Genetic Translation Among Mouse Subspecies

Ye Liu, Kaivalya Pitale, Kurt Wanner
{liu.9756, pitale.4, wanner.61}@osu.edu
The Ohio State University

1 Abstract

Cross-species gene translation is crucial for understanding among biological organisms. An appropriate translation mechanism facilitates gene-level experiments in clinical contexts, making them more feasible and predictable. Traditional cross-species gene translation relies heavily on large-scale databases and manual annotations to process genotypes and RNA. However, fully annotating complex species is time-consuming, which presents a barrier to the practical application of cross-species genetic research. This study treats the fully annotated RNA sequences of Mus pahari and Mus caroli as languages and explores a translation mechanism by training on k-mer translation pairs between the two species.

2 Introduction

Cross-species gene expression translation is a critical topic in comparative genomics. A successful translation mechanism enables the interpretation of a new species' genotype through reference to a fully annotated species[16]. In recent years, natural language processing (NLP) techniques have been increasingly applied to the study of RNA sequences [18], treating nucleotide or amino acid chains as structured, language-like data. This approach allows models to learn sequence-level patterns in a manner analogous to human language translation.

In this study, we investigate the cross-species translation of RNA sequences between Mus caroli and Mus pahari, two closely related mouse species. We developed a robust sequence-to-sequence pipeline to capture underlying patterns in orthologous gene expression. To this end, we curated a dataset comprising RNA sequences extracted from brain, liver, heart, and kidney tissues of both species. By aligning gene annotations and extracting the corresponding nucleotide sequences, we constructed a large-scale paired dataset of orthologous genes. After applying various k-mer segmentation strategies, we organized k-mer translation pairs that serve as the foundation for training on RNA sequence translation. (Note: This paragraph was written in Chinese and ChatGPT was used to translate the text from Chinese to English.)

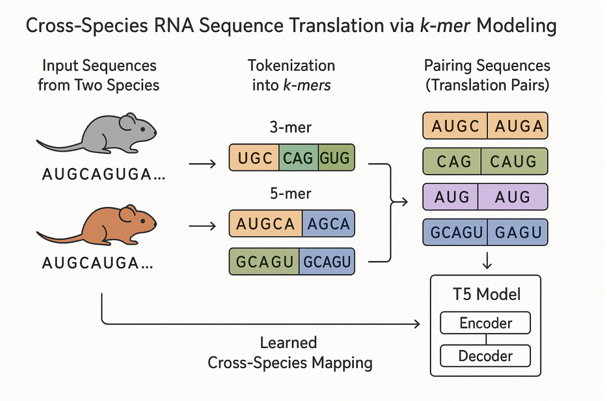
Cross-Species RNA Sequence Translation via *k-mer* Modeling
Input Sequences from Two Species
Tokenization into *k-mers*
Pairing Sequences (Translation Pairs)
AUGCAGUGA...
AUGCAUGA...
3-mer
UGC
CAG
GUG
5-mer
AUGCA
AGCA
GCAGU
GCAGU
AUGC
AUGA
CAG
CAUG
AUG
AUG
GCAGU
GAGU
T5 Model
Encoder
Decoder
Learned Cross-Species Mapping

Figure 1: Overview of the cross-species RNA sequence translation pipeline between *Mus caroli* and *Mus pahari*.

3 Related Work

In the fields of bioinformatics and genomics, traditional statistical methods have provided the academic foundation for the concept of gene translation. Particularly in the area of cross-species gene comparison, many research groups have developed tools based on various metrics to facilitate such analyses. At the same time, the idea of treating RNA sequences as a form of language has also been explored in several studies.

CoSIA: CoSIA introduces a metric-driven framework to assess gene expression variability, diversity, and tissue specificity across species.It integrates various data processing pipelines, including the use of the Bgee database for curated wild-type, non-diseased RNA-seq datasets, variance stabilization using

DESeq2[13], and ortholog mapping through resources such as NCBI HomoloGene and BiomaRt. By calculating coefficients of variation, it highlights genes with high inter-species or inter-tissue variability; diversity and specificity metrics further characterize expression uniformity or concentration within tissues[8].

The traditional methods used for this task before are exhaustive, but can be computationally slow.

DNABERT: A notable advancement in applying language models to biological sequences is DNABERT, introduced by Ji et al. This study proposed a novel paradigm where DNA sequences are treated as language-like data and modeled using BERT. DNABERT is pre-trained on genome-scale sequences using overlapping k-mers, capturing local and global dependencies within nucleotide sequences. This approach demonstrated significant improvements in various downstream genomics tasks, including transcription factor binding site prediction and promoter identification[9].

BLASTX alignment (pre-DIAMOND baseline): original BLASTX searches routinely took hours to days on typical metagenomic datasets (DIAMOND was introduced as a 20,000× faster alternative to address this bottleneck) [3].

Maximum-likelihood phylogeny (RAxML): On the 16S.B.ALL dataset (3,000 sequences), RAxML required 647–2,150 hours to infer trees, compared with just 2–6.3 hours for FastTree on the same data[12].

Statistical GWAS (SF-GWAS): A biobank-scale study, the secure federated GWAS pipeline ran 5.3 days in total on the UK Biobank (including QC, PCA, and association testing)[5].

4 Methodology

In the methodology section, we introduce how our project leverages the collected data to construct a training objective from several key perspectives. First, we describe the preprocessing steps and the formation of translation pairs. Next, we perform k-mer segmentation on these pairs to obtain RNA sequence translation pairs suitable for training. Finally, we train our data using a T5-based model[15] architecture to learn the cross-species translation patterns.

4.1 Data Selection and pre-processing

In this study, we selected two closely related rodent species, *Mus caroli* and *Mus pahari*, as our model organisms for cross-species RNA sequence translation. These species were chosen due to their phylogenetic proximity, high-quality reference genomes, and availability of comparable transcriptomic datasets[11].

4.1.1 Reference Genome and Gene Annotation Alignment

To construct accurate gene-level sequence pairs between *Mus caroli* and *Mus pahari*, we retrieved their reference genome assemblies and annotation files from the National Center for Biotechnology Information (NCBI). The reference genome for *Mus caroli* was obtained under accession **GCF_900094665.2**, while that for *Mus pahari* was retrieved as **GCF_900095145.1**. Each dataset includes the full genomic sequence in FASTA format and corresponding gene annotations in GTF format.

The genomic FASTA files (`.fna`) were used as the reference for sequence extraction, while the GTF files provided structured annotations of gene features, including gene identifiers, chromosomal positions, strand orientation, and gene biotypes. These annotations were used to extract strand-aware gene-level sequences using coordinate-based slicing and reverse-complement operations where necessary.

Table 1: Reference Genome Assemblies Used

Species	Accession ID
Mus caroli	GCF_900094665.2
Mus pahari	GCF_900095145.1

From these annotations, we extracted 32,457 gene sequences from *Mus caroli* and 29,483 gene sequences from *Mus pahari*. A set of 15,833 orthologous gene identifiers shared across both species was then used to construct cross-species translation pairs.

4.1.2 K-mer Tokenization

To convert nucleotide sequences into a format suitable for natural language processing models, we employed a sliding window tokenization technique using fixed-length substrings known as **k-mers**.

$$\text{k-mer}(S,k) = \left\{ S_i^{(k)} = S[i:i+k] \mid 0 \le i \le |S| - k \right\} \quad (1)$$

where k and $S_i^{(k)}$ represents the i-th k-mer substring of length k within sequence S. The choice of k was motivated by a balance between biological relevance and model complexity, allowing sufficient contextual representation while maintaining a tractable vocabulary size of 4^k.

The use of k-mer tokenization is inspired by previous work in immunogenomics, where deep learning models were applied to somatic hypermutation prediction using variable-length k-mers ranging from 5 to 21 bases [17]. This modeling captured both local and extended nucleotide dependencies.

4.1.3 Translation Pair Construction

To formulate the cross-species RNA translation task, we constructed a parallel corpus of gene-level k-mer token sequences between *Mus caroli* and *Mus pahari*. After extracting the full set of annotated gene sequences from both species, we identified a shared set of 15,833 orthologous gene identifiers based on consistent `gene_id` annotations. For each orthologous gene $g \in G_{\text{shared}}$, let S^g_{caroli} and S^g_{pahari} denote the nucleotide sequences of gene g in *Mus caroli* and *Mus pahari*, respectively. Each sequence was tokenized into overlapping k-mers, yielding:

$X^g = \text{kmer}(S^g_{\text{caroli}}, 7), \; Y^g = \text{kmer}(S^g_{\text{pahari}}, 7)$ The resulting dataset consists of sequence-to-sequence training pairs:

$$\mathcal{D} = \{(X^g, Y^g) \mid g \in G_{\text{shared}}\}$$

This formulation enables the learning of a mapping function:

$$f : X \to Y$$

where f is the model trained to translate k-mer sequences from *Mus caroli* to their corresponding sequences in *Mus pahari*.

4.2 Implementation

This project's implementation, training procedure, and code can be found here.

Table 2: Example rows from the translation_pairs_k7.tsv dataset. Each input-output pair consists of overlapping 7-mer tokens representing orthologous gene sequences in *Mus caroli* and *Mus pahari*.

Input (Caroli 7-mers)	**Output (Pahari 7-mers)**
ATGCGTA TGCG-TAC ...	ATGGTGA TGGT-GAC ...
TACCTGC ACCT-GCA ...	TACCAGT ACCAGTT ...
GAGTCAA AGT-CAAG ...	GAGTTAA AGT-TAAC ...

4.2.1 Model Selection

We finetuned the Google T5-Base model [14] with 220 million parameters. We choose this model because it treats all tasks as text-to-text transformations, making it suitable for the purpose of RNA translation. This model size was chosen to accommodate memory requirements, ensuring both the model and training data fit onto a single GPU.

4.2.2 Tokenizing

Each k-mer sequence was assigned its own tokenizer, ensuring that each possible substring of RNA characters was identified by a single token. There are 4 possible types of Nucleotides: adenine (A), thymine (T), guanine (G), and cytosine (C). However, in our dataset, the generic (N) character was used to indicate an undetermined nucleotide. The vocabulary size for each tokenizer was determined by the combinatorial possibilities of k-mers, resulting in a total of 5^k tokens plus miscellaneous tokens for padding and the start of sequences.

4.2.3 Training

Each experiment was trained for 24 hours on an NVIDIA H100 GPU w/ 94 GB memory.

The selected datasets were divided into training (70%), evaluation (20%), and test (10%) data. Due to the file size of the dataset, the model was trained on inputs of 512 tokens at a time.

The following hyperparameters were used:

Loss Function : Cross Entropy Loss
Batch Size : 8
Learning Rate : $3e^{-6}$
Weight Decay : $1e^{-2}$
LR Optimizer : AdamW

4.2.4 Evaluation Metrics

We considered the translation task to be successful if the translated RNA sequence embedded the same set of genes as the input. To extract the set of mammalian genes embedded in a given RNA sequence, we used BLAST [1], which maps RNA sequences onto its embedded set of protein sequences (genes).

We evaluated our model by using various set similarity scores on the sets of identified genes in the reference genome and our model's output. The metrics used were the Jaccard Index:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

and the Dice-Sørensen coefficient:

$$DC(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{3}$$

To evaluate the biological relevance of the predicted RNA sequences, we performed a BLASTn-based validation against the Mus pahari reference transcriptome (GCF_900095145.1_rna.fna). A nucleotide database was constructed using NCBI's BLAST+ toolkit, ensuring that all matches were restricted to the target species.

Each predicted sequence and its corresponding source sequence were aligned independently to the Mus pahari RNA database. For each query, only the best alignment (based on the highest bitscore) was considered for subsequent analysis. The primary metrics extracted were the average sequence identity percentage and average bitscore across all queries.

For each query sequence, only the best matching alignment of highest bitscore [2] was retained. The following metrics were computed:

For each query sequence, only the best matching alignment with highest bitscore was retained. The following metrics were computed:

- **Identity Percentage**: For a given alignment between a query and a subject sequence, the identity percentage is calculated as:

$$\text{Identity } (\%) = \frac{\text{Number of matched nucleotides}}{\text{Total alignment length}}$$

- **Average Identity**: Across all queries, the average identity percentage was computed by:

$$\text{Average Identity} = \frac{1}{N} \sum_{i=1}^{N} \text{Identity}_i$$

where N is the number of queries and Identity$_i$ is the best match identity for the i-th query.

- **Bitscore**: The bitscore reflects the alignment quality and is calculated internally by BLAST based on the alignment score and statistical parameters of the database. We computed the average bitscore similarly:

$$\text{Average Bitscore} = \frac{1}{N} \sum_{i=1}^{N} \text{Bitscore}_i$$

4.3 Baseline Construction

To evaluate the effectiveness of our RNA k-mer translation task, we constructed multiple baselines using progressively more expressive machine learning models. These baselines serve both as performance references and as conceptual contrasts for our sequence modeling approach.

N-gram Language Model. As the simplest statistical baseline, we implemented a bigram model [10, 6, 4] to generate target sequences based on the frequency of k-mer co-occurrence. This model captures only local dependencies and lacks the ability to generalize beyond observed token transitions. Given a target RNA sequence $y = (y_1, y_2, \ldots, y_T)$ consisting of k-mers, the bigram model estimates the probability of the sequence as:

$$P(y_1, y_2, \ldots, y_T) \approx \prod_{t=1}^{T} P(y_t \mid y_{t-1}) \tag{4}$$

This model captures only local co-occurrence statistics between adjacent k-mers in the output sequence.

Embedding + MLP Mapping. To assess the importance of sequence order, we also constructed a bag-of-k-mers baseline using average embeddings of source sequences followed by a multilayer perceptron (MLP) [7]to predict the first target k-mer. This method does not account for positional information, serving as a useful comparison point to the encoder-decoder approach. As a non-sequential baseline, we treat the input as a bag of k-mers. Each k-mer x_i is mapped to an embedding vector and the overall input is summarized by average pooling:

$$v_x = \frac{1}{N} \sum_{i=1}^{N} \text{Embed}(x_i) \tag{5}$$

The first target k-mer is then predicted using a multilayer perceptron (MLP):

$$\hat{y}_1 = \text{MLP}(v_x) \tag{6}$$

This model ignores positional information and serves as a useful baseline to quantify the contribution of sequence ordering.

Each baseline was trained and evaluated on the same set of aligned k-mer translation pairs derived from orthologous genes between *Mus caroli* and *Mus pahari*. Performance was measured using token-level accuracy on the target sequence. (Note: ChatGPT was used for in-text equations and grammar correction.)

5 Results and Analysis

This section presents the experimental results from baseline models and neural training. Overall, the neural models significantly outperform the statistical and embedding-based baselines.

5.1 Baseline

To establish performance references, we implemented three baseline models using 3-mer segmented RNA sequences. This choice balances biological interpretability with computational efficiency.

The results, evaluated using token-level accuracy (only first token for MLP), are summarized in Table 3.

Table 3: Baseline and Neural Model Token Accuracy

Model	**Accuracy (%)**
N-gram (Bigram)	2.18
MLP + Avg Embed	9.54
T5 Fine-tuned	12.71

These baselines provide a performance spectrum from statistical memorization to deep contextual modeling, and demonstrate the critical role of sequence modeling in cross-species RNA translation.

5.2 Results

Our top-performing experiment, which utilized 5-mer sequences, achieved a Dice-Sørensen coefficient of 0.7090 and a Jaccard index of 0.6314. Also, we utilize BLAST for validating the translated results with the overall identity accuracy.

Table 4: BLAST Summary Results: Comparison between Source and Predicted RNA Sequences

Type	**Avg Identity (%)**	**Avg Bitscore**
Source	99.95%	398.99
Prediction	95.67%	480.85

6 Future Work

Future work related to this project includes evaluating the effects of different k-mer sequences on model performance, expanding the dataset to test different subspecies of mouse and potentially other rodent species, and testing this translational model for clinical applications.

7 Contributions

This section identifies the contributions made to the project by each member.

Ye Liu

- Processed raw RNA sequences from *Mus caroli* and *Mus pahari*, and constructed training translation pairs using k-mer tokenization.
- Built a BLAST nucleotide database from the *Mus pahari* reference transcriptome to support cross-species validation.
- Designed and executed BLASTn-based evaluation pipelines to quantify translation performance based on identity percentage and bitscore.

Kaivalya Pitale

- Traditional methods testing: evolutionary tree alignment, statistical analysis and BLAST analysis.
- Attempted to run a minimalist version of BERT on a local device, but encountered repeated crashes due to insufficient memory and hardware limitations.

Kurt Wanner

- Created, ran, and managed model training and evaluation environment on OSC.
- Paid $9 per month for an Overleaf student plan.

References

[1] Stephen F. Altschul et al. “Basic local alignment search tool”. In: *Journal of Molecular Biology* 215.3 (1990), pp. 403–410. DOI: 10.1016/S0022-2836(05)80360-2.

[2] Ricardo Avila. *E-values and Bit-scores in BLAST*. https://ravilabio.info/notes/bioinformatics/e-value-bitscore.html. 2021.

[3] Benjamin Buchfink, Chao Xie, and Daniel H. Huson. “Fast and sensitive protein alignment using DIAMOND”. In: *Nature Methods* 12.1 (Jan. 2015), pp. 59–60. ISSN: 1548-7105. DOI: 10.1038/nmeth.3176. URL: https://doi.org/10.1038/nmeth.3176.

[4] Ciprian Chelba, Mohammad Norouzi, and Samy Bengio. *N-gram Language Modeling using Recurrent Neural Network Estimation*. 2017. arXiv: 1703.10724 [cs.CL]. URL: https://arxiv.org/abs/1703.10724.

[5] Hyunghoon Cho et al. “Secure and federated genome-wide association studies for biobank-scale datasets”. In: *Nature Genetics* 57.4 (Apr. 2025), pp. 809–814. ISSN: 1546-1718. DOI: 10.1038/s41588-025-02109-1. URL: https://doi.org/10.1038/s41588-025-02109-1.

[6] Tanya Garg, Daljeet Singh Bawa, and Sumayya Khalid. “A Novel GRU Based Encoder-Decoder Model (GRUED) Using Inverse Distance Weighted Interpolation for Air Quality Forecasting”. In: *International Journal of Image, Graphics and Signal Processing (IJIGSP)* 15.6 (2023), pp. 13–27. DOI: 10.5815/ijigsp.2023.06.02. URL: https://doi.org/10.5815/ijigsp.2023.06.02.

[7] Wen Guo et al. *Back to MLP: A Simple Baseline for Human Motion Prediction*. 2022. arXiv: 2207.01567 [cs.CV]. URL: https://arxiv.org/abs/2207.01567.

[8] Ananya Haldar et al. “CoSIA: an R Bioconductor package for CrOss Species Investigation and Analysis”. In: *Bioinformatics (Oxford, England)* 39.12 (2023), btad759. DOI: 10.1093/bioinformatics/btad759. URL: https://doi.org/10.1093/bioinformatics/btad759.

[9] Yanrong Ji et al. “DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome”. In: *Bioinformatics* 37.15 (Feb. 2021), pp. 2112–2120. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab083. eprint: https://academic.oup.com/bioinformatics/article-pdf/37/15/2112/50578892/btab083.pdf. URL: https://doi.org/10.1093/bioinformatics/btab083.

[10] Huayang Li et al. *N-gram Is Back: Residual Learning of Neural Text Generation with n-gram Language Model*. 2022. arXiv: 2210.14431 [cs.CL]. URL: https://arxiv.org/abs/2210.14431.

[11] Yang I Li et al. “Annotation-free quantification of RNA splicing using LeafCutter”. In: *Genome Research* 28.4 (2018), pp. 448–458. DOI: 10.1101/gr.234728.117. URL: https://genome.cshlp.org/content/28/4/448.

[12] Kevin Liu, C. Randal Linder, and Tandy Warnow. “RAxML and FastTree: Comparing Two Methods for Large-Scale Maximum Likelihood Phylogeny Estimation”. In: *PLOS ONE* 6.11 (Nov. 2011), pp. 1–11. DOI: 10.1371/journal.pone.0027731. URL: https://doi.org/10.1371/journal.pone.0027731.

[13] Michael I Love, Wolfgang Huber, and Simon Anders. “Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2”. In: *Genome Biology* 15.12 (2014), p. 550. DOI: 10.1186/s13059-014-0550-8. URL: https://doi.org/10.1186/s13059-014-0550-8.

[14] Colin Raffel et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

[15] Colin Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2023. arXiv: 1910.10683 [cs.LG]. URL: https://arxiv.org/abs/1910.10683.

[16] Katherine L Seib, Gordon Dougan, and Rino Rappuoli. "The key role of genomics in modern vaccine and drug design for emerging infectious diseases". In: *PLoS genetics* 5.10 (2009), e1000612. DOI: 10.1371/journal.pgen.1000612. URL: https://doi.org/10.1371/journal.pgen.1000612.

[17] Catherine Tang, Artem Krantsevich, and Thomas MacCarthy. "Deep learning model of somatic hypermutation reveals importance of sequence context beyond hotspot targeting". In: *iScience* 25.1 (2022), p. 103668. ISSN: 2589-0042. DOI: 10.1016/j.isci.2021.103668. URL: https://www.sciencedirect.com/science/article/pii/S2589004221016382.

[18] Zhihan Zhou et al. *DNABERT-2: Efficient Foundation Model and Benchmark For Multi-Species Genome*. 2024. arXiv: 2306.15006 [q-bio.GN]. URL: https://arxiv.org/abs/2306.15006.