Yang et al. *Zool. Res.* 2026, 47(5): 1417−1431
https://doi.org/10.24272/j.issn.2095-8137.2025.648
https://cstr.cn/32248.14.zr.2025.648

ZR | Zoological Research

Article Open Access

Deciphering ribosomal frameshifting determinants across species with a semi-supervised hybrid learning framework

Yuhao Yang[1,#], Juan Yang[2,#], Zijia Liu[1], Yuan Li[2], Weibo Song[3], Naomi A. Stover[4], Xiao Chen[2,*]

[1] *SDU-ANU Joint Science College*, *Shandong University*, *Weihai*, *Shandong* 264209, *China*

[2] *Marine College*, *Shandong University*, *Weihai*, *Shandong* 264209, *China*

[3] *Institute of Evolution and Marine Biodiversity*, *Ocean University of China*, *Qingdao*, *Shandong* 266003, *China*

[4] *Department of Biology*, *Bradley University*, *Peoria* 61625, *USA*

ABSTRACT

Programmed ribosomal frameshifting (PRF) is a conserved translational recoding mechanism that expands proteome diversity and regulates important biological processes across viruses, prokaryotes, and eukaryotes. However, existing predictors are often limited by their reliance on extensive annotated data, poor performance in non-model organisms, and restricted applicability across species or frameshift types. Here, we present FScanpy, a hybrid machine-learning model integrating histogram-based gradient boosting (HistGB) and a bidirectional long short-term memory-convolutional neural network (BiLSTM-CNN) architecture for cross-species PRF prediction. The model integrates short-range codon-position features associated with ribosomal pausing and long-range sequence contexts that reflect structural constraints around candidate shift sites. Using semi-supervised learning on 8 049 candidate sequences, FScanpy reduces reliance on manual annotation and improves model generalization. Across multiple datasets, it achieved an area under the receiver operating characteristic curve (AUC) of 0.951, and it also performed well on euplotid protists (AUC=0.877), supporting its utility in non-model systems with non-canonical recoding patterns. Feature interpretation further showed that AAA and GGG motifs, together with mRNA secondary-structure stability, as major determinants across datasets, highlighting conserved features of PRF from viruses to eukaryotes. These insights not only validate the model's accuracy but also deepen understanding of PRF's conserved biological roles. FScanpy establishes a novel and effective computational framework for PRF analysis by resolving challenges in feature representation and species generalizability, enabling the study of frameshifting sites and accelerating discovery of evolutionarily divergent recoding events.

Keywords: Protozoa; Cross-species; Ribosomal frameshifting; Hybrid learning; Slippery sequence; mRNA secondary structure

INTRODUCTION

During canonical translation, the ribosome maintains a single reading frame from the start codon to the first in-frame stop codon. Insertions or deletions not divisible by three disrupt this frame and often generate aberrant proteins. Ribosome-mediated protein synthesis is the final stage of biological information transfer, signifying irreversible commitment to gene expression (Rodnina, 2012; Voorhees & Ramakrishnan, 2013). Thus, the fidelity with which messenger RNA is translated is crucial for all living organisms, and spontaneous errors arising from the translation mechanism are extremely rare (Kramer & Farabaugh, 2007; Kurland, 1992). Unlike irreversible frameshift mutations caused by nucleotide insertions or deletions, programmed ribosomal frameshifting (PRF) is a dynamic translational event where the mRNA sequence remains physically unaltered, but the translating ribosome shifts its reading frame. When a ribosome encounters a specific signal sequence during translation, the ribosome shifts the reading frame in a predetermined manner. This allows the same mRNA template to produce different proteins. Initially discovered in viruses, this mechanism has since been found in various other organisms. In viruses, PRF regulates reading frames, thereby controlling the production of key enzymes. For example, gag-pol frameshifting occurs in human immunodeficiency virus (HIV) and other retroviruses (Korniy et al., 2019). In mammals, the antizyme gene (OAZ) regulates polyamine metabolism via +1 PRF (Matsufuji et al., 1995). The expression of the ABP140 and EST3 genes in

This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/4.0/), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright ©2026 Editorial Office of Zoological Research, Kunming Institute of Zoology, Chinese Academy of Sciences

Received: 13 December 2025; Accepted: 15 May 2026; Online: 02 June 2026

Foundation items: This work was supported by National Natural Science Foundation of China (Grant No. 32270512), the Programs of Distinguished Young Scholar and Interdisciplinary Cultivation by Shandong University and US National Institutes of Health (Grant No. P40OD010964).

#Authors contributed equally to this work

*Corresponding author, E-mail: xc@sdu.edu.cn

yeast also relies on programmed frameshifting (Fenton et al., 2025; Taliaferro & Farabaugh, 2007).

PRF generally occurs at a very low frequency, approximately 10^{-3} to 10^{-4} per codon (Chen et al., 2014). However, in a group of ciliates called euplotids, the frequency of PRF is significantly higher, with more than 10% of their genes predicted to undergo PRF events (Jin et al., 2023; Wang et al., 2016). Unlike other ciliates (Lyu et al., 2024; Ye et al., 2025), the conventional stop codon UGA is reassigned to encode cysteine (Meyer et al., 1991) or selenocysteine (Turanov et al., 2009) in these species. The genes in *Euplotes* that have been shown to undergo +1 PRF events include the regulatory subunit of cyclic AMP-dependent protein kinase and a nuclear protein kinase in *Euplotes octocarinatus* (Tan et al., 2001a, 2001b), a La motif protein (p43) in *E. aediculatus* (Aigner et al., 2000), the ORF2 protein of the Tec2 transposon in *E. crassus* (Doak et al., 2003; Jahn et al., 1993), and the reverse transcriptase subunits of telomerase (TERT) (Möllenbeck et al., 2004; Wang et al., 2002).

PRF typically involves two key elements within the mRNA: a "slippery" sequence and a secondary structure element (Hansen et al., 2007; Wu et al., 2018). The −1 slippery sequence generally conforms to the consensus pattern X-XXY-YYZ. In this pattern, the codons XXY and YYZ realign to form XXX and YYY, respectively. XXX represents a triplet of identical nucleotides, YYY represents a triplet of either adenine (A) or uracil (U), and Z can be either adenine (A), cytosine (C), or uracil (U). This motif alone usually supports only a low basal level of frameshifting, typically around 1%. However, the presence of downstream mRNA secondary structures, such as a stem-loop or pseudoknot, can significantly increase frameshifting efficiency, sometimes by up to 30–50% (Giedroc et al., 2000). The coronavirus infectious bronchitis virus (IBV) possesses a pseudoknot characterized by a long S1 stem of 11–12 base pairs coupled with a long L2 loop, which efficiently stimulates frameshifting when positioned downstream of the slippery sequence U UUA AAC (Brierley, 1995; Brierley et al., 1989). In beet western yellows virus (BWYV) and related plant luteoviruses, an H-type pseudoknot promotes −1 frameshifting within the P1 gene. This regulates the expression of an RNA-dependent RNA polymerase encoded by the P2 viral gene as a P1-P2 fusion protein (Brierley et al., 1989). In contrast to the well-characterized −1 PRF, +1 PRF is less common overall; however, with the advancement of high-throughput sequencing and other technical approaches in protozoan research (Gao et al., 2025; Hao et al., 2025; Jiang et al., 2025; Wang et al., 2025b; Ye et al., 2025; Zhang et al., 2025; Zheng et al., 2025), PRF events in *Euplotes* have been extensively identified, and notably, +1 PRF is utilized at an exceptionally high frequency in ciliates of the genus *Euplotes* (Jin et al., 2023; Wang et al., 2016). This frameshifting is not typically stimulated by downstream secondary structures; in fact, studies have shown that elements like pseudoknots are not associated with these PRF events in *Euplotes*. Instead, the process is driven by a specific slippery motif, AAA-UAR (where R is a purine, A or G), which contains a "shifty stop" codon (UAA or UAG) (Lobanov et al., 2017).

The biological significance of PRF is widely recognized, and many case studies have elucidated the molecular events during frameshifting (Atkins et al., 2016; Korniy et al., 2019). Conventional methodologies employed in PRF research include fluorescence-based reporter assays, ribosome profiling, sequence analysis, and computational modeling. Systematic research and modeling of these data to precisely predict PRF still face multiple challenges. Existing tools, such as KnotInFrame (designed for −1 PRF detection) (Theis et al., 2008) and ORFeus (based on ribosome profiling data) (Richardson & Eddy, 2023), each addresses part of the problem, but collectively they still face three major challenges: (1) they cannot handle multi-frameshift events, such as simultaneous +1 and −1 PRF; (2) they rely on high-quality ribosome profiling data, which is often unavailable; and (3) they perform poorly in eukaryotic systems, especially for +1 PRF, due to their overreliance on prokaryotic paradigms. FSFinder2, which analyzes mRNA sequence features (Byun et al., 2007), and the recently proposed PRFect model have attempted to overcome these limitations (Mcnair et al., 2024). However, their species generalizability remains inadequate, and their performance in test datasets is unsatisfactory. Recent studies emphasize that a comprehensive understanding of PRF requires integrating multiple sequence and structural features beyond isolated motifs (Mathew et al., 2015; Mikl et al., 2020). Specifically, a previous study has shown that relying solely on partial frameshifting features, such as slippery site identity or downstream secondary structure rigidity, is insufficient for accurate prediction, even within the same species and frameshifting type (Mikl et al., 2020). Furthermore, the heterogeneity of eukaryotic PRF regulatory mechanisms, coupled with the scarcity of relevant databases, severely impedes research progress in this area. Therefore, a central challenge is to overcome the dearth of high-confidence PRF annotations, particularly in non-model eukaryotes, and to develop a robust computational predictor that can deliver biologically interpretable and evolutionarily informative results without depending on abundant Ribo-seq data.

Here we present FScanpy, a hybrid model that combines histogram-based gradient boosting (HistGB) and bidirectional long short-term memory-convolutional neural network (BiLSTM-CNN) for cross-species PRF prediction (Figure 1). The model uniquely integrates short-range codon-position (11-codon windows) features with long-range sequence (133-codon windows) contexts to simultaneously capture ribosomal pausing dynamics and mRNA structural constraints. Through semi-supervised learning applied to both high-confidence and expanded sampling sequences, FScanpy reduces annotation dependency and enhances cross-species generalization. Furthermore, it systematically extracts conserved frameshifting determinants, including sequence motifs, structural signatures and eukaryotic regulatory patterns, to create a unified, mRNA-sequence-driven framework for accurately predicting frameshift probability. We demonstrate that not only does FScanpy achieve high prediction accuracy (AUC=0.951), it also provides interpretable insights into the evolutionarily conserved sequence-structure determinants of PRF. The model provides a prioritized list of candidate targets for subsequent experimental validation by integrating confidence scores with predicted probabilities. FScanpy provides a novel and effective computational framework for PRF analysis, addressing key challenges in feature representation and cross-species generalization. This framework allows researchers to study potential frameshifting sites and speed up the study of the mechanisms of evolutionarily divergent recoding events.

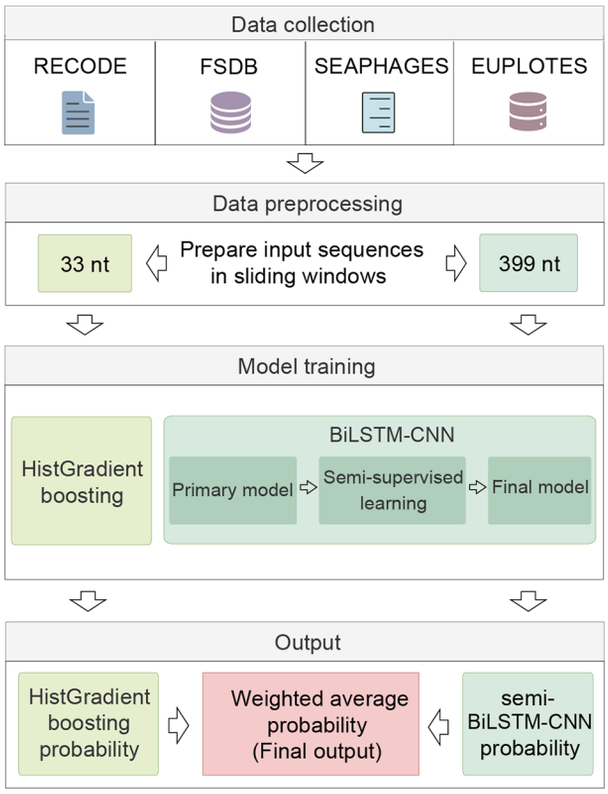

Figure 1 Framework of FScanpy

Data were acquired from RECODE, FSDB, SEAPHAGES and EUPLOTES databases. Frameshifting regions were identified and extracted as part of data preprocessing, and these data were used to generate 33 nt and 399 nt sequence window inputs. The model was trained using HistGB and BiLSTM-CNN approaches, and their strengths were integrated using a weighted average method.

MATERIALS AND METHODS

Data acquisition

Programmed ribosomal frameshifting (PRF) data were obtained from the RECODE (Recoding Event Compilation Database; https://recode.ucc.ie) (Baranov et al., 2001), FSDB (Frameshift Signal Database; http://wilab.inha.ac.kr/fsdb) (Moon et al., 2007), SEAPHAGES (Science Education Alliance–Phage Hunters Advancing Genomics and Evolutionary Science; https://phagesdb.org) (Russell & Hatfull, 2017), and EUPLOTES datasets (https://evan.ciliate.org) (Chen et al., 2019). RECODE is a database that compiles "programmed" translational recoding events, including PRF and codon redefinition. These recoding events play important roles in gene expression, particularly in viruses and bacteria. FSDB is a database dedicated to PRF signals. It provides detailed information on both experimentally verified and computationally predicted frameshifting events. The database also offers a graphical signal viewer and allows users to identify potential frameshifting sites in genomic sequences using the FSFinder program. The data for RECODE, FSDB, and SEAPHAGES were all sourced from the raw GenBank files provided in McNair et al. (Mcnair et al., 2024), the acquisition method of the SEAPHAGES data is similar to that used in their work, which involves searching for discontinuous locations within these files. EUPLOTES dataset contains the high-confidence PRF sites in macronuclear genomes of *Euplotes vannus*, *E. octocarinatus, E. woodruffi* and *E. crassus* predicted by FScanR (https://github.com/seanchen607/FScanR), a method for *de novo* prediction of PRF events across species. FScanR is based on the BLASTX alignment results of protein sequences from homologous species. It identifies DNA sequences where PRF occurs during translation, resulting in hit1 and hit2 not being in the same reading frame in the BLASTX output as positive samples. This similarity search-based pipeline is a well-established and robust methodology for PRF identification in *Euplotes*, having been previously employed in both genome-wide and comparative transcriptome studies that successfully identified thousands of frameshift sites in this organism (Wang et al., 2016, 2025a). Confidence levels were assigned using a weighted score combining mismatch count, e-value significance, and gap distance (<10 bases), with higher scores indicating greater reliability. To avoid errors caused by translation interruptions, we defined the gap open between hit1 and hit2 as less than ten nucleotides during the alignment process.

To evaluate the performance of the model, we used five external validation datasets: *Euplotes aediculatus*, *E. woodruffi*, *E. crassus*, Xu, and Atkins (Atkins et al., 2016; Feng et al., 2022; Jin et al., 2023; Xu et al., 2004). The three additional Euplotes datasets contain 504, 541, and 781 frameshifting events identified by FScanR, respectively. The Xu dataset includes 28 conserved frameshifting events in bacteriophage tail assembly chaperone genes on phage genomes. The Atkins dataset includes 110 confirmed/presumed ribosomal frameshifting genes from eukaryotic viruses.

Data preprocessing

A semi-supervised machine learning approach was employed to screen and analyze the SEAPHAGES and EUPLOTES datasets. For the SEAPHAGES dataset, we identified 68 high-confidence samples based on gene products, gene annotations, and annotation completeness. The high-confidence subset was defined via a strict ranking system based on keyword filtering (e.g., ribosomal slippage, frameshift) in the GenBank product and note fields. Samples were assigned a rank level (1 to 8) according to the presence of these keywords and annotation completeness, and only samples with Rank≤5 were designated as high-confidence and served as anchors for the KNN expansion (68 high-confidence samples). We then used the K-Nearest Neighbors (KNN) algorithm to identify medium- and low-confidence samples within a defined distance from these high-confidence samples and incorporated them into the training set. For KNN selection, distance was calculated using the Euclidean method on standardized feature vectors derived from one-hot encoded sequences. The threshold was set dynamically: a candidate sample was included if its average distance to its five nearest high-confidence neighbors was below the 80th percentile of intra-class distances for medium-confidence candidates (Rank=6) or below the 50th percentile for expanded candidates (a total of 2 407 samples were used as the input pool, and 1 704 samples were obtained as the expanded candidates).

For the EUPLOTES dataset, we obtained the data using the FScanR tool and marked the reliability of the samples based on mismatch counts, gap-open distances, and e-value, thereby selecting high-confidence EUPLOTES samples (1 987 high-confidence samples). Samples falling below the

screening threshold were assigned to the expanded sampling pool (3 587 expanded samples). In summary, after screening by semi-supervised learning, the 8 049 candidate sequences (5 574 for EUPLOTES, 2 475 for SEAPHAGES) yielded 7 346 sequences for model construction. The SEAPHAGES dataset comprised 68 high-confidence samples and 1 704 expanded samples, while the corresponding numbers for the EUPLOTES dataset were 1 987 and 3 587, respectively.

Finally, we combined the high-confidence and expanded samples from the SEAPHAGES data (1 772 samples), the high-confidence EUPLOTES data (1 987 samples), and data from the RECODE (665 samples) and FSDB databases (282 samples) to form an initial training set (4 706 samples in total).

Model architectures and feature extraction

Histogram-based gradient boosting (HistGB) feature extraction: Histogram-based gradient boosting (HistGB) is an efficient ensemble learning algorithm that accelerates training by binning feature values into histograms, offering strong performance and robustness for tabular and sequence-derived feature data. In this study, the HistGB model was used to characterize local sequence patterns within an 11-codon (33 nt) window centered at the FS_start site in the 0-reading frame. To explicitly capture codon usage preferences and sequence composition associated with PRF, we constructed a comprehensive set of handcrafted features, including: (1) nucleotide composition and GC content; (2) frequencies of 3-mer (codon) motifs; (3) positional features of individual codons relative to the frameshift site; and (4) sequence complexity measured by Shannon entropy. These features were concatenated into a unified feature vector and used as input to a HistGradientBoostingClassifier, which integrates multiple decision trees with histogram-based discretization to achieve stable and accurate classification of frameshifting events.

BiLSTM-CNN architecture and internal feature fusion: The BiLSTM-CNN model is a hybrid deep learning framework that integrates Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (BiLSTM) networks, designed to synergistically capture both local sequence motifs and long-range contextual dependencies—two key characteristics critical for PRF prediction. To achieve this, the model processes input sequences within a 133-codon (399 nt) window centered at the FS_start site, adopting a hybrid architecture that takes integer-encoded nucleotide sequences as input.

This composite architecture employs two parallel branches to simultaneously extract local motif features and model long-term sequence dependencies, as detailed below:

CNN branch: Three parallel convolutional layers were utilized, with kernel sizes of 3, 5, and 7, respectively, each equipped with 64 filters. Each convolutional layer was followed by Batch Normalization to accelerate training convergence, a ReLU activation function to introduce non-linearity, and Max Pooling (kernel size=2) to downsample feature maps while preserving key local patterns.

BiLSTM branch: This branch consists of a two-layer Bidirectional Long Short-Term Memory (BiLSTM) network. The first BiLSTM layer contains 64 hidden units, outputting 128 features (due to bidirectional processing), while the second layer contains 32 hidden units, yielding a 64-dimensional output feature vector.

Internal feature fusion: To fully integrate the complementary representations learned by the two branches, the local motif features extracted by the CNN branch (dimension: 38 208) and the global sequence dependencies captured by the BiLSTM branch (dimension: 64) were concatenated into a single unified feature vector (total dimension: 38 272). This concatenated feature vector was then fed into subsequent fully connected layers. This internal fusion mechanism enables the model to synergistically weigh both local sequence motifs (critical for frameshift site recognition) and long-range structural constraints (relevant for contextual sequence regulation).

Regularization and optimization: To mitigate overfitting and enhance model generalization, a Dropout rate of 0.5 was applied immediately before the final classification layer. The model was optimized using the Adam optimizer with an initial learning rate of 1e-4. A ReduceLROnPlateau scheduler (factor=0.5, patience=2) was employed to adjust the learning rate dynamically, with validation loss as the monitoring metric (Supplementary Figure S1).

Model training

The model training procedure consisted of three stages, followed by a final decision-level fusion. The HistGB model was trained in a fully supervised manner, whereas the BiLSTM-CNN model was further refined through self-training.

Stage 1. HistGB model (fully supervised): The HistGB model was trained on short-sequence features (33 nt) extracted from the curated high-confidence dataset, including samples from FSDB, selected SEAPHAGES, RECODE, and the high-confidence EUPLOTES subset. Training was performed under full supervision until early stopping with a patience of 10 was triggered (Supplementary Figure S2A). The model outputs a probability between 0 and 1 for each candidate frameshift site.

Stage 2. Initial BiLSTM-CNN model: An initial BiLSTM-CNN model was trained on the corresponding long-sequence features (399 nt) derived from the same training samples used for the HistGB model. This initial model was trained for 5 epochs and produced a baseline probability between 0 and 1 for each candidate sequence.

Stage 3. BiLSTM-CNN self-training: The initial BiLSTM-CNN model was then applied to the expanded EUPLOTES sampling pool as unlabeled data. Prediction uncertainty was quantified using entropy, as defined in Eq. (1).

$$Entropy\,(p) = -(p \cdot \log\,(p) + (1-p) \cdot \log\,(1-p)) \quad (1)$$

Samples with high prediction confidence (probability>0.7 and entropy<0.5) were selected from the expanded EUPLOTES pool and added to the training set for iterative retraining, with the aim of improving generalization.

If the initial model is trained for too few epochs, it is likely to underfit the labeled data, resulting in low-confidence and low-quality pseudo-labels for subsequent self-training. In contrast, training the initial model for too many epochs increases the risk of early overfitting, which reduces the benefit of incorporating newly selected samples during later self-training rounds. We therefore set the initial training length to 5 epochs for the BiLSTM-CNN model (Supplementary Figure S2B), as this model generally reaches strong baseline performance within 5 to 10 epochs. We did not apply self-training to the HistGB model. Compared with the BiLSTM-CNN model, HistGB showed weaker fitting performance on our training data, suggesting that its pseudo-labels would be less reliable. Under these conditions, self-training would be more likely to

propagate noise and introduce additional errors than to improve generalization.

After this initial training stage, the BiLSTM-CNN model entered the self-training phase. In each round, confidently predicted positive samples from the expanded EUPLOTES pool were added to the training set, and the model was retrained iteratively until no new samples were selected and early stopping (patience=5) was triggered (Supplementary Figure S2B).

The final semi-supervised BiLSTM-CNN model was obtained after the second round of self-training. Across five repeated runs, self-training consistently reduced training loss and improved validation loss, supporting the effectiveness of this iterative refinement strategy (Supplementary Figure S2C). Both the HistGB and BiLSTM-CNN models output a probability for each candidate frameshift site, representing the likelihood of PRF. The final prediction score was obtained by weighted averaging of the two model outputs.

Model ensemble fusion (final output)

For the final prediction, we use a decision-level fusion (weighted average) rather than feature concatenation. The probability outputs of the HistGB model (P_{HistGB}) and the Semi-BiLSTM-CNN model ($P_{BiLSTM\text{-}CNN}$) are combined as (2):

$$P_{Final} = 0.4 \times P_{HistGB} + 0.6 \times P_{BiLSTM-CNN} \quad (2)$$

Feature importance analysis

To comprehensively understand how machine learning models process mRNA sequence features and gain deeper insights into the underlying mechanisms of PRF, we implemented multiple feature importance analysis approaches. These methods were applied to our two primary models: HistGB model and the Bidirectional LSTM-CNN model.

Histogram-based gradient boosting model analysis

Permutation feature importance: To evaluate HistGB feature importance, we first employed permutation importance analysis, which quantifies the impact of features by measuring the change in model performance when feature values are randomly shuffled. The importance score for feature i is mathematically defined as (3):

$$Importance_i = \frac{1}{n}\sum_{j=1}^{n}\left[L\left(y, \mathcal{M}\left(X^{i}_{perm}\right)\right) - L\left(y, \mathcal{M}\left(X\right)\right)\right] \quad (3)$$

where X is the feature matrix, y is the target vector, L is the loss function, M is the trained model, X^{i}_{perm} is the feature matrix with feature i randomly permuted, and n is the number of permutation repetitions. A higher importance score indicates a more significant contribution of the feature to the model's predictions.

SHAP value analysis: To obtain a more robust feature importance assessment, we employed Shapley Additive Explanations (SHAP) value analysis based on game theory. SHAP values quantify feature importance by calculating the marginal contribution of features to model predictions, mathematically defined as (4):

$$\phi_i(f,x) = \sum_{S\subseteq N\setminus\{i\}} \frac{|S|!(|N|-|S|-1)!}{|N|!}\left[f_x(S\cup\{i\}) - f_x(S)\right] \quad (4)$$

where N is the set of all features, S is a feature subset excluding feature i, and $f_x(S)$ represents the model's prediction for sample x when using only the feature subset S.

Bidirectional LSTM-CNN model analysis

Integrated gradients analysis: For the BiLSTM-CNN model, we first applied the Integrated Gradients method, which attributes feature importance by integrating the gradient path from a baseline input to the actual input. The mathematical expression for Integrated Gradients is as (5):

$$IntegratedGrads_i(x) = \left(x_i - x'_i\right) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} d\alpha \quad (5)$$

where x is the input sample, x' is the baseline input (we use non-frameshifting sequence as baseline), F is the model function, and $IntegratedGrads_i(x)$ is the gradient of the model output with respect to input feature i. In practical computation, we used a discrete approximation as (6):

$$IntegratedGrads_i(x) \approx \left(x_i - x'_i\right) \times \sum_{k=1}^{m} \frac{\partial F\left(x' + \frac{k}{m}\left(x - x'\right)\right)}{\partial x_i} \times \frac{1}{m} \quad (6)$$

where m is the number of integration steps (set to 50 in our implementation). This method captures feature contributions in non-linear models and satisfies desirable properties such as sensitivity and implementation invariance.

Gradient-based saliency analysis: We also employed gradient-based saliency analysis, which directly calculates the magnitude of the gradient of the model output with respect to input features to determine which input regions are most sensitive to prediction. The calculation formula for gradient-based saliency is as (7):

$$Saliency(x) = \left|\frac{\partial F(x)}{\partial x}\right| \quad (7)$$

To enhance interpretability, we normalized the gradient magnitudes as (8):

$$NormalizedSaliency(x) = \frac{Saliency(x)}{\max(Saliency(x))} \quad (8)$$

This method is computationally efficient and intuitively demonstrates the model's sensitivity to different positions in the input sequence.

Codon conservation and divergence analysis

To quantify the conservation and unique features of codon usage around PRF sites, we employed two information-theoretic metrics: codon-level entropy and Kullback-Leibler (KL) divergence.

Codon-level entropy: We calculated the Shannon entropy for each codon position relative to the frameshift site across all aligned sequences within a given dataset. This metric measures the degree of codon conservation at each specific position. The entropy H for a position was calculated as (9):

$$H = -\sum_{i=1}^{64} p_i \log_2 p_i \quad (9)$$

where p_i is the frequency of the i-th codon (out of 64 possible codons) at that specific position. A lower entropy value indicates higher conservation (less variability), while a higher value signifies greater diversity of codons at that position.

Kullback-Leibler (KL) divergence: To measure how the codon distribution at frameshift sites deviates from a background distribution, we calculated the KL divergence. This metric quantifies the difference between the codon

frequency distribution at a specific position in the positive (frameshift) dataset (P) and the corresponding distribution in the negative (non-frameshift) dataset (Q). The KL divergence *D_KL* was calculated as (10):

$$D_{KL}(P \parallel Q) = \sum_{i=1}^{64} P(i) \log_2 \frac{P(i)}{Q(i)} \quad (10)$$

where *P(i)* is the frequency of the *i*-th codon at a given position in the frameshift sequences, and *Q(i)* is its frequency at the same relative position in the non-frameshift sequences. A high KL divergence value indicates that the codon usage at that position is distinct in frameshifting sequences compared to the background, highlighting positions with potentially significant functional roles.

RESULTS

Design and overview of FScanpy

FScanpy combines HistGB and BiLSTM-CNN models through weighted averaging within a semi-supervised learning framework designed to improve prediction accuracy and cross-species applicability. The workflow consists of three main steps. First, training data were collected from the RECODE, FSDB, SEAPHAGES, and EUPLOTES datasets, and 33 nt and 399 nt sequence windows centered on candidate frameshift sites were extracted as short- and long-sequence features, respectively. A high-confidence dataset was then defined through a semi-supervised framework integrating KNN-based data selection and entropy-guided self-training, and subsequently utilized for model optimization. Next, the 399 nt long-sequence features were used to train the BiLSTM-CNN model through initial supervised learning followed by semi-supervised refinement, whereas the 33 nt short-sequence features were used to train the HistGB model. Finally, the outputs of the two models were combined by weighted averaging to generate the final prediction (Figure 1; Supplementary Figure S3).

FScanpy supports three types of input. First, users can provide a cDNA or mRNA sequence and scan the full sequence for potential PRF sites, with the model returning the predicted probability at each position. Second, users can input a sequence region surrounding a presumed frameshift site and obtain the probability for that specific site. Third, FScanR can be used to generate candidate PRF sites from BLASTX results, after which FScanpy scores those candidates for each locus.

Preparation of PRF dataset for training and test

To prepare the training and test sets, high-confidence PRF data were obtained from the RECODE (Baranov et al., 2001), FSDB (Moon et al., 2007), SEAPHAGES (Russell & Hatfull, 2017), and EUPLOTES datasets (https://evan.ciliate.org). Negative samples were generated by randomly selecting a nucleotide as the FS_start from mRNA regions without annotated PRF events, but not actively filtering out or removing intrinsic slippery motifs from these negative sequences. The training set comprised 4 106 positive samples (9.4%) and 39 550 negative samples (90.6%). The sample counts per source are as follows: EUPLOTES (1 837 positive vs. 19 850 negative), SEAPHAGES (1 622 positive vs. 9 850 negative), FSDB (132 positive vs. 9 850 negative), and RECODE (515 positive vs. 0 negative) (Figure 2A). The validation and test set for training the models was constructed by randomly selecting a portion of positive and negative samples from each dataset (only positive samples were selected from RECODE, and all validation and test sets used high-confidence samples). The validation set consisted of 150 positive samples and 200 negative samples (Figure 2B). The test set comprised 400 positive samples (57.1%) and 300 negative samples (42.9%) (Figure 2C). We also visualized the structure and distribution of the data after dimensionality reduction using t-SNE, PCA, and UMAP (Figure 2D–F; Supplementary Figure S4).

To address potential concerns about data imbalance and model stability, we maintained the same data ratio and performed repeated resampling experiments with five independent random seeds to generate different data partitions for both HistGB and Semi-BiLSTM-CNN models. Performance metrics across these random splits exhibited minimal variation, with AUC showing a standard deviation of approximately 0.006–0.007 and Accuracy, F1, Precision, and Recall all varying within a narrow range (Supplementary Figure S5). These results confirm that the data split strategy and imbalance level have little influence on model robustness and the reported performance.

Development and optimization of FScanpy

Input: Data inputs comprised short sequence features using a 33 nt segment centered (11 codons) around the frameshift site and long sequence features using a 399 nt segment (133 codons). During sequence truncation, we ensured that the truncated sequence remained on the original 0-reading frame, and the frameshifting site was located at the center codon of the sequence. The short sequences focused on codon preference, the frameshift sequence itself, and amino acid polarity, with the latter referring to the charged amino acid located before the frameshift site (Bhatt et al., 2021; Kelly et al., 2021; Li et al., 2019). The long sequences took into account upstream specific sequences, downstream mRNA secondary structure, and the distance between stop and start codons (Anokhina & Miller, 2021).

Key components: The key component of FScanpy integrates machine learning models, combining a Histogram-based Gradient Boosting (HistGB) model and a BiLSTM-CNN model, and utilizes a weighted average method to output probabilities for more robust results. The HistGB model is designed to capture local codon preferences and sequence complexities. It takes a short mRNA sequence (11 codons centered on the frameshift site) as input and utilizes engineered features, including k-mer compositions and Shannon entropy, to predict frameshifting probability based on local sequence context. The BiLSTM-CNN model is designed to capture both local motifs and long-range structural constraints. It takes a longer mRNA sequence (133 codons) as input. In this framework, Convolutional Neural Networks (CNNs) are utilized to extract local motif features, while Bidirectional Long Short-Term Memory networks (BiLSTMs) capture long-range dependencies through bidirectional traversal. By concatenating the learned representations from both models through a weighted ensemble strategy, FScanpy achieves comprehensive feature extraction of the sequence data. Detailed feature engineering processes, network architectures, and optimization hyperparameters for both models are described in the MATERIALS AND METHODS section and Supplementary Figure S1.

Weighted Ensemble Strategy: For the final output, we weigh

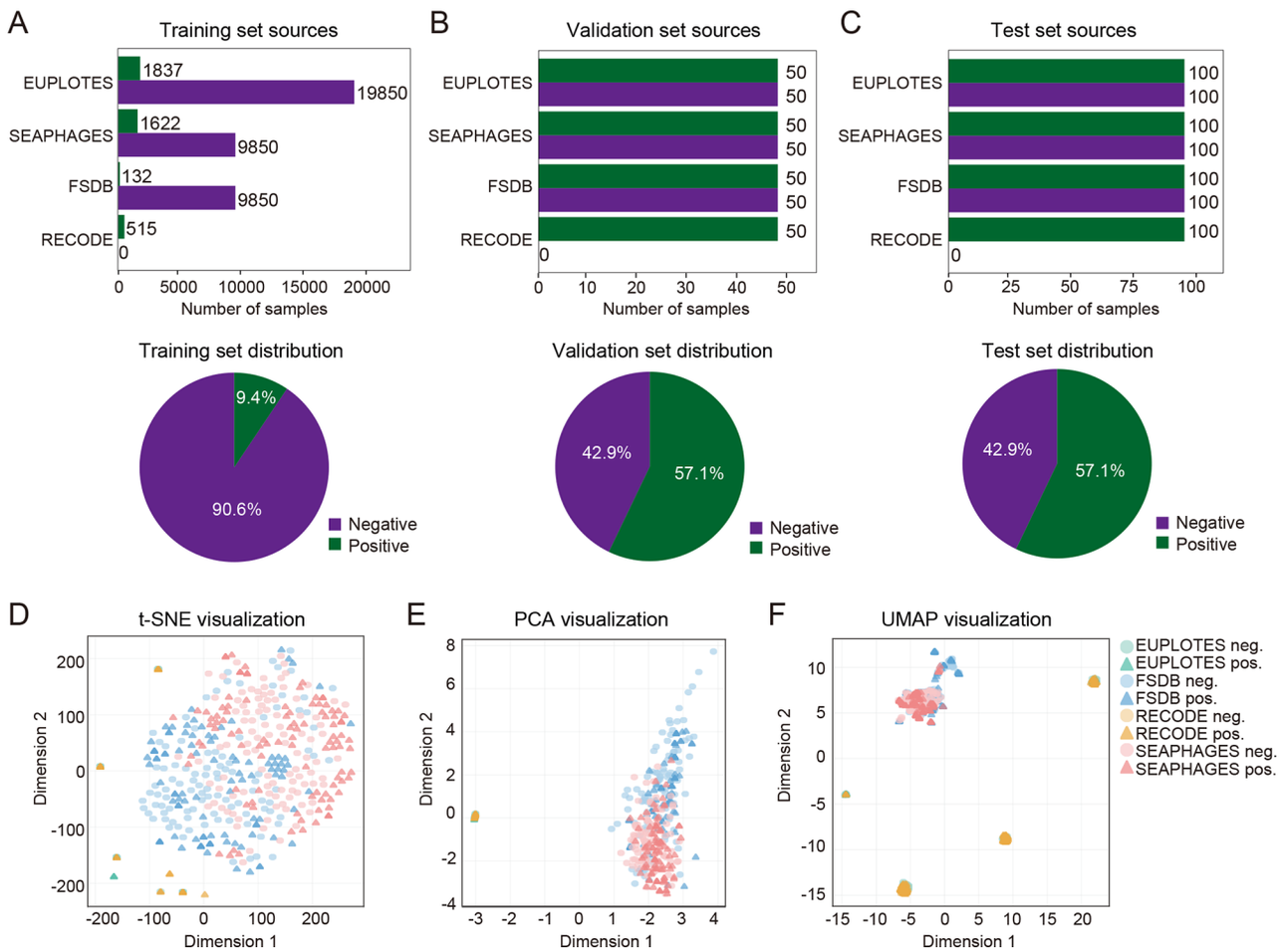

Figure 2 Data preparation for the training and test sets

A–C: Distribution of positive (green) and negative (purple) samples across the sources in the training (A), validation (B) and test (C) datasets. The RECODE dataset consists entirely of positive samples. D–F: Visualization of data distribution via t-SNE (D), PCA (E), and UMAP (F) dimensionality reduction.

the output probabilities of the HistGB and BiLSTM-CNN models. Analysis of the individual models revealed distinct and complementary strengths: the BiLSTM-CNN model demonstrates superior performance in identifying true positive samples, whereas the HistGB model is more effective at correctly recognizing true negative samples (Supplementary Figure S6A, B). Combining them by weighted averaging therefore produced a more balanced and stable prediction than either model alone. We set the default weight ratio to be a 4:6 ratio for HistGB and BiLSTM-CNN, respectively, because this setting achieved the highest AUC on the validation data while maintaining balanced predictive performance (Supplementary Figure S6C). This weighting reduces the influence of extreme predictions from the BiLSTM-CNN model while also compensating for the more conservative behavior of HistGB. FScanpy also allows users to adjust this ratio according to their analytical priorities. For example, users emphasizing broad candidate discovery may place more weight on BiLSTM-CNN, whereas users who wish to reduce false positives may increase the weight assigned to HistGB.

Semi-supervised learning: FScanpy uses semi-supervised learning to improve BiLSTM-CNN performance through graph-based KNN selection and self-training, whereas the HistGB model is trained statically on high-confidence data to preserve stability and avoid error propagation. The implementation proceeded as follows: (1) an initial model was trained on a small curated set of high-confidence samples (including selected expanded phage samples identified via KNN, see Data preprocessing section in MATERIALS AND METHODS); (2) this model was used to predict labels for the expanded sampling pool, with particular emphasis on the EUPLOTES dataset; (3) samples with sufficiently confident predictions were iteratively added to the training set for model refinement (see Model training section in MATERIALS AND METHODS); (4) in parallel, the HistGB model was trained only on high-confidence data to reduce overfitting and limit noise propagation. As a result, the semi-supervised approach enhanced overall model performance, achieving an AUC increase from 0.872 to 0.932 overall while demonstrating improvements across all sub-databases (Figure 3A). It significantly improved recall (an overall increase of 0.285, with a notably significant rise of 0.670 in SEAPHAGES), and F1 scores (an overall increase of 0.08, especially a 0.368 improvement in SEAPHAGES) (Figure 3B). These results indicate that iterative self-training expanded the effective training set while avoiding some limitations of purely static training. A detailed overview of the data processing pipeline is provided in Supplementary Figure S7.

The evaluation of FScanpy on test sets

Model reliability was evaluated using several standard metrics,

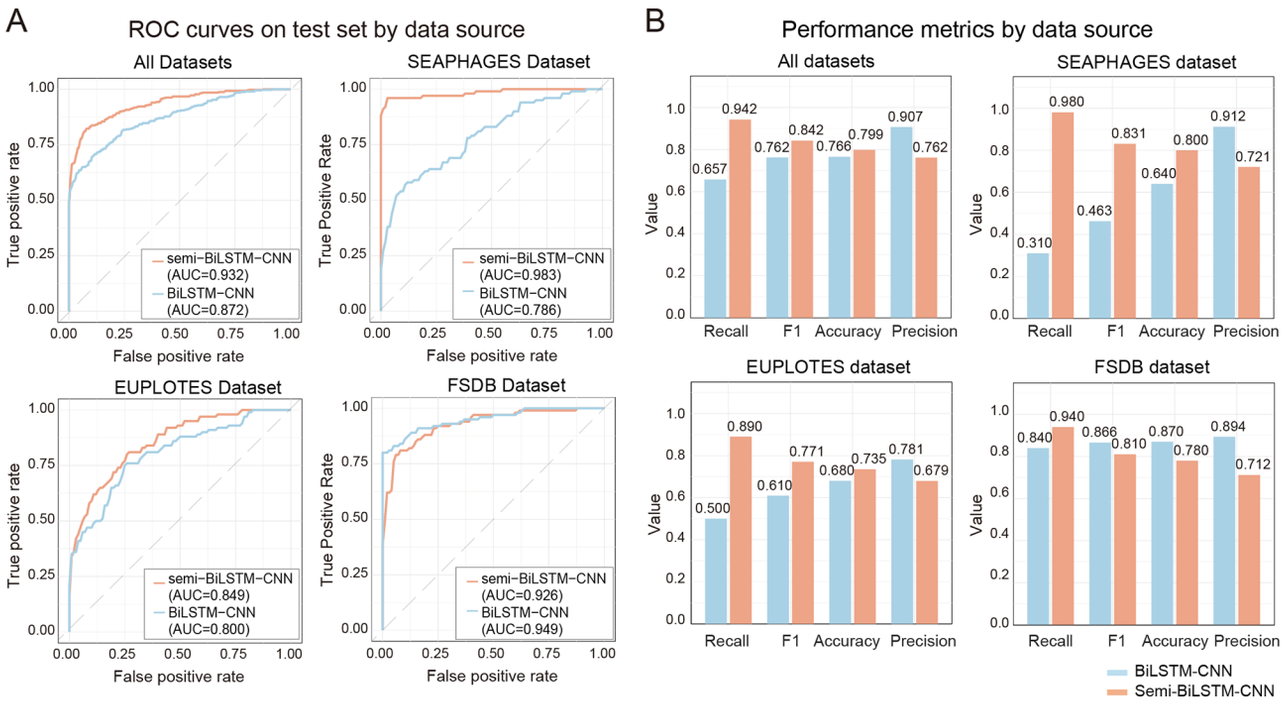

Figure 3 Model performance evaluation of fully supervised learning (BiLSTM-CNN) and semi-supervised learning (semi-BiLSTM-CNN) across different data sources

A: ROC curves of BiLSTM-CNN and semi-BiLSTM-CNN models on test sets containing all data sources and individual data sources (SEAPHAGES, EUPLOTES and FSDB database). B: Comparative performance metrics (Recall, F1 score, Accuracy and Precision) using BiLSTM-CNN and semi-BiLSTM-CNN models (denoted in blue and orange, respectively).

including accuracy, precision, recall, and F1 score. Collectively, these metrics indicated stable performance, although performance varied across datasets. The HistGB model was evaluated on the combined dataset (ALL, including EUPLOTES, FSDB, RECODE, and SEAPHAGES) as well as on each individual dataset. It performed better on the viral and prokaryotic datasets (FSDB and SEAPHAGES) than on the eukaryotic EUPLOTES dataset, with an overall AUC of 0.917 across all datasets (Figure 4A). Threshold analysis showed that precision increased consistently as the threshold increased across all datasets, whereas recall declined to varying degrees. This decline was smallest in SEAPHAGES and most pronounced in EUPLOTES. F1 score and accuracy fluctuated with threshold, but the magnitude of these changes was relatively small (Supplementary Figure S8A).

For the semi-BiLSTM-CNN model, the overall performance was superior to the HistGB model, with an overall AUC value of 0.932. The AUC values for EUPLOTES, SEAPHAGES, and FSDB are 0.849, 0.983, and 0.926, respectively (Figure 4B). The threshold analysis showed that precision gradually increased across all datasets as the threshold rose, while recall decreased slightly, with the most significant decline observed in the EUPLOTES dataset. F1 score and accuracy exhibited varying degrees of increase with the increase in the threshold, except for EUPLOTES and FSDB databases (Supplementary Figure S8B). This suggests that the semi-BiLSTM-CNN model performs less well in predicting PRF events in eukaryotes.

The weighted average model, which integrates the strengths of the HistGB and semi-BiLSTM-CNN models, further improved prediction performance and stability. Compared with either individual model, the weighted ensemble produced a higher AUC (0.951) together with more stable F1 score, accuracy, and recall. Its receiver operating characteristic (ROC) curve approached the upper-left region of the plot, indicating strong discrimination between positive and negative samples (Figure 4C). The balance between increasing precision and decreasing recall was also more favorable, with the F1 score remaining relatively high across thresholds, making the model suitable for applications that are sensitive to missed detections (Figure 4D). These results indicate that FScanpy provides a reliable framework for PRF prediction. Because threshold choice can strongly influence performance, especially in imbalanced datasets, FScanpy uses 0.5 as the default threshold while allowing users to adjust the threshold according to their specific needs.

To further evaluate the generalization capability of FScanpy on the challenging EUPLOTES dataset, we performed predictions on three *Euplotes* species, namely *E. aediculatus*, *E. woodruffi* and *E. crassus*. The model achieved AUC values of 0.801, 0.811, and 0.747 on these three species respectively (Supplementary Figure S9A). The threshold-metric analysis supported this finding, as while performance metrics for the EUPLOTES dataset used for training were consistently higher across most thresholds, other *Euplotes* species also exhibited robust F1-Score, Precision, and Accuracy curves (Supplementary Figure S9B). This indicates that FScanpy successfully learned generalizable features of PRF in the *Euplotes* genus, rather than simply overfitting to the specific sequence patterns of the species included in the training set. Additionally, we tested its performance on two additional datasets, Xu and Atkins, documenting PRF events in phages and viruses (Atkins et al., 2016; Xu et al., 2004). The results showed that the AUC values were 0.733 for the Xu dataset and 0.671 for the Atkins dataset (Supplementary Figure S9C). The threshold-metric analysis demonstrates that both the Xu and Atkins datasets exhibit favorable performance across Precision, F1-Score, and Accuracy curves (Supplementary

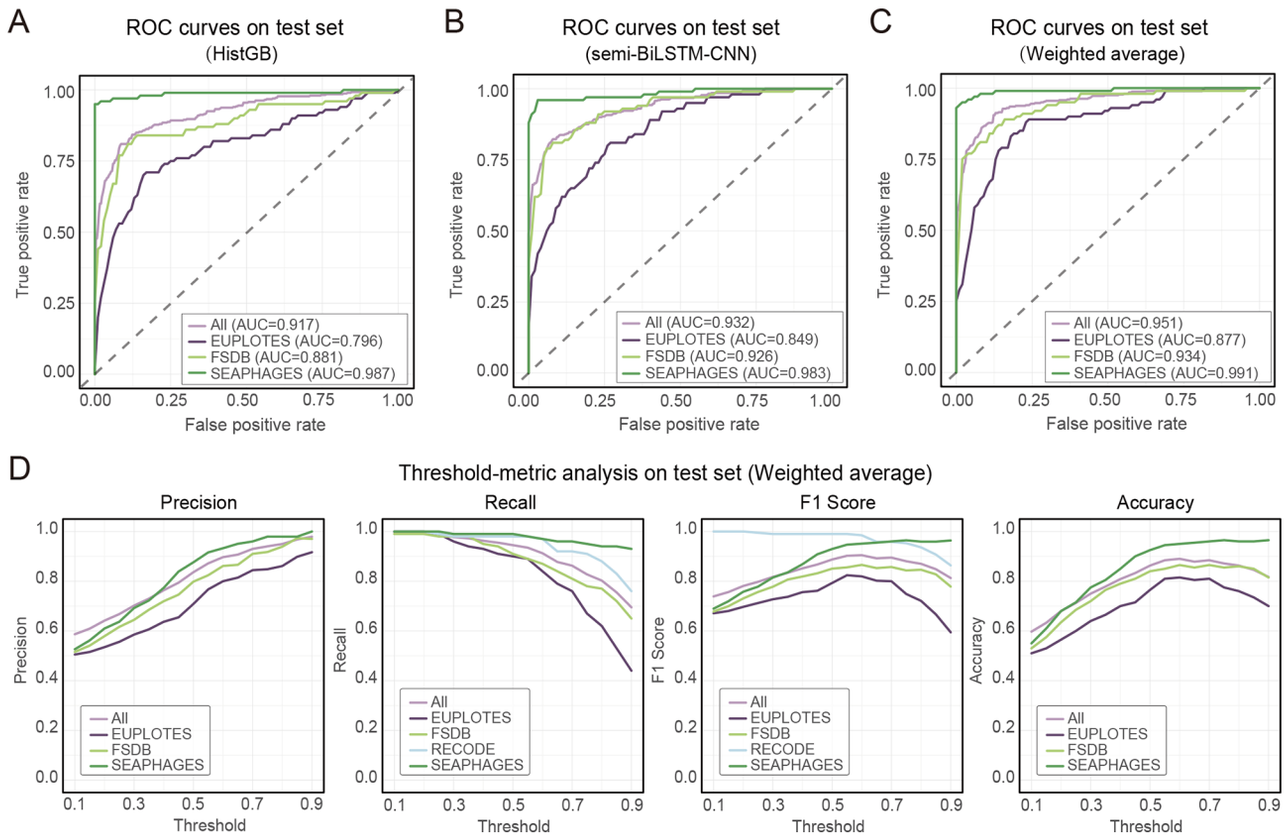

Figure 4 A hybrid neural network (weighted average) outperforms individual models (HistGB and semi-BiLSTM-CNN) based on ROC curves and threshold-metric analysis

A–C: ROC curves for the HistGB model (A), the semi-supervised BiLSTM-CNN model (B) and the weighted average ensemble model (C), evaluated on the test set containing all data sources (All) and on individual data sources (EUPLOTES, FSDB and SEAPHAGES). The specific AUC values for each model on various datasets are denoted at the bottom right corner. D: Threshold-metric analysis for the Weighted Average model, presenting the variations of Precision, Recall, F1 score and Accuracy with different classification thresholds to facilitate optimal threshold selection.

Figure S9D).

FScanpy has achieved better performance to date in predicting PRF sites

FScanpy outperformed existing PRF predictors (PRFect, KnotInFrame, FSfinder2) across multiple validation datasets, with clear improvements in metrics such as F1 score and Jaccard Index (Figure 5). At the default threshold of 0.5, FScanpy achieved high recall across diverse datasets (0.910 on FSDB; 0.643 on Xu; 0.545 on Atkins; 0.980 on RECODE). This translated into strong F1 scores, ranging from 0.594 to 0.990, and correspondingly high Jaccard index values, ranging from 0.423 to 0.980 (Figure 5A–C). For precision, FScanpy performed better than competing methods on FSDB and RECODE, but remained slightly below PRFect on the Xu and Atkins datasets. FScanpy also achieved F2 scores of 0.564–0.984 and Fowlkes-Mallows indices of 0.596–0.990, outperforming the other evaluated models (Figure 5D). Different threshold settings produced different trade-offs among evaluation metrics. Lower thresholds generally improved recall and F1/F2 scores by classifying more candidates as positive, whereas higher thresholds favored precision by applying stricter criteria. Detailed performance at thresholds of 0.2 and 0.8 is shown in Supplementary Figure S10. By comparison, PRFect showed relatively limited sensitivity, with recall values ranging from 0.084 to 0.357, which led to lower F1 scores (0.155–0.513) and Jaccard index values (0.084–0.345) (Figure 5B, C). Traditional methods performed less well overall: KnotInFrame showed very limited sensitivity across all datasets, with Jaccard index values never exceeding 0.038, whereas FSFinder2 produced F1 scores up to 0.473 and Jaccard index values below 0.309 (Figure 5B, C). Together, these comparisons indicate that FScanpy provides the most consistent balance of precision and recall across the evaluated datasets (Figure 5). As an illustrative example, we used FScanpy to identify PRF sites in two RefSeq genes (YP_010013633.1 and WP_166486895.1) (Figure 5G). These genes encode a tail assembly chaperone in mycobacteriophage Aziz and a peptide release factor 2 in *Rhizobium etli* CFN 42, respectively. Peaks in predicted frameshifting probability were detected at 336 nt and 69 nt, with probabilities of 0.852 and 0.980, respectively. Additional examples are shown in Supplementary Figure S11.

A systematic comparison of prediction mechanisms among FScanpy, PRFect, ORFeus, FSscan, FSFinder2 and KnotInFrame is provided in Table 1. KnotInFrame, which only requires DNA sequence input, is specialized for –1 PRF events in model organisms like *Saccharomyces cerevisiae*, but its utility is limited by its exclusive focus on –1 frameshifts. ORFeus accurately predicts all frameshift types by leveraging high-resolution Ribo-seq data alongside gene annotations and genomic sequences, but its reliance on such specialized data increases the barrier to entry. FSFinder2 handles multiple frameshift types with flexible input options, yet it struggles with predicting events in overlapping genes. PRFect is tailored for eukaryotic genomes and uses GenBank files directly, but its

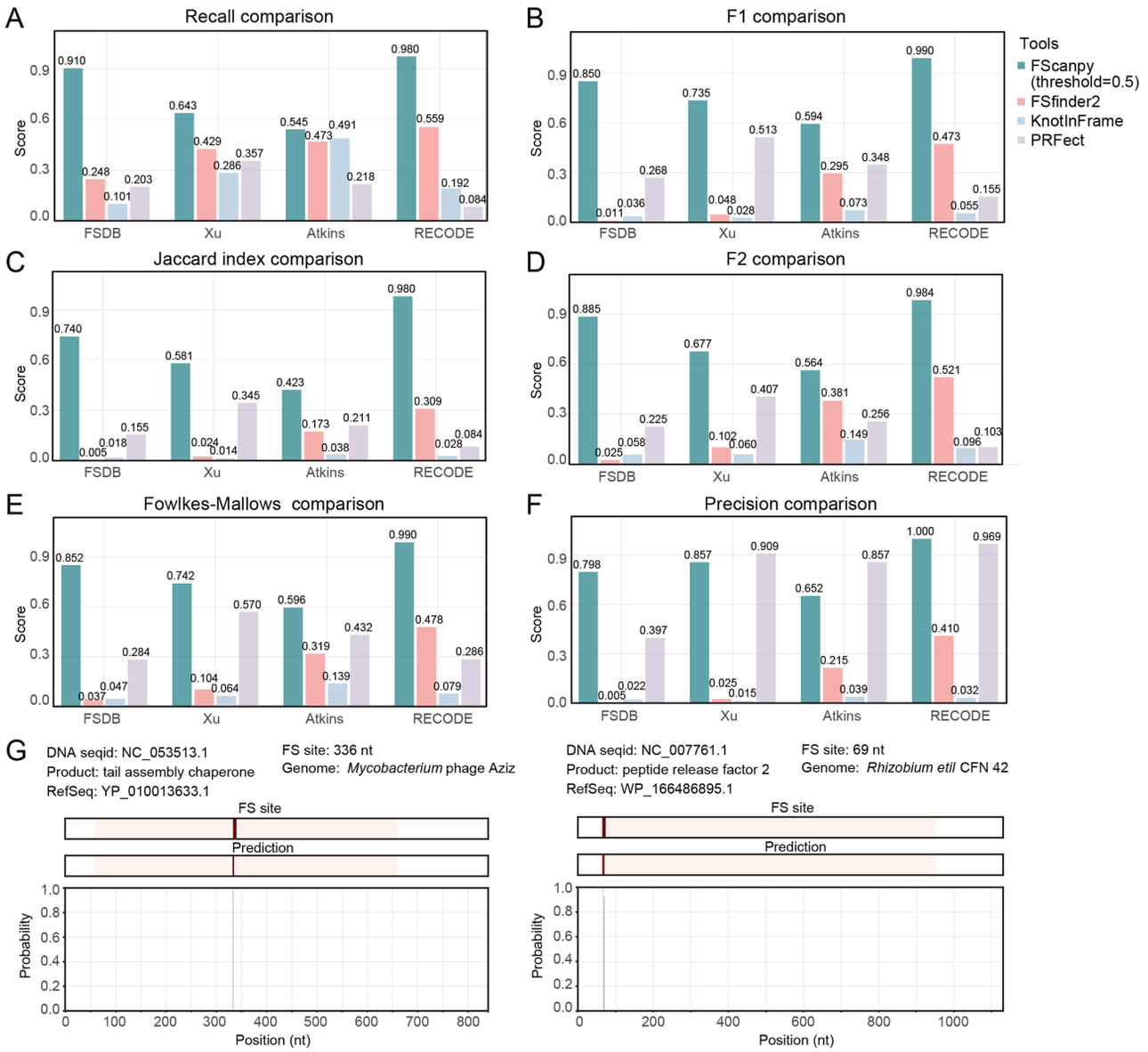

Figure 5 FScanpy outperforms existing methods in terms of predictive performance

A–F: Performance evaluation of prediction tools across four datasets (FSDB, Xu, Atkins, RECODE) visualized by metrics including recall, F1 score, Jaccard index, F2 score, Fowlkes-Mallows, and precision. FScanpy is represented with a default threshold of 0.5 (dark cyan). Other tools are denoted as FSFinder2 (pink), KnotInFrame (light blue) and PRFect (light purple). G: Two representative frameshift sites predicted by FScanpy, depicting the annotated FS site (dark red), predicted frameshift sites (red) and probability distributions along DNA sequences (YP_010013633.1 and WP_166486895.1).

Table 1 Comparison of the prediction mechanisms for programmed ribosomal frameshifting (PRF) events between FScanpy, PRFect, ORFeus, FSscan, FSFinder2, and KnotInFrame

Tools	Source data	Input data required	Scope of application	Limitations	Reference
FScanpy	SEAPHAGES, RECODE, FSDB, EUPLOTES	mRNA sequence or BLASTX result	All PRF in all species	PRF type not provided	The current work
KnotInFrame	RECODE, PseudoBase, *Saccharomyces cerevisiae*	DNA sequences	–1 PRF in all species	Limited to –1 ribosomal frameshift events	Theis et al. (2008)
ORFeus	RECODE (partial)	Ribo-seq data, genomic sequences and annotation	All PRF in all species	Rely on high-resolution Ribo-seq data	Richardson & Eddy (2023)
FSFinder2	–1, +1 PRF	DNA or mRNA sequences	–1, +1 PRF in all species	Difficult to predict overlapping genes	Byun et al. (2007)
PRFect	SEAPHAGES	Genome GenBank files	All PRF in eukaryotes mainly	Rely on high quality annotation	Mcnair et al. (2024)
FSscan	*Escherichia coli* XL1 blue MRF	mRNA sequences	+1 PRF in *Escherichia coli*	Code not available	Liao et al. (2009)

performance is heavily dependent on the availability of high-quality genome annotations. FSscan is specifically designed for +1 PRF in Escherichia coli XL1 blue MRF using mRNA sequences, but its practical application is severely restricted by the lack of publicly available code. FScanpy is versatile, supporting all frameshift types and accepting either mRNA sequences. Users can directly utilize the current model to predict frameshifting probabilities at PRF sites using FScanpy. Simply input a target sequence, and the model will output site-specific frameshifting probabilities.

For users whose main focus is –1 frameshifting, KnotInFrame may still be useful because it is tailored to –1-specific features. If Ribo-seq data are available, ORFeus is also an appropriate choice because it combines model-based prediction with experimental evidence. FSFinder2, PRFect, and FScanpy can all be used to predict multiple classes of frameshifting events, with FScanpy showing the most consistent performance across the evaluated datasets.

FScanpy interprets the significant features of PRF sites

HistGB: The Histogram-based Gradient Boosting component also makes FScanpy more interpretable. To better understand the learned representations, we evaluated the importance of different feature classes using two complementary approaches: permutation feature importance and SHAP analysis (Figure 6A, B; Supplementary Table S1, S2). Both approaches consistently ranked GGG, AAA, mfe_120 nt (minimum free energy of the 120 nt sequence downstream of the frameshift site), FS –1 codon AAA, and ATA among the most important features, although their exact order differed slightly. GGG was the top-ranked feature in both analyses, indicating that it had a broad influence on model predictions. The repeated appearance of GGG, FS –1 codon AAA, and AAA across both methods suggests that the model relies on a combination of codon-position signals (e.g., FS –1 codon AAA) and local nucleotide composition (e.g., GGG/AAA) when identifying candidate frameshift sites. Feature-importance patterns stratified by data source are presented in Supplementary Figure S12 (details are provided in

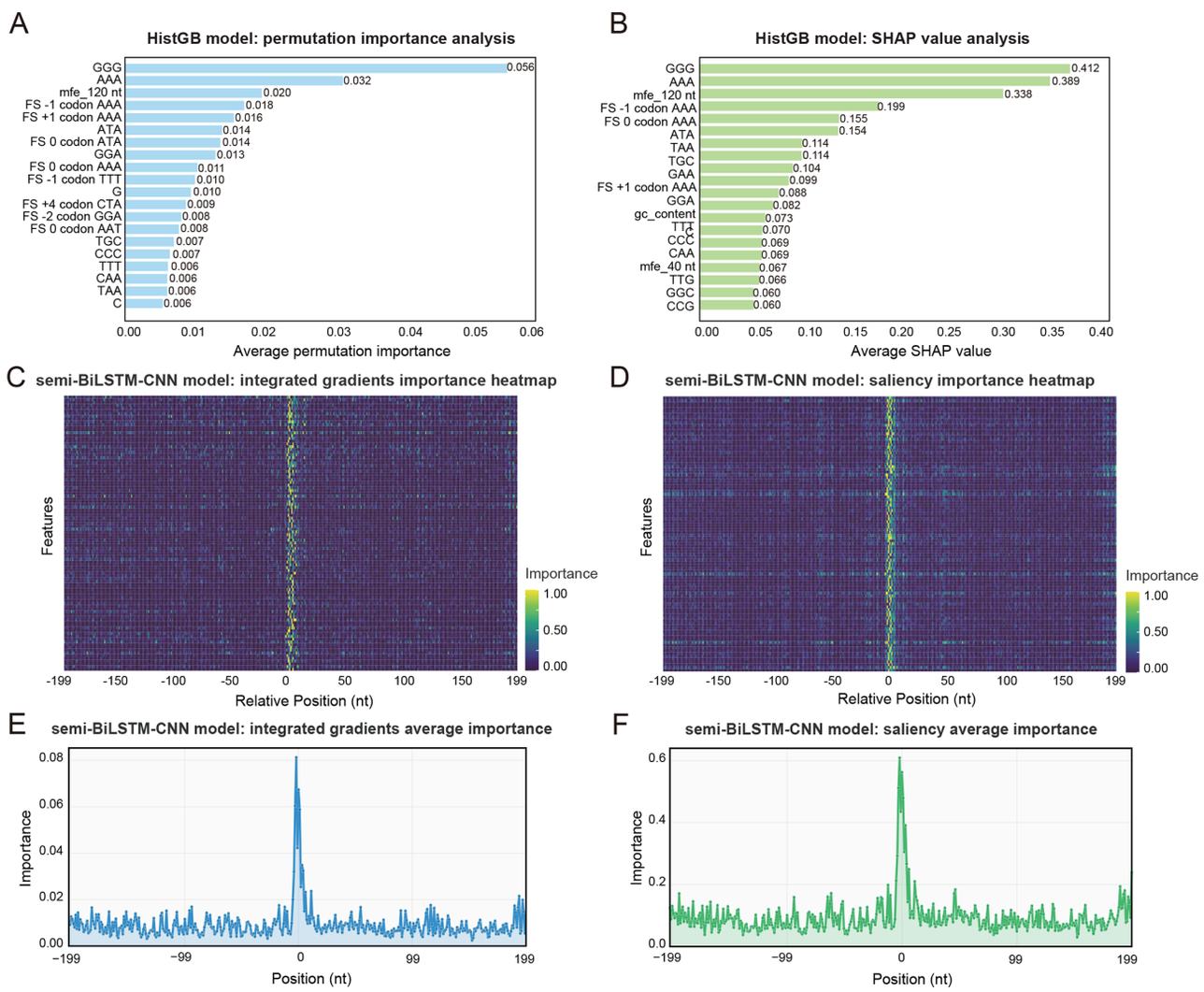

Figure 6 The feature importance analysis reveals sequence motifs and downstream mRNA secondary structures as the dominant factors in PRF events

A: Overall feature importance derived through permutation importance analysis, which quantifies the average importance of individual features. B: Overall feature importance derived through SHAP value analysis, illustrating the magnitude of feature contributions to predictive outputs. C, D: Heatmaps depicting feature importance for the semi-BiLSTM-CNN model, generated via Integrated Gradients analysis (C) and Gradient Saliency analysis (D). E, F: Average importance profiles derived from Integrated Gradients analysis (E) and Gradient Saliency analysis (F), respectively, showing the position-wise distribution of feature importance.

Supplementary Table S1, S2). In SEAPHAGES, both methods highlighted GGG as a prominent contributor, whereas AAA was more influential in RECODE. In EUPLOTES, AAA, TAA, and the FS 0 codon consistently ranked among the most important features. In FSDB, AAA, mfe_120 nt, GGG, and FS +1 codon AAA were among the strongest contributors, indicating their broad relevance to frameshifting prediction.

Semi-BiLSTM-CNN: To examine feature importance in the semi-BiLSTM-CNN model, we applied Integrated Gradients and Gradient Saliency analyses. In both heatmaps, the sequence region from relative positions –10 to +10 showed a pronounced bright-yellow signal, indicating strong contribution to model predictions and suggesting that the model is particularly sensitive to slippery sequences and nearby regulatory elements (Figure 6C, D; Supplementary Table S3). The average importance curves provided a quantitative view of this pattern: both methods showed distinct peaks around relative positions ±10, consistent with increased model sensitivity near the middle of the sequence and at terminal regions. Although the overall trends were similar, the Saliency method displayed a localized spike at distal positions (e.g., relative distance>151 nt), which may reflect gradient-based sensitivity to noise or edge effects (Figure 6E, F). Together, these heatmaps and importance curves suggest that the model focuses on restricted functional regions of the input sequence, offering a useful framework for interpreting how the deep-learning component distinguishes candidate PRF signals.

DISCUSSION

FScanpy addresses key challenges in the computational prediction of PRF sites

PRF is an important translational regulatory mechanism with major consequences for gene expression in viruses and some eukaryotes. Many viruses, including SARS-CoV-2, rely on PRF to complete their life cycles (Lee et al., 2025). At the same time, the diversity of PRF types and the mechanistic complexity of frameshifting make accurate prediction challenging. We developed FScanpy to address this problem using a semi-supervised framework that can operate even when annotated data are limited. In our analyses, the model showed strong performance, stability, and cross-species applicability. A central contribution of this work lies in the combination of data curation, semi-supervised sample expansion, and model integration. The overall workflow first uses high-confidence data to train an initial model and then refines that model through iterative self-training. Users can provide cDNA or mRNA sequences, or candidate frameshift sites, and obtain predicted frameshifting probabilities for downstream analysis.

The components of FScanpy include a semi-supervised model, a HistGB model, and a BiLSTM-CNN model, and their advantages are integrated through a weighted average method. The HistGB model has good performance on structured data, capability in handling nonlinear relationships, and relatively low requirements for feature engineering. PRF events often occur in specific slippery sequences and depend on specific upstream and downstream signatures, such as slippery sequence features, codon preference, and amino acid polarity. The HistGB model's ability to learn and extract short-sequence features enhances the prediction accuracy of sequence features near the slippery sequence. In addition, the occurrence of PRF events is also closely related to upstream and downstream signatures, such as downstream mRNA secondary structures (pseudoknots or stem-loops). The BiLSTM-CNN model excels in efficient feature extraction and strong sequence modeling capabilities, especially for long-sequence feature extraction. Through character-level encoding and sequence padding, the sequence is converted into an integer sequence using a Tokenizer. Sequence padding is performed during the extraction of the frameshifting region, ensuring that the frameshifting site is at the center of the sequence. This retains the distance features between the frameshifting site and the start or stop codons, allowing for the analysis and recognition of long-range sequence features upstream and downstream of the frameshifting site. Recent investigations have underscored the sequence dependency of frameshift features, demonstrating that variations in base composition within identical frameshift features can markedly influence frameshift efficiency (Mathew et al., 2015). Building on these insights, the weighted average method synergistically integrates the two models, incorporating both local codon patterns and global sequence contexts. This approach addresses the inherent complexity of frameshift events by capturing the nuanced sequence dependencies that modulate programmed ribosomal frameshifts (PRFs), thereby enhancing the model's capacity to discern the intricate interplay between sequence elements and structural features that govern PRF efficiency, resulting in a more stable AUC value, improved F1 score and accuracy, and less noticeable performance fluctuations. The overall prediction performance is significantly enhanced and more stable. Compared to existing PRF prediction models, FScanpy achieves an AUC value of 0.951 and demonstrates excellent performance across multiple datasets, indicating its strong generalization capability.

Compared with other PRF prediction algorithms, we provided an alignment-based method, FScanR (https://github.com/seanchen607/FScanR), for recognizing the most possible PRF sites in the genomes of any species including the eukaryotic ciliate *Euplotes*. *Euplotes vannus* and *E. octocarinatus* exhibit a high frequency of +1 PRF events, and the traditional stop codon UGA is reassigned to cysteine while only UAA and UAG are used as stop codons (Kervestin et al., 2001; Lozupone et al., 2001). FScanR identifies potential PRF sites through BLAST alignment between the DNA/cDNA and peptide sequences from the same species or its close relatives. Given the genomic and proteomic sequences of a particular species, users can derive the preliminary output of FScanR and subsequently utilize these results as input for FScanpy. The ROC curve achieves an AUC of 0.78, indicating strong discriminative ability (Supplementary Figure S13).

Heterogeneity across training sets aids in the precise identification of frameshift sites

We further compared frameshift and non-frameshift features across databases using codon-based entropy (red line) and Kullback-Leibler divergence (green line) (Supplementary Figure S14). Both metrics showed that forward frameshifting (+1/+2) in EUPLOTES produced clear peaks near the relative frameshift site. In contrast, reverse frameshifting (–1/–2) in EUPLOTES showed more frequent fluctuations in conservation scores and lacked a similarly concentrated peak in codon information content. For FSDB and SEAPHAGES

(with FSDB as a representative example), the distributions of conservation scores and entropy for forward frameshifting were more dispersed. By contrast, reverse frameshifting showed sharper increases at specific positions, together with corresponding peaks in information content, a pattern that may be consistent with the predominance of reverse frameshifting annotations in these datasets. These dataset-specific differences in codon conservation and information content provide a useful basis for exploring variation in frameshifting mechanisms across taxa. More detailed usage instructions are available on the GitHub page (https://github.com/ykongxiang/FScanpy-package/blob/master/tutorial%2Ftutorial.md). This variability in PRF direction underscores the value of diverse training datasets and helps explain why shift-site identification remains difficult across distantly related organisms. To facilitate such comparisons, we provide rank-ordered PRF feature lists from multiple data sources, offering a more accurate and species-aware reference for identifying frameshift determinants across organisms (Supplementary Figure S12).

FScanpy sheds light on the mechanisms of PRF

FScanpy integrates local codon patterns with global sequence background characteristics, thereby enabling it to discern the subtle sequence dependencies that affect PRF. This integration facilitates a more nuanced understanding of the PRF mechanism. Through comprehensive feature analysis of the FScanpy model using diverse methodologies, we demonstrated that nucleotide motifs and codon positions critically influenced PRF. GGG-rich motifs have been known to interact with slippery sequences (e.g., X XXY YYZ) in HIV to induce frameshifting (Mathew et al., 2015). In our analysis of marine phages, GGG-rich motifs were also identified as the highest contributing feature, which is consistent with the previous report (Mathew et al., 2015). As a highly conserved intercodon, mutations in GGG can disrupt the stem-loop structure and affect frameshifting efficiency. In addition, GGG also has the functions of maintaining the stem-loop structure and regulating reverse transcriptase (RT) slippage (Cardno et al., 2015; Penno et al., 2017). In our results, GGG showed a high contribution in the results of both feature analyses, which also illustrates the importance of this feature for PRF formation. Similarly, the AAA codon (encoding lysine) dominates across datasets, likely due to lysine's charged side chain affecting ribosome-EF-Tu-tRNA kinetics, thereby causing translational pausing or frameshifting (Condron et al., 1991). The TTT codon (encoding phenylalanine), frequently observed in slippery sequences (e.g., UUU UUA), may directly promote frameshifting via tRNAPhe slippage, as reported in coronaviruses (Liao et al., 2009). The AAT-associated frameshifting mechanism could involve codon usage bias (e.g., amino acid starvation) or local mRNA structure (Türkel, 2016).

Minimum free energy (MFE) structures, exemplified by the mfe_120 nt feature in our results, further modulate PRF efficiency, aligning with RNA structural stability principles (Tinoco Jr & Bustamante, 1999). Mutating the sequence downstream of the frameshift site substantially reduces frameshifting frequency, and the relationship between downstream sequences and ribosomal frameshifting likely operates by altering the kinetics of how the ribosome encounters the slippery site (Mouzakis et al., 2013). Another study used thermodynamic methods including MFE to predict the most stable structure of coronavirus frameshifting pseudoknots, demonstrating a direct correlation between pseudoknot stability and frameshifting efficiency (Trinity et al., 2024). In our SHAP analysis, mfe_120 nt and mfe_40 nt ranked among the 10 most important features, and both contributed strongly across multiple datasets. Lower MFE values correspond to more stable RNA secondary structures, such as pseudoknots and stem-loops downstream of the frameshift site. In many reported systems, these stable structures can promote frameshifting by slowing ribosome progression and increasing the opportunity for slippage at the slippery sequence. The relationship between coronavirus pseudoknot stability and frameshifting provides one example of this broader principle (Trinity et al., 2024).

As a quantitative indicator of structural stability, the MFE feature can capture information that sequence features alone cannot reflect. For example, identical sequences may form different structures due to variations in MFE, which in turn affect frameshifting efficiency. By integrating MFE features (e.g., mfe_120 nt) with sequence features, the model can more comprehensively cover the multi-dimensional factors regulating frameshifting, avoiding prediction biases caused by relying solely on sequence features. The MFE feature exhibits high contribution to the model, which is consistent with the known biological mechanism that RNA structure regulates frameshifting. Across different biological datasets (RECODE, SEAPHAGES, FSDB, and EUPLOTES), the impact of MFE on frameshifting is very consistent, ranking among the top 20 important features in SHAP and permutation feature importance analysis. This indicates that the model can stably identify frameshifting-related structural signals across different systems, demonstrating excellent stability.

By integrating our feature analysis results, we systematically revealed the pivotal roles of sequence and structural features, such as the AAA and GGG motifs, and minimum free energy, in PRF events. We also highlighted the widespread conservation of these features across viruses, prokaryotes and eukaryotes. The database-specific presentations provide researchers with cross-species, multi-source references for frameshift features, thereby enhancing the interpretability and applicability of the predictive model. These findings validate the accuracy of FScanpy in capturing key biological signals and deepen our understanding of the evolutionary conservation and functional significance of PRF mechanisms across the tree of life. This work provides a foundation for future research into the regulatory logic and biological implications of PRF in diverse species.

Limitations and perspectives

Despite the demonstrated capabilities, FScanpy exhibits several inherent limitations when used to analyze PRF events. The initial training of FScanpy relies on high-confidence data, thus in sparsely annotated species FScanpy may have limited performance. Additionally, while the BiLSTM-CNN model excels at long-sequence feature extraction, its computational complexity may hinder scalability for large genomes. The heterogeneity of PRF mechanisms across species also poses challenges, as FScanpy's training data may not fully capture all regulatory paradigms. Finally, although sequence alignment-based approaches are widely used to identify PRF-positive samples, such predictions lack experimental validation and the results likely suffer from inherent circularity.

To address the risk of overstating performance, given that

PRF events are rare in real-world, transcriptome-wide applications, users should utilize the adjustable output threshold when deploying FScanpy for genome-wide scans. This will prioritize precision and filter out the inevitable larger volume of false positives more stringently. Future work will focus on expanding FScanpy's training dataset to include more species, improving computational efficiency through model optimization, and incorporating real-time ribosome profiling data to validate predictions. Integrating multi-omics data (e.g., transcriptomics, proteomics, Ribo-seq) could potentially enhance FScanpy's ability to model complex PRF regulatory networks. Additionally, developing a user-friendly interface for FScanpy will broaden its accessibility to non-computational biologists.

CONCLUSIONS

In this study, we present FScanpy for cross-species PRF prediction. By combining short-range codon features with longer-range sequence context and applying semi-supervised learning to expanded sampling sequences, the framework reduces annotation dependence and improves generalization. On the evaluated datasets, the model achieved an AUC of 0.951 and also performed well in non-model organisms such as *Euplotes* (AUC = 0.877). FScanpy expands the taxonomic scope of PRF analysis, particularly in systems with abundant PRF sites, and provides interpretable feature analyses that quantify the relative contributions of candidate frameshift determinants. The high-ranking AAA and GGG motifs, together with minimum free energy, support the importance of conserved sequence and structural signals associated with PRF sites from viruses to higher eukaryotes. In combination with FScanR, FScanpy enables large-scale PRF prediction, identifies candidate frameshift sites, and helps prioritize high-confidence recoding events for experimental validation.

CODE AVAILABILITY

The FScanpy Python package and example data are available at GitHub: https://github.com/ykongxiang/FScanpy-package. The repository includes a README with installation and usage instructions, and a detailed tutorial is available at https://github.com/ykongxiang/FScanpy-package/blob/master/tutorial%2Ftutorial.md.

COMPETING INTERESTS

The authors declare that they have no competing interests.

AUTHORS' CONTRIBUTIONS

Y.H.Y., J.Y., and X.C.: conceptualization; Y.H.Y., J.Y., Z.J.L., Y.L., W.B.S., N.A.S. and X.C.: investigation; Y.H.Y., J.Y., and X.C.: methodology, software; Y.H.Y., J.Y., and X.C.: writing–original draft preparation. All authors read and approved the final version of the manuscript.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Dr. Ruanlin Wang (Shanxi University) for his kind help during the data preparation of the genomic data of *Euplotes aediculatus* and *E. octocarinatus*. The authors also acknowledge the computing resources provided on the SDU-LMPBE, a high-performance computing cluster operated by the Laboratory of Marine Protozoan Biodiversity and Evolution, Shandong University.

REFERENCES

Aigner S, Lingner J, Goodrich KJ, et al. 2000. *Euplotes* telomerase contains an La motif protein produced by apparent translational frameshifting. *The EMBO Journal*, **19**(22): 6230–6239.

Anokhina VS, Miller BL. 2021. Targeting ribosomal frameshifting as an antiviral strategy: from HIV-1 to SARS-CoV-2. *Accounts of Chemical Research*, **54**(17): 3349–3361.

Atkins JF, Loughran G, Bhatt PR, et al. 2016. Ribosomal frameshifting and transcriptional slippage: From genetic steganography and cryptography to adventitious use. *Nucleic Acids Research*, **44**(15): 7007–7078.

Baranov PV, Gurvich OL, Fayet O, et al. 2001. RECODE: a database of frameshifting, bypassing and codon redefinition utilized for gene expression. *Nucleic Acids Research*, **29**(1): 264–267.

Bhatt PR, Scaiola A, Loughran G, et al. 2021. Structural basis of ribosomal frameshifting during translation of the SARS-CoV-2 RNA genome. *Science*, **372**(6548): 1306–1313.

Brierley I. 1995. Ribosomal frameshifting viral RNAs. *Journal of General Virology*, **76**(Pt 8): 1885–1892.

Brierley I, Digard P, Inglis SC. 1989. Characterization of an efficient coronavirus ribosomal frameshifting signal: requirement for an RNA pseudoknot. *Cell*, **57**(4): 537–547.

Byun Y, Moon S, Han K. 2007. A general computational model for predicting ribosomal frameshifts in genome sequences. *Computers in Biology and Medicine*, **37**(12): 1796–1801.

Cardno TS, Shimaki Y, Sleebs BE, et al. 2015. HIV-1 and human *PEG10* frameshift elements are functionally distinct and distinguished by novel small molecule modulators. *PLoS One*, **10**(10): e0139036.

Chen J, Petrov A, Johansson M, et al. 2014. Dynamic pathways of -1 translational frameshifting. *Nature*, **512**(7514): 328–332.

Chen X, Jiang YH, Gao F, et al. 2019. Genome analyses of the new model protist *Euplotes vannus* focusing on genome rearrangement and resistance to environmental stressors. *Molecular Ecology Resources*, **19**(5): 1292–1308.

Condron BG, Gesteland RF, Atkins JF. 1991. An analysis of sequences stimulating frameshifting in the decoding of gene 10 of bacteriophage T7. *Nucleic Acids Research*, **19**(20): 5607–5612.

Doak TG, Witherspoon DJ, Jahn CL, et al. 2003. Selection on the genes of *Euplotes crassus* Tec1 and Tec2 transposons: evolutionary appearance of a programmed frameshift in a Tec2 gene encoding a tyrosine family site-specific recombinase. *Eukaryotic Cell*, **2**(1): 95–102.

Feng Y, Neme R, Beh LY, et al. 2022. Comparative genomics reveals insight into the evolutionary origin of massively scrambled genomes. *eLife*, **11**: e82979.

Fenton DA, Bożko M, Świrski MI, et al. 2025. Comprehensive analysis of yeast +1 ribosomal frameshifting unveils a novel stimulator supporting two distinct frameshifting mechanisms. *Nucleic Acids Research*, **53**(22): gkaf1301.

Gao F, Bai Y, Chi Y, et al. 2025. Current status of phylogenetic studies on ciliated protists (Alveolata, Protozoa, Ciliophora) by the OUC-group: advances, challenges and future perspectives. *Marine Life Science & Technology*, **7**(4): 643–669.

Giedroc DP, Theimer CA, Nixon PL. 2000. Structure, stability and function of RNA pseudoknots involved in stimulating ribosomal frameshifting. *Journal of Molecular Biology*, **298**(2): 167–185.

Hansen TM, Reihani SN, Oddershede LB, et al. 2007. Correlation between mechanical strength of messenger RNA pseudoknots and ribosomal frameshifting. *Proceedings of the National Academy of Sciences of the United States of America*, **104**(14): 5830–5835.

Hao TT, Su H, Quan ZJ, et al. 2025. Distinct evolutionary origins and mixed-mode transmissions of methanogenic endosymbionts are revealed in anaerobic ciliated protists. *Marine Life Science & Technology*, **7**(4): 700–716.

Jahn CL, Doktor SZ, Frels JS, et al. 1993. Structures of the *Euplotes crassus Tec1* and *Tec2* elements: identification of putative transposase coding regions. *Gene*, **133**(1): 71–78.

Jiang YH, Chen X, Wang CD, et al. 2025. Genes and proteins expressed at different life cycle stages in the model protist *Euplotes vannus* revealed by both transcriptomic and proteomic approaches. *Science China Life*

Sciences, **68**(1): 232–248.

Jin DD, Li C, Chen X, et al. 2023. Comparative genome analysis of three euplotid protists provides insights into the evolution of nanochromosomes in unicellular eukaryotic organisms. *Marine Life Science & Technology*, **5**(3): 300–315.

Kelly JA, Woodside MT, Dinman JD. 2021. Programmed -1 ribosomal frameshifting in coronaviruses: a therapeutic target. *Virology*, **554**: 75–82.

Kervestin S, Frolova L, Kisselev L, et al. 2001. Stop codon recognition in ciliates: *Euplotes* release factor does not respond to reassigned UGA codon. *EMBO Reports*, **2**(8): 680–684.

Korniy N, Samatova E, Anokhina MM, et al. 2019. Mechanisms and biomedical implications of -1 programmed ribosome frameshifting on viral and bacterial mRNAs. *FEBS Letters*, **593**(13): 1468–1482.

Kramer EB, Farabaugh PJ. 2007. The frequency of translational misreading errors in *E. coli* is largely determined by tRNA competition. *RNA*, **13**(1): 87–96.

Kurland CG. 1992. Translational accuracy and the fitness of bacteria. *Annual Review of Genetics*, **26**(1): 29–50.

Lee S, Yan ST, Dey A, et al. 2025. A cascade of conformational switches in SARS-CoV-2 frameshifting: coregulation by upstream and downstream elements. *Biochemistry*, **64**(4): 953–966.

Li YH, Firth AE, Brierley I, et al. 2019. Programmed -2/-1 ribosomal frameshifting in simarteriviruses: an evolutionarily conserved mechanism. *Journal of Virology*, **93**(16): e00370–19.

Liao PY, Choi YS, Lee KH. 2009. FSscan: a mechanism-based program to identify +1 ribosomal frameshift hotspots. *Nucleic Acids Research*, **37**(21): 7302–7311.

Lobanov AV, Heaphy SM, Turanov AA, et al. 2017. Position-dependent termination and widespread obligatory frameshifting in *Euplotes* translation. *Nature Structural & Molecular Biology*, **24**(1): 61–68.

Lozupone CA, Knight RD, Landweber LF. 2001. The molecular basis of nuclear genetic code change in ciliates. *Current Biology*, **11**(2): 65–74.

Lyu L, Zhang X, Gao YY, et al. 2024. From germline genome to highly fragmented somatic genome: genome-wide DNA rearrangement during the sexual process in ciliated protists. *Marine Life Science & Technology*, **6**(1): 31–49.

Mathew SF, Crowe-Mcauliffe C, Graves R, et al. 2015. The highly conserved codon following the slippery sequence supports -1 frameshift efficiency at the HIV-1 frameshift site. *PLoS One*, **10**(3): e0122176.

Matsufuji S, Matsufuji T, Miyazaki Y, et al. 1995. Autoregulatory frameshifting in decoding mammalian ornithine decarboxylase antizyme. *Cell*, **80**(1): 51–60.

Mcnair K, Salamon P, Edwards RA, et al. 2024. PRFect: a tool to predict programmed ribosomal frameshifts in prokaryotic and viral genomes. *BMC Bioinformatics*, **25**(1): 82.

Meyer F, Schmidt HJ, Plümper E, et al. 1991. UGA is translated as cysteine in pheromone 3 of *Euplotes octocarinatus*. *Proceedings of the National Academy of Sciences of the United States of America*, **88**(9): 3758–3761.

Mikl M, Pilpel Y, Segal E. 2020. High-throughput interrogation of programmed ribosomal frameshifting in human cells. *Nature Communications*, **11**(1): 3061.

Möllenbeck M, Gavin MC, Klobutcher LA. 2004. Evolution of programmed ribosomal frameshifting in the TERT genes of *Euplotes*. *Journal of Molecular Evolution*, **58**(6): 701–711.

Moon S, Byun Y, Han K. 2007. FSDB: a frameshift signal database. *Computational Biology and Chemistry*, **31**(4): 298–302.

Mouzakis KD, Lang AL, Vander Meulen KA, et al. 2013. HIV-1 frameshift efficiency is primarily determined by the stability of base pairs positioned at the mRNA entrance channel of the ribosome. *Nucleic Acids Research*, **41**(3): 1901–1913.

Penno C, Kumari R, Baranov PV, et al. 2017. Specific reverse transcriptase slippage at the HIV ribosomal frameshift sequence: potential implications for modulation of GagPol synthesis. *Nucleic Acids Research*, **45**(17): 10156–10167.

Richardson MO, Eddy SR. 2023. ORFeus: a computational method to detect programmed ribosomal frameshifts and other non-canonical translation events. *BMC Bioinformatics*, **24**(1): 471.

Rodnina MV. 2012. Quality control of mRNA decoding on the bacterial ribosome. *Advances in Protein Chemistry and Structural Biology*, **86**: 95–128.

Russell DA, Hatfull GF. 2017. PhagesDB: the actinobacteriophage database. *Bioinformatics*, **33**(5): 784–786.

Taliaferro D, Farabaugh PJ. 2007. An mRNA sequence derived from the yeast *EST3* gene stimulates programmed +1 translational frameshifting. *RNA*, **13**(4): 606–613.

Tan M, Heckmann K, Brünen-Nieweler C. 2001a. Analysis of micronuclear, macronuclear and cDNA sequences encoding the regulatory subunit of cAMP-dependent protein kinase of *Euplotes octocarinatus*: evidence for a ribosomal frameshift. *Journal of Eukaryotic Microbiology*, **48**(1): 80–87.

Tan M, Liang AH, Brünen-Nieweler C, et al. 2001b. Programmed translational frameshifting is likely required for expressions of genes encoding putative nuclear protein kinases of the ciliate *Euplotes octocarinatus*. *Journal of Eukaryotic Microbiology*, **48**(5): 575–582.

Theis C, Reeder J, Giegerich R. 2008. KnotInFrame: prediction of -1 ribosomal frameshift events. *Nucleic Acids Research*, **36**(18): 6013–6020.

Tinoco Jr I, Bustamante C. 1999. How RNA folds. *Journal of Molecular Biology*, **293**(2): 271–281.

Trinity L, Stege U, Jabbari H. 2024. Tying the knot: unraveling the intricacies of the coronavirus frameshift pseudoknot. *PLoS Computational Biology*, **20**(5): e1011787.

Turanov AA, Lobanov AV, Fomenko DE, et al. 2009. Genetic code supports targeted insertion of two amino acids by one codon. *Science*, **323**(5911): 259–261.

Türkel S. 2016. Amino acid starvation enhances programmed ribosomal frameshift in metavirus Ty3 of *Saccharomyces cerevisiae*. *Advances in Biology*, **2016**(1): 1840782.

Voorhees RM, Ramakrishnan V. 2013. Structural basis of the translational elongation cycle. *Annual Review of Biochemistry*, **82**(1): 203–236.

Wang LB, Dean SR, Shippen DE. 2002. Oligomerization of the telomerase reverse transcriptase from *Euplotes crassus*. *Nucleic Acids Research*, **30**(18): 4032–4039.

Wang RL, Liu XY, Meng QY, et al. 2025a. Evolutionary origin of the frameshift sites in the ribosomal frameshifting genes of *Euplotes*. *BMC Genomics*, **26**(1): 676.

Wang RL, Xiong J, Wang W, et al. 2016. High frequency of +1 programmed ribosomal frameshifting in *Euplotes octocarinatus*. *Scientific Reports*, **6**(1): 21139.

Wang YY, Nan B, Ye F, et al. 2025b. Dual modes of DNA N^6-methyladenine maintenance by distinct methyltransferase complexes. *Proceedings of the National Academy of Sciences of the United States of America*, **122**(3): e2413037121.

Wu B, Zhang HB, Sun RR, et al. 2018. Translocation kinetics and structural dynamics of ribosomes are modulated by the conformational plasticity of downstream pseudoknots. *Nucleic Acids Research*, **46**(18): 9736–9748.

Xu J, Hendrix RW, Duda RL. 2004. Conserved translational frameshift in dsDNA bacteriophage tail assembly genes. *Molecular Cell*, **16**(1): 11–21.

Ye F, Chen X, Li Y, et al. 2025. Comprehensive genome annotation of the model ciliate *Tetrahymena thermophila* by in-depth epigenetic and transcriptomic profiling. *Nucleic Acids Research*, **53**(2): gkae1177.

Zhang TT, Fu JY, Li C, et al. 2025. Novel findings on the mitochondria in ciliates, with description of mitochondrial genomes of six representatives. *Marine Life Science & Technology*, **7**(1): 79–95.

Zheng WB, Li C, Zhou ZR, et al. 2025. Unveiling an ancient whole-genome duplication event in *Stentor*, the model unicellular eukaryotes. *Science China Life Sciences*, **68**(3): 825–835.