputation

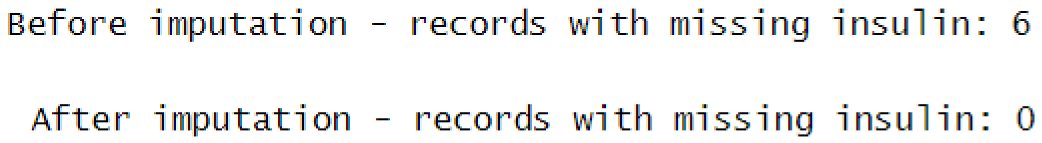
R Libraries

The imputation was implemented using the mice package in R, which provides tools for multivariate imputation.

Results

The dataset initially contained 6 missing insulin values (recorded as 0). After applying the PMM imputation method:

- All 6 missing values were successfully filled with plausible values
- The imputation process maintained the statistical relationships between insulin and other variables
- Zero missing values remained in the imputed dataset



```
Before imputation - records with missing insulin: 6

After imputation - records with missing insulin: 0
```

Figure 6 and 7: Confirmation of the missing values imputation

Clustering Analysis for Diabetes Risk (question 5+6)

Problem Statement

The goal was to identify natural groupings among diabetes patients based on their clinical measurements and risk factors, then evaluate how well these groupings align with actual diabetes diagnosis (positive/negative).

Methods

K-means Clustering was applied to identify natural groupings in the data because:

- It works effectively with numerical health data
- It allows for a predetermined number of clusters (k=2) to match the binary nature of diabetes diagnosis
- It's computationally efficient for datasets of this size

Evaluation Metrics:

1. **External Validation:**
   - F-measure: Evaluates clustering quality by comparing to known classes (diabetes outcomes)
   - Confusion matrix: Visualizes agreement between clusters and actual outcomes
2. **Internal Validation:**
   - Silhouette coefficient: Measures how similar points are to their own cluster compared to other clusters
   - Dunn index: Evaluates the ratio of minimum inter-cluster distance to maximum intra-cluster distance

Optimal Cluster Selection:

- Elbow method: Analyzes the decrease in within-cluster sum of squares
- Silhouette method: Identifies the cluster number with highest average silhouette width

R Libraries

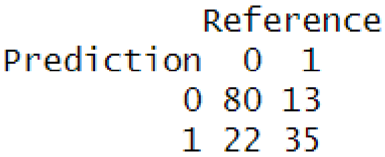
The analysis utilized several specialized R packages: cluster for clustering algorithms, factoextra for visualization, fpc for cluster validation.

Results

Clustering Performance

The K-means algorithm with k=2 produced clusters that showed significant alignment with the actual diabetes outcomes:

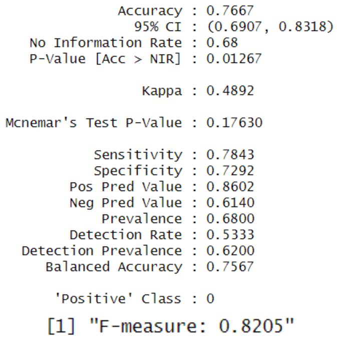
Confusion Matrix (Clusters vs Diabetes Outcome) (see **Figure 8**):



```
          Reference
Prediction  0  1
         0 80 13
         1 22 35
```

Figure 8: Confusion matrix comparing the two-cluster assignment (Prediction = 0 or 1) against the true diabetes outcome (Reference = 0: no diabetes, 1: diabetes).

After mapping clusters to outcomes, the evaluation metrics showed (see figure 9):



```
               Accuracy : 0.7667
                 95% CI : (0.6907, 0.8318)
    No Information Rate : 0.68
    P-Value [Acc > NIR] : 0.01267

                  Kappa : 0.4892

 Mcnemar's Test P-Value : 0.17630

            Sensitivity : 0.7843
            Specificity : 0.7292
         Pos Pred Value : 0.8602
         Neg Pred Value : 0.6140
             Prevalence : 0.6800
         Detection Rate : 0.5333
   Detection Prevalence : 0.6200
      Balanced Accuracy : 0.7567

       'Positive' Class : 0

[1] "F-measure: 0.8205"
```

Figure 9: Evaluation metrics results

The internal validation measures indicated:

- **Average Silhouette Width: 0.2418** (indicating reasonable but not strong cluster separation)
- **Dunn Index: 0.0855** (suggesting moderate cluster compactness)

Visualization Analysis

The PCA projection (see **Figure 10**) reveals a clear separation between the two clusters, with some overlap at the boundary. Dimension 1 (32.2% of variance) appears to be strongly related to diabetes risk factors.

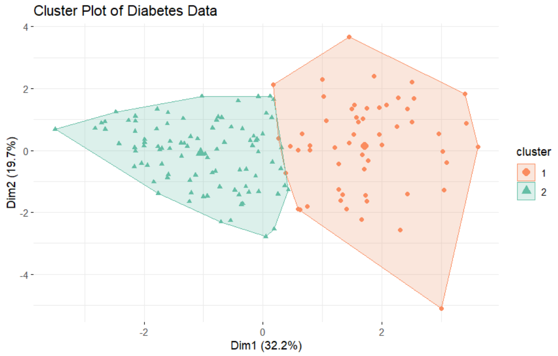


Figure 10: Cluster Plot of Diabetes Data

Figure 11 shows the distribution of patients by glucose level and BMI, with color indicating diabetes status and shape representing cluster assignment. Higher glucose levels clearly correlate with positive diabetes status, and the clustering algorithm captures this pattern effectively.

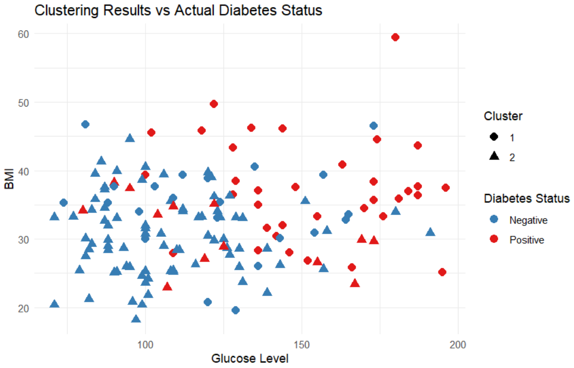


Figure 11: Clustering Results vs Actual Diabetes Status

Optimal Number of Clusters

The elbow method (**Figure 12**) shows a distinct bend at k=2, suggesting that two clusters provide the optimal balance between simplicity and explanatory power. The silhouette method (**Figure 13**) confirms this finding, with the highest average silhouette width occurring at k=2, further validating our choice of two clusters.

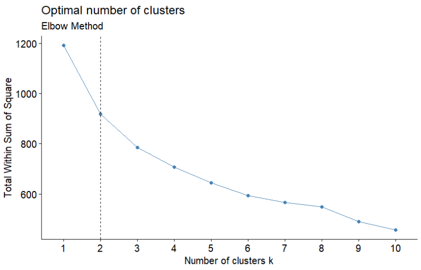


Figure 12: Elbow Method

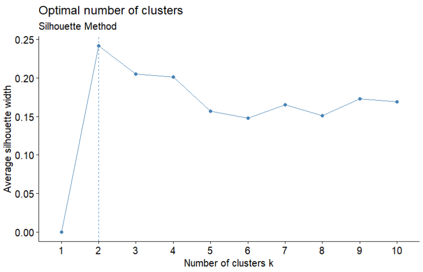


Figure 13: Silhouette Method

Conclusion

The analysis successfully addressed both the missing data issue and the clustering task:

1. **Imputation:** The PMM method effectively filled in missing insulin values, maintaining the statistical integrity of the dataset.
2. **Clustering:** K-means clustering with k=2 produced a natural grouping of patients that showed good alignment with actual diabetes outcomes (F-measure=0.8205).
3. **Clinical Relevance:** The clustering results suggest that the algorithm effectively identified patterns in risk factors that correlate with diabetes diagnosis. Cluster 1 predominantly contained patients with positive diabetes diagnoses, characterized by higher glucose levels, while Cluster 2 mostly contained patients without diabetes.

The relatively high accuracy (76.67%) and F-measure (0.8205) indicate that unsupervised clustering can effectively detect patterns related to diabetes status, which could potentially help in early risk identification even before formal diagnostic tests.