Communications in Information and Systems
Volume 0, Number 0, 1, 2025

Asymmetric Natural Vector Method for Predicting Ambiguous Nonstandard Base Codes

Guoqing Hu, Hao Wang, Stephen S.-T. Yau*

With the rapid development of genome sequencing technology, genomic sequence analysis has become an important field in modern biological research. However, sequencing errors, repetitive regions, and complex biological processes often lead to missing or ambiguous bases in genomic sequences, which are typically represented by non-standard symbols (such as R, Y, S, W, K, etc.). These issues severely affect the accuracy of genomic data, especially in tasks such as gene assembly and variant detection. To address this issue, this study proposes an encoding method based on asymmetric covariance natural vectors to characterize genomic sequences and predict ambiguous bases using Gated RecurrentUnit(GRU). Experimental results demonstrate that, compared with traditional encoding methods (such as One-hot encoding), the asymmetric covariance natural vector can more effectively utilize the information surrounding missing nucleotides for prediction, showing significant advantages in recovering nucleotides missing from intermediate positions. Additionally, this method also performs well on the SARS-CoV-2 Alpha variant dataset, with an error rate of only 1.09% in predicting non-standard bases during the encoding recovery process, further validating its effectiveness and potential for practical genomic data analysis.

Keywords and phrases: Gene Recovery, Natural Vector, GRU, SARS-CoV-2.

1. Introduction

In recent years, the rapid development of genome sequencing technology has facilitated the accumulation of large-scale genomic data, providing unprecedented opportunities for biological research [1, 2, 3]. However, genomic sequences often contain missing or ambiguous bases due to sequencing errors, sample contamination, the presence of repetitive regions, and complex biological processes. These bases are typically represented by IUPAC codes[4],

*Corresponding author.

which pose challenges to tasks such as gene assembly and variant detection, thereby affecting the accuracy of the data and the reliability of downstream bioinformatics analyses.

Figure 1 shows the S protein data of different SARS-CoV-2 strains downloaded from the GISAID database [5] as of October 2024. We calculated the missing rate by identifying sequences that contained at least one non-standard base (i.e., any character other than A, G, C, or T). A sequence was considered 'missing' if it contained at least one such character, and the reported missing rate represents the proportion of these sequences in the total sample.

The results show that the missing rate in all strains is nearly over 70%, with Beta and Omicron strains reaching up to 85%. This indicates that a large number of missing regions exist in the SARS-CoV-2 genome, which may affect the accuracy of downstream analyses. Therefore, how to effectively recover missing or ambiguous bases to improve the integrity of genomic data has become an important issue that needs to be addressed urgently in current research.

Currently, there are various methods for handling missing or ambiguous bases in genomic sequences. For example, IMGT/GENE-DB [6] can be used to identify the correct immunogenome and perform gene recovery. However, this method is computationally intensive and time-consuming, and the recovery

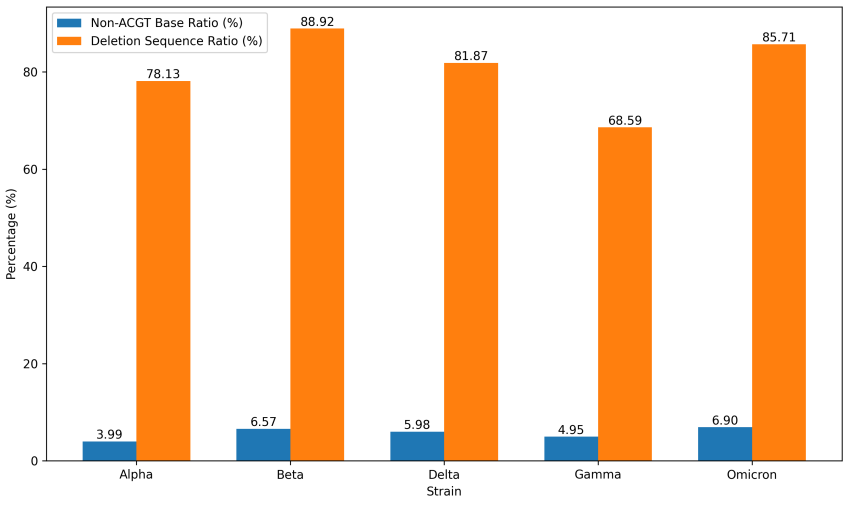

Figure 1: Comparison of Base Deletions in S protein Among Different SARS-CoV-2 Strains.

results may be uncertain. In recent years, deep learning has been widely applied in genomic sequence analysis[7, 8, 9]. For example, a common approach is to perform One-hot encoding of the nucleotides A, G, C, and T, and use deep learning models to predict missing bases. However, deep learning models typically require a fixed input dimension, which means that truncation or padding is needed when processing genomic sequences. This may lead to information loss or the introduction of noise[10, 11], thereby affecting prediction accuracy.

Therefore, a new encoding method is needed to reduce information loss and improve prediction accuracy. In recent years, Alignment-free methods have gained increasing attention. These methods map genomic sequences of arbitrary length to fixed-dimension vectors based on the statistical features of nucleotides, thereby reducing information loss to some extent. For example, the Natural Vector method converts genomic sequences into fixed-dimension vectors by extracting their statistical features [12]. The Covariance Natural Vector further incorporates the covariance information between nucleotides, thereby enhancing the representation of sequence features [13]. However, these methods primarily focus on the statistical relationships between nucleotides and fail to fully consider the directionality information of nucleotides. Directionality information is crucial in the recovery of genomic sequences, and its absence may affect the accurate prediction of missing bases. Therefore, how to effectively integrate directionality information to improve the accuracy of genomic sequence recovery remains an urgent issue to be addressed.

This study proposes a method based on the Asymmetric Covariance Natural Vector (ACNV) [14], which further improves the accuracy of recovering missing bases by incorporating the directionality information of nucleotides. We first encoded the genomic sequences using the Asymmetric Covariance Natural Vector and then trained a deep learning model to learn the correspondence between DNA sequences and their natural vector representations, thereby predicting ambiguous nucleotides in genomic sequences. Experimental results show that, compared with traditional One-hot encoding, this method performs particularly well in predicting nucleotides at intermediate positions, indicating its ability to more effectively utilize information from neighboring nucleotides to improve prediction accuracy. This method provides a powerful technical support for the integrity repair and further analysis of genomic data.

2. Materials and Methods

2.1. Methods

Natural Vector The Natural Vector is a method that converts nucleotide sequences into numerical form to characterize the distribution features of genomic sequences [12]. For example, for any genomic sequence $S = s_1, s_2, \ldots, s_n$ (with length n), where each nucleotide s_i belongs to the set $L = \{A, C, G, T/U\}$, the indicator function $I_k(\cdot) : L \to \{0, 1\}$ is defined for a nucleotide $k \in L$, with the expression:

$$I_k(s_i) = \begin{cases} 1, & \text{if } s_i = k, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Let $s_i \in L$, where $i = 1, 2, \ldots, n$ denotes the i-th nucleotide in the sequence. We define the following statistical metrics:

- The occurrence count of nucleotide k in sequence S:

$$n_k = \sum_{i=1}^{n} I_k(s_i) \tag{2}$$

- The average position of nucleotide k in sequence S:

$$\mu_k = \frac{1}{n_k} \sum_{i=1}^{n} i \cdot I_k(s_i) \tag{3}$$

- The j-th central moment of the positions of nucleotide k in sequence S:

$$D_k^j = \sum_{i=1}^{n} \frac{(i - \mu_k)^j I_k(s_i)}{n_k^{j-1} n^{j-1}}, \quad j = 2, \ldots, n_k \tag{4}$$

Here, j denotes the order of the central moment. When $j = 2$, a 12-dimensional Natural Vector is obtained, representing the counts, average positions, and second-order central moments of the four nucleotides in the sequence, as follows:

$$(n_A, n_C, n_G, n_T, \mu_A, \mu_C, \mu_G, \mu_T, D_A^2, D_C^2, D_G^2, D_T^2)$$

Symmetric Covariance Natural Vector The 12-dimensional Natural Vector can be used to describe the distribution features of individual nucleotides in genomic sequences, but it fails to effectively represent the interrelationships between nucleotides, thereby limiting its application in genomic sequence analysis. To address this issue, Sun et al. [13] introduced covariance to describe the interrelationships between nucleotide positions, thereby enhancing the representation capability of sequence features. For nucleotides $k, l \in L$, the indicator functions are defined as follows:

$$I_{kl}(s_i) = I_{lk}(s_i) = \begin{cases} 1, & \text{if } s_i = k \text{ or } s_i = l, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

The covariance of nucleotide positions $\mathrm{Cov}(k, l)$ for nucleotides k and l is defined as follows:

$$\mathrm{Cov}(k, l) = \frac{1}{n} \sum_{i=1}^{n} \frac{[i - \mu_k][i - \mu_l] I_{kl}(s_i)}{\sqrt{n_k}\sqrt{n_l}}, \quad n_k \neq 0 \text{ and } n_l \neq 0$$

Here, μ_k and μ_l represent the average positions of nucleotides k and l in the sequence, while n_k and n_l represent their respective occurrence counts. The covariance quantifies the positional correlation between nucleotides k and l in the genomic sequence. Therefore, any genomic sequence can be represented by an 18-dimensional Natural Vector, as follows:

$$(n_A, n_C, n_G, n_T, \mu_A, \mu_C, \mu_G, \mu_T, D_A^2, D_C^2, D_G^2, D_T^2, \mathrm{Cov}(A, C), \mathrm{Cov}(A, G), \ldots, \mathrm{Cov}(G, T))$$

Asymmetric Covariance Natural Vector The Symmetric Covariance Natural Vector only focuses on the positional relationships between nucleotides and fails to capture their directionality information. However, in nucleotide sequences, the order of adjacent nucleotides $\{s_i, s_j\}$ and $\{s_j, s_i\}$ may have different biological implications. The Asymmetric Covariance Natural Vector can distinguish directional relationships such as A→C and C→A, effectively capturing the sequential features of nucleotide arrangements.

Let the nucleotide sequence be $S = s_1 s_2 \ldots s_n$, where $s_i \in \{A, C, G, T/U\}$. For adjacent nucleotide pairs (s_i, s_{i+1}), the indicator function of nucleotide pairs $I_{m_1 m_2}(s_i s_{i+1})$ is defined as follows:

$$I_{m_1 m_2}(s_i s_{i+1}) = \begin{cases} 1, & \text{if } m_1 m_2 = s_i s_{i+1}, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Here, the indicator function $I_{m_1m_2}(s_i, s_{i+1})$ takes a value of 1 when $(s_i, s_{i+1}) = (m_1, m_2)$, and 0 otherwise. Based on this indicator function, the Asymmetric Covariance $a\text{-}Cov(m_1, m_2)$ between nucleotides m_1 and m_2 is defined as:
(7)

$$a\text{-}Cov(m_1, m_2) = \frac{1}{n}\sum_{i=1}^{n-1} \frac{[i-\mu_{m_1}][i-\mu_{m_2}]I_{m_1m_2}(s_is_{i+1})}{\sqrt{n_{m_1}}\sqrt{n_{m_2}}}, \quad n_{m_1} \neq 0 \text{ and } n_{m_2} \neq 0$$

Here, n_{m_1} and n_{m_2} represent the occurrence counts of nucleotides m_1 and m_2, while μ_{m_1} and μ_{m_2} represent their average positions.

Therefore, any given genomic sequence can be characterized by the Asymmetric Covariance Natural Vector, as follows:

$$(n_A, n_C, n_G, n_T, \mu_A, \mu_C, \mu_G, \mu_T, a\text{-}Cov(A, C), \ldots, a\text{-}Cov(C, A), \ldots, a\text{-}Cov(G, T)) .$$

Here, n_A, n_C, n_G, n_T represent the counts of these nucleotides in the sequence, while $\mu_A, \mu_C, \mu_G, \mu_T$ represent the position means of these nucleotides in the sequence. The remaining part of the vector represents the Asymmetric Covariance between nucleotide pairs with respect to their positional information.

Example Below, we illustrate the calculation process using the nucleotide sequence ACGGTAGTCA as an example. First, we calculate the counts and average positions of each nucleotide:

$$n_A = 3, \quad n_C = 2, \quad n_G = 3, \quad n_T = 2,$$
$$\mu_A = 5.67, \quad \mu_C = 5.5, \quad \mu_G = 4.67, \quad \mu_T = 6.5$$

According to Equation (6), the indicator functions of nucleotides are calculated, as shown in the following table:

Table 1: Indicator function between nucleic acids

sequence	**A**	**C**	**G**	**G**	**T**	**A**	**G**	**T**	**C**	**A**
Position encoding	1	2	3	4	5	6	7	8	9	10
$I_{AC}(s_i, s_{i+1})$	1	0	0	0	0	0	0	0	0	0
$I_{AG}(s_i, s_{i+1})$	0	0	0	0	0	1	0	0	0	0
$I_{CA}(s_i, s_{i+1})$	0	0	0	0	0	0	0	0	1	0
$I_{CG}(s_i, s_{i+1})$	0	1	1	0	0	0	0	0	0	0
$I_{GG}(s_i, s_{i+1})$	0	0	1	0	0	0	0	0	0	0
$I_{GT}(s_i, s_{i+1})$	0	0	0	1	0	0	1	0	0	0
$I_{TA}(s_i, s_{i+1})$	0	0	0	0	1	0	0	0	0	0
$I_{TC}(s_i, s_{i+1})$	0	0	0	0	0	0	0	1	0	0

Using the data from the above table and combining it with Equation (7), the Asymmetric Covariance values for each nucleotide pair can be calculated step by step:

$$
\begin{aligned}
a\text{-}Cov(A,C) &= \sum_{i\in\{1,2\}} \frac{[i-5.67][i-5.5]}{10\cdot\sqrt{3}\cdot\sqrt{2}} = 1.381,\\
a\text{-}Cov(A,G) &= \sum_{i\in\{6,7\}} \frac{[i-5.67][i-4.67]}{12\cdot\sqrt{2}\cdot\sqrt{3}} = 0.119,\\
a\text{-}Cov(G,G) &= \sum_{i\in\{3,4\}} \frac{[i-4.67][i-4.67]}{10\cdot\sqrt{3}\cdot\sqrt{3}} = 0.107,\\
a\text{-}Cov(G,T) &= \sum_{i\in\{4,5,7,8\}} \frac{[i-4.67][i-6.5]}{10\cdot\sqrt{3}\cdot\sqrt{2}} = 0.299,\\
&\cdots,\\
a\text{-}Cov(C,A) &= \sum_{i\in\{9,10\}} \frac{[i-5.5][i-5.67]}{10\cdot\sqrt{2}\cdot\sqrt{3}} = 1.272
\end{aligned}
$$

Therefore, the nucleotide sequence ACGGTAGTCA can be represented by a 24-dimensional Asymmetric Covariance Natural Vector as follows:

$$(3,2,3,2,5.67,5.5,4.67,6.5,0,0.119,1.381,0,0,0.107,0,0.299,1.272,0.551,0,0,0.034,0,0.625,0).$$

High-Dimensional Extension of ACNV Asymmetric Covariance has significant advantages in the high-dimensional extension of feature space. Symmetric Covariance only considers the relationships between nucleotide pairs, and its feature dimensions are limited by the combination numbers (e.g., $C_4^2 = 6$, $C_4^3 = 4$, and $C_4^4 = 1$), making it difficult to fully represent the relationships among multiple nucleotides. In contrast, Asymmetric Covariance introduces directionality information, allowing 2-mers to generate $4 \times 4 = 16$ features and 3-mers to expand to $4^3 = 64$, significantly enhancing feature representation capabilities.

Traditional k-mer approaches construct features by counting the frequencies of subsequences of length k, but they fail to effectively capture directionality information and often result in redundant positional information. In comparison, the Asymmetric Covariance Natural Vector inherits the feature richness of k-mer approaches and combines directional indicator functions with positional information to enhance the representation capability of genomic sequences.

To achieve high-dimensional feature space extension, the formula for k-mer Asymmetric Covariance is provided below. For any nucleotide sequence fragment of length k, $m_1, \cdots, m_k$, the Asymmetric Covariance is defined as follows:

(8)
$$a\text{-}Cov(m_1, \cdots, m_k) = \frac{1}{n} \sum_{i=1}^{n-k+1} \prod_{j=1}^{k} \frac{[i - \mu_{m_j}] \cdot I_{m_1,\cdots,m_k}(s_j \cdots s_{j+k-1})}{\sqrt{n_{m_j}}}, \quad n_{m_j} \neq 0.$$

Here, the indicator function $I_{m_1,\cdots,m_k}(s_j \cdots s_{j+k-1})$ is defined as:

(9)
$$I_{m_1,\cdots,m_k}(s_j \cdots s_{j+k-1}) = \begin{cases} 1, & \text{if } s_j \cdots s_{j+k-1} = m_1, \cdots, m_k \\ 0, & \text{otherwise} \end{cases}$$

2.2. Data

HIV Dataset The HIV dataset is sourced from the HIV Database (https://www.hiv.lanl.gov), containing 5666 genomic sequences with lengths ranging from 8023 to 10280, and an average length of 8998. Each sequence was segmented into subsequences of length 32, resulting in a total of 1257470 samples. After deduplication and preprocessing, a total of 740051 unique subsequences were retained for experimentation.

SARS-CoV-2 Dataset The SARS-CoV-2 dataset was sourced from the *GISAID* database (https://www.gisaid.org). Using the *EPICOV* interface, we downloaded the relevant *FASTA* files and metadata. After preprocessing, a total of *16902654* sequences were obtained. For ease of analysis, the data were categorized by variant type, with the following counts: *Delta* (4624735), *Omicron* (7755563), *Alpha* (1212374), *Gamma* (136649), *Beta* (44918) and *Other* (3128415), where "other" represents an uncertain strain type. Based on this, we extracted the S protein sequences for each category and performed deduplication. The statistical results after deduplication are shown in Table 2.

Table 2: SPotein data statistics

Category	Pre-deduplication count	Post-deduplication count	Deduplication ratio (%)
alpha S protein	1191343	206826	17.4
delta S protein	4500061	1242423	27.6
gamma S protein	134204	37802	28.2
beta S protein	35986	16432	45.7
omicron S protein	7496995	1911689	25.5
other S protein	2959606	1089979	36.8

In the experiment, we selected the S protein sequences of both the Alpha and Delta variants for analysis.

3. Results

3.1. Distribution Characteristics of ACNV in Gene Sequence Recovery

Dataset and Experimental Setup. In this experiment, we used the HIV dataset and segmented each genomic sequence into subsequences of length 32. Subsequently, nucleotides in each subsequence were sequentially replaced with "N" from left to right to simulate gene sequence missingness. The label for each sample was the original nucleotide sequence that had not been replaced with "N". To evaluate the effectiveness of the Asymmetric Covariance Natural Vector (ACNV) encoding method, we compared it with the traditional Natural Language Processing (NLP) embedding encoding method [15, 16].

First, we encoded the genomic sequences using the Asymmetric Covariance Natural Vector with k-mer = 3. The input sequences, consisting of $\{A, G, C, T, N\}$, generated 160-dimensional feature vectors, calculated as:

$$10 + 5^2 + 5^3 = 160$$

where:

- The 10 represents the counts of nucleotides A, C, G, T, and N along with their average positional information.
- The term 5^2 corresponds to the second-order asymmetric covariance calculation for nucleotide pairs ($k = 2$).
- The term 5^3 corresponds to the third-order asymmetric covariance calculation ($k = 3$).

Since 'N' does not carry any meaningful biological information, directly using the 160-dimensional features containing 'N' as input may lead the model to overemphasize missing data, thereby affecting its performance. Therefore, we removed the values associated with 'N' and retained 88 valid features, which helps reduce computational noise and ensures that the model focuses on informative sequence features, ultimately improving prediction accuracy.

Therefore, the final input features consist of an 88-dimensional feature vector, which are computed as follows:

$$8 + 4^2 + 4^3 = 88$$

This follows the same principles, with the nucleotide set being $\{A, G, C, T\}$. In practice, the input feature for each sample is reshaped to a tensor of shape $B \times 88 \times 1$, where B denotes the batch size.

For a detailed list of all possible 3-mer combinations and the corresponding dimension calculations, please refer to **Appendix A**.

Model Training and Evaluation The model was trained and evaluated using GRU [17]. We converted the Asymmetric Covariance Natural Vector into the input dimensions required by the GRU (num2seq), which include batch size, natural vector length (88 in this experiment), and feature dimension (set to 1). Additionally, we used traditional NLP encoding method (seq2seq) to encode the genomic sequences and conducted a comparison experiment. For all experiments, the learning rate was set to 0.0001, the batch size was 128, two GRU layers were used, each with 1024 hidden units, the activation function was tanh, and the loss function was CrossEntropyLoss.

The experimental results are shown in Figure 2, where we tested two cases with sequence lengths of 32 and 64. When comparing seq2seq with our

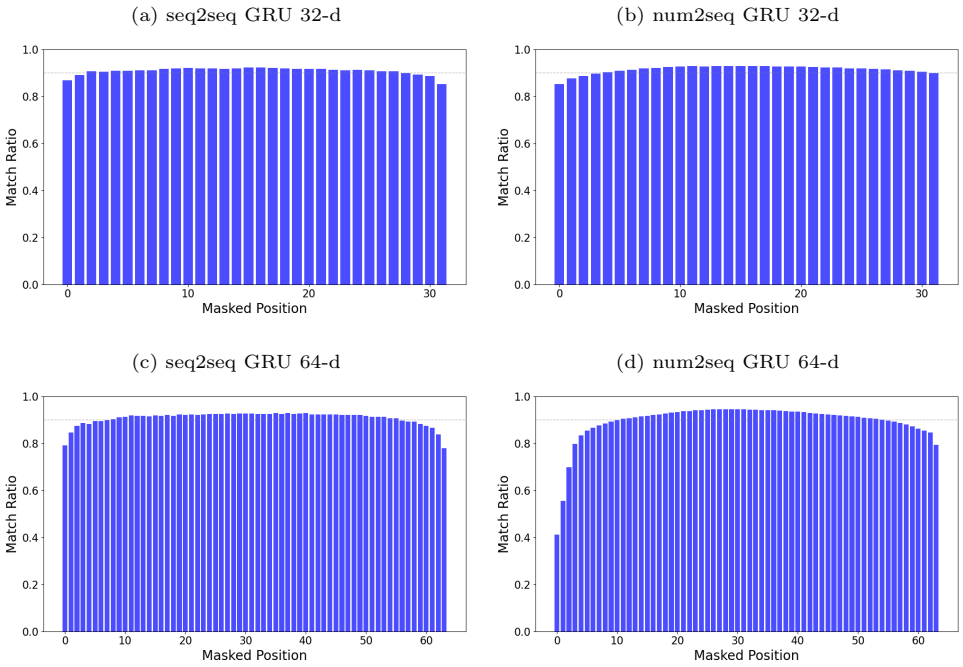

Figure 2: Matching ratio distributions for different encoding methods

num2seq method, both achieved similar overall accuracy. For 32-length sequences, the highest match ratio was 92.97% vs 92.42%, and for 64-length sequences, it was 93.84% vs 92.94%. However, our num2seq method shows a slight advantage in predicting central positions. As seen in the figure, compared to the seq2seq, the asymmetric natural vector encoding method exhibits a more stable match ratio. Specifically, the asymmetric natural vector encoding method shows a gradual and stable increase in accuracy from both sides toward the center, particularly at positions 16 and 17, where the prediction accuracy improves significantly. Additionally, the prediction results exhibit characteristics of a normal distribution (Figures b and d). This suggests that the method effectively utilizes information from adjacent positions to accurately predict the characteristics of the central positions, leading to more stable prediction results.

In contrast, the traditional seq2seq method shows higher accuracy at the central positions, but its performance lacks consistency. As shown in Figure a, although the highest accuracy occurs at the very center, the entire process does not demonstrate a stable increasing trend. Furthermore, in Figure d, the seq2seq method does not achieve the highest accuracy at the central position, and in some cases, its prediction accuracy exceeds that at the central positions.

This disparity highlights the advantages of the asymmetric natural vector encoding method. Through a stable growth trend, this method requires only the information from adjacent positions of the missing base to accurately predict its value, while also demonstrating a high degree of stability in the prediction results.

3.2. Base Recovery of SARS-CoV-2 Variant Using ACNV

Model Setup. To evaluate the superiority of the Asymmetric Covariance Natural Vector method in gene sequence recovery, we used the S protein gene sequences of the SARS-CoV-2 Alpha and Delta variant from the dataset to further validate its performance in recovering missing bases. During the training phase, given the presence of numerous missing bases in the dataset, we first removed sequences containing non-standard bases. Based on preliminary experimental analysis, we replaced the bases in the center of each sequence with 'N' to simulate missing bases and trained the model using a GRU network. For all experiments, the learning rate was set to 0.0001, the batch size was 128, two GRU layers were used, each with 1024 hidden units, the activation function was tanh, and the loss function was CrossEntropyLoss. Using the best-performing model obtained from training, we predicted the non-standard bases replaced with "N".

Prediction on Non-Standard Bases. To further validate the model's performance on real-world data, we applied the trained GRU model to sequences containing non-standard bases for prediction. First, we selected sequences containing non-standard bases and replaced these bases with "N". Subsequently, we used the trained GRU model to predict these modified sequences and checked the prediction results against the IUPAC rules (see Table A1 in Appendix A) to ensure compliance with standard base encoding criteria.

In the Alpha dataset, we first replace each non-standard base with "N" and then predict the scores for each category using the model. The highest SoftMax score is selected as the prediction for the non-standard base, as shown in step 1 of Figure 3. Next, we perform a table lookup comparison of the predicted results based on the IUPAC rules and count the samples that conform and do not conform to the IUPAC rules, as shown in step 2 of Figure 3.

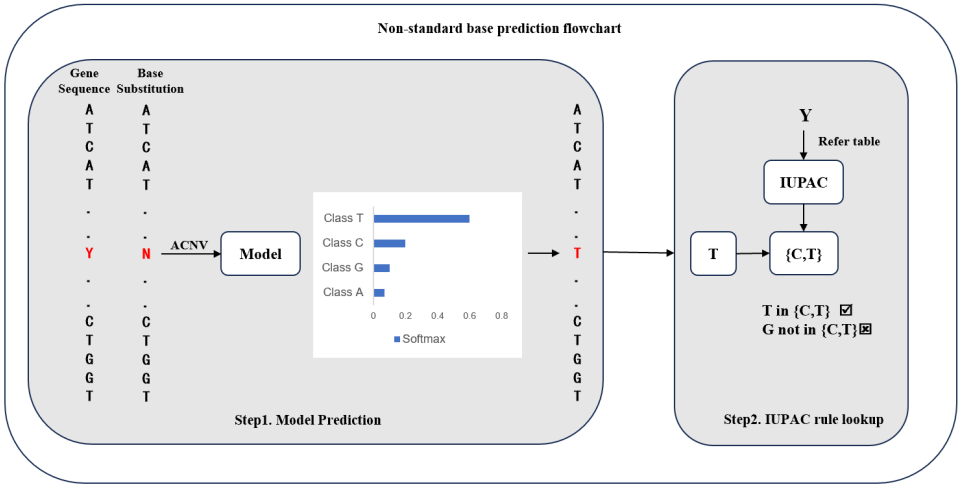

Figure 3: Flowchart of non-standard base prediction. Step 1: First, extract the sequences of equal length from both sides of the non-standard base, then replace them with "N". The sequences are then encoded using asymmetric natural vectors, and the model predicts the corresponding AGCT sequence. Step 2: Query and determine compliance based on the IUPAC rules. If compliant, the prediction is considered correct.

The left panel of Figure 4 shows the SoftMax scores of the predicted results that conform to the IUPAC rules for both the Alpha and Delta datasets.

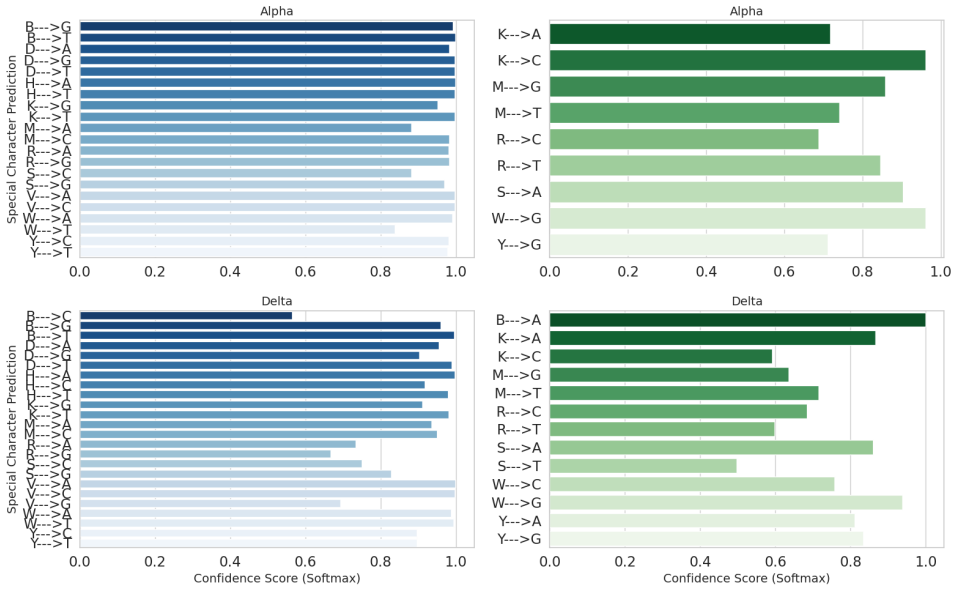

Figure 4: Average probability scores for predicting special characters in the invalid and validation datasets.

As observed, the majority of predictions have scores exceeding 95%, suggesting that our method has high confidence in restoring missing bases. The right panel displays the predicted results that do not conform to the IUPAC rules, where the maximum SoftMax scores are lower, likely due to the limitations of sequencing technology, which may prevent accurate restoration of standard bases.

Finally, we calculated the proportion of predictions that did not conform to the IUPAC rules. In the Alpha dataset, there are 47,915 samples that conform to the IUPAC rules and 52 that do not, resulting in an error rate of 0.11% (52 / (52 + 48,392)). In the Delta dataset, there are 329,791 samples that conform to the IUPAC rules and 717 that do not, resulting in an error rate of 0.22% (717 / (717 + 329,791)). These results demonstrate that the asymmetric covariance natural vector method has a low error rate in restoring missing bases, validating its effectiveness across different datasets.

4. Discussion

This study proposes a method based on the Asymmetric Covariance Natural Vector, combined with Recurrent Neural Networks (GRU), to predict

missing bases in gene sequences. Compared to traditional NLP embedding encoding methods, Asymmetric Covariance Natural Vector introduces directionality information, overcoming the limitations of these methods in capturing the relative relationships between nucleotides. In gene sequences, the order of adjacent nucleotides may carry important biological information. Therefore, directionality information is crucial for improving prediction accuracy. Asymmetric Covariance Natural Vector effectively captures these sequential relationships, thereby significantly enhancing the precision of missing base recovery.

On the HIV dataset, this method outperforms traditional NLP embedding encoding methods in recovering missing bases at the most central positions, where Asymmetric Covariance Natural Vector significantly improves recovery accuracy. For the SARS-CoV-2 Alpha variant dataset, Asymmetric Covariance Natural Vector achieves a recovery error rate of only 0.11% under the IUPAC encoding rules. For the Delta dataset, Asymmetric Covariance Natural Vector achieves a recovery error rate of 0.22%. These results demonstrate that Asymmetric Covariance Natural Vector effectively enhances prediction accuracy when dealing with gene sequences containing missing bases, meeting the accuracy requirements for practical applications.

Despite its promising performance in this study, Asymmetric Covariance Natural Vector still has some limitations. First, the current study is based on fixed-length gene fragments of 32 nucleotides for prediction. Future research needs to explore how to handle longer gene sequences, such as increasing the sequence length to 512 and sampling by randomly masking 1-2 positions, to further enhance the model's robustness and generalization ability. Second, although the method achieved excellent results on the SARS-CoV-2 Alpha and Delta variant dataset, its applicability and stability need further validation on other viral strains or more complex genomic data, such as the human genome.

Future research can further improve the Asymmetric Covariance Natural Vector method in several aspects. First, the applicability of the method can be expanded to more complex genomic data, such as the human genome and bacterial genomes. Second, integrating more advanced deep learning techniques, such as the self-attention mechanism (Transformer), can improve the modeling capability for long sequences and further enhance the model's prediction accuracy.

In summary, the Asymmetric Covariance Natural Vector Asymmetric Covariance Natural Vector provides an effective encoding method for gene sequence analysis, particularly excelling in the recovery of missing bases. The

experimental results validate its potential for repairing the integrity of genomic data and provide technical support for further genomic data analysis tasks. Future work will focus on expanding the application scope of this method, improving its computational efficiency, and exploring how to achieve better performance in a broader range of genomic data analyses.

Data, Materials, and Software Availability

The code used for implementing experimental methods and evaluating validation algorithms has been uploaded to the GitHub repository (https://github.com/karlieswift/ACNVMaskRecover). All other relevant data are described in the manuscript.

Appendix A. Supplementary Information

A.1. IUPAC Base Encoding Rules

Table A1: IUPAC Nucleotide Base Encoding Rules

Symbol	Corresponding Bases	Description
A	Adenine (A)	
C	Cytosine (C)	
G	Guanine (G)	
T	Thymine (T)	
U	Uracil (U)	Used only in RNA
R	Adenine (A) or Guanine (G)	Purine (A or G)
Y	Cytosine (C) or Thymine (T)	Pyrimidine (C or T)
S	Guanine (G) or Cytosine (C)	Strong interaction (G or C)
W	Adenine (A) or Thymine (T)	Weak interaction (A or T)
K	Guanine (G) or Thymine (T)	Keto bases (G or T)
M	Adenine (A) or Cytosine (C)	Amino bases (A or C)
B	Cytosine (C), Guanine (G), or Thymine (T)	Not Adenine (A)
D	Adenine (A), Guanine (G), or Thymine (T)	Not Cytosine (C)
H	Adenine (A), Cytosine (C), or Thymine (T)	Not Guanine (G)
V	Adenine (A), Guanine (G), or Cytosine (C)	Not Thymine (T)
N	Any base (A, T, C, or G)	Any nucleotide

Appendix B. Detailed Explanation of 3-mer Combinations and Feature Dimension Calculation

In this appendix, we provide a detailed explanation of how to calculate gene sequences containing standard bases (A, C, G, T) and non-standard bases (N)

using the Asymmetric Covariance Natural Vector.

B.1. Calculation Process of Asymmetric Covariance Natural Vector

Masked gene sequences consist of elements from the set $S = \{A, C, G, T, N\}$, where A, C, G, T represent the four standard nucleotides, and N represents masked positions (i.e., uncertain bases).

For asymmetric 2-mers, the total number of possible combinations is $10 + 5 \times 5 = 35$, as shown below:

$$\begin{aligned} &AA,\ AC,\ AG,\ AT,\ AN \\ &CA,\ CC,\ CG,\ CT,\ CN \\ &GA,\ GC,\ GG,\ GT,\ GN \\ &TA,\ TC,\ TG,\ TT,\ TN \\ &NA,\ NC,\ NG,\ NT,\ NN \end{aligned}$$

The combinations containing N are in the 5th, 10th, 15th, 20th, 25th, 30th, 31th, 32th, 33th, 34th, and 35th positions. Thus, the number of effective 2-mer combinations is $35 - 11 = 24$. For the output sequence set $S = \{A, C, G, T\}$, the dimension contribution is:

$$8 + 4^2 = 8 + 16 = 24$$

In this experiment, we used the Asymmetric Covariance Natural Vector with $k = 3$ to recover missing bases. For input sequences composed of the set $\{A, C, G, T, N\}$, the number of possible 3-mer combinations is $5^3 = 125$.

Ultimately, the feature dimensions of the Asymmetric Covariance Natural Vector are calculated by summing the contributions of base frequencies, average positions, and 2-mer and 3-mer combinations. The specific calculation steps are as follows:

- **For the input sequence set** $S = \{A, C, G, T, N\}$, the dimension contribution is:

$$10 + 5^2 + 5^3 = 10 + 25 + 125 = 160$$

- **For the valid input sequence set** $S = \{A, C, G, T\}$, the dimension contribution is:

$$8 + 4^2 + 4^3 = 8 + 16 + 64 = 88$$

Acknowledgements

This work is supported by National Natural Science Foundation of China (NSFC) grant (12171275) and Tsinghua University Education Foundation fund (042202008).

References

[1] Ewing, Brent, et al. "Base-calling of automated sequencer traces using-Phred. I. Accuracy assessment." Genome research 8.3 (1998): 175-185.

[2] Li, Heng, and Richard Durbin. "Fast and accurate short read alignment with Burrows–Wheeler transform." bioinformatics 25.14 (2009): 1754-1760.

[3] Shendure, Jay, and Hanlee Ji. "Next-generation DNA sequencing." Nature biotechnology 26.10 (2008): 1135-1145.

[4] Cornish-Bowden, Athel. "Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984." Nucleic acids research 13.9 (1985): 3021.

[5] Shu, Yuelong, and John McCauley. "GISAID: Global initiative on sharing all influenza data–from vision to reality." Eurosurveillance 22.13 (2017): 30494.

[6] Giudicelli, Véronique, Denys Chaume, and Marie-Paule Lefranc. "IMGT/GENE-DB: a comprehensive database for human and mouse immunoglobulin and T cell receptor genes." Nucleic acids research 33.suppl_1 (2005): D256-D261.

[7] Angermueller, Christof, et al. "Deep learning for computational biology." Molecular systems biology 12.7 (2016): 878.

[8] Shen, Xiaoxi, et al. "A brief review on deep learning applications in genomic studies." Frontiers in Systems Biology 2 (2022): 877717.

[9] Liu, Jianxiao, et al. "Application of deep learning in genomics." Science China Life Sciences 63 (2020): 1860-1878.

[10] Alipanahi, Babak, et al. "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning." Nature biotechnology 33.8 (2015): 831-838.

[11] Lopez-del Rio, Angela, et al. "Effect of sequence padding on the performance of deep learning models in archaeal protein functional prediction." Scientific reports 10.1 (2020): 14634.

[12] Deng, Mo, et al. "A novel method of characterizing genetic sequences: genome space with biological distance and applications." PloS one 6.3 (2011): e17293.

[13] Sun, Nan, Xin Zhao, and Stephen S-T. Yau. "An efficient numerical representation of genome sequence: Natural vector with covariance component." PeerJ 10 (2022): e13544.

[14] Hu, Guoqing, Zhou, Tao, Zhou, Piyu, and Yau, Stephen. "Novel Natural Vector with Asymmetric Covariance for Classifying Biological Sequences." Gene (Submitted, 2025).

[15] Bengio, Yoshua, et al. "A neural probabilistic language model." Journal of machine learning research 3.Feb (2003): 1137-1155.

[16] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[17] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).

Guoqing Hu
Beijing Institute of Mathematical Sciences and Applications (BIMSA), Beijing 101408, P. R. China
E-mail address: drhu@bimsa.cn

Hao Wang
Institute of Statistics and Big Data, Renmin University of China, Beijing 100872, P. R. China
E-mail address: wanghao11@ruc.edu.cn

Stephen S.-T. Yau
Beijing Institute of Mathematical Sciences and Applications (BIMSA), Beijing 101408, P. R. China
Department of Mathematical Sciences, Tsinghua University, Beijing 100084, P. R. China
E-mail address: yau@uic.edu