Supplemental material for
Deep structural clustering reveals hidden systematic biases in RNA sequencing data

Qiang Su[1,4,‡,*], Yi Long[2,‡], Deming Gou[3], Junmin Quan[4], Xiaoming Zhou[5,*] and Qizhou Lian[1,6,7,*]

Qiang Su, Xiaoming Zhou, and Qizhou Lian.
Email: su@chemie.uni-siegen.de, zhouxm@scnu.edu.cn and qz.lian@siat.ac.cn

This PDF file includes:
Methods

Methods

Detailed Computational Procedure for Analyzing RNA Multi-Dimensional Structure Using *K*-Mers:

Step-by-Step Procedure for VAE-GMM-Based RNA-read Conversion Tracking Model:

1. Transcript mRNA splitting based on the kmer setting (e.g., 50 nt):
    - Step 1.1: Divide the first identical mRNA sequence into non-overlapping segments of 50 nucleotides
        - Example: For an mRNA sequence “AUCGUACG…” and a *k*-mer size of 50 nucleotides (nt), we divide the sequence into non-overlapping 50 nt segments. This means segmenting from the 1st to the 50th position, from the 51st to the 100th position, and so on until the end of the sequence.
    - Step 1.2: Single-base shift for subsequent mRNAs. Process the second identical mRNA with a single-base shift as the start position along the mRNA, resulting in a different set of 50-mers.
        - Example: If the mRNA sequence is “AUCGUACG…” and the *k*-mer size is 50 nt, generate 50 nt segments from the 2nd to the 51st position, from the 52nd to the 101st position, and so on.
2. Consistent *k*-mer splitting across all identical mRNAs:
    - Step 2.1: Uniform *k*-mer Splitting with Single-Base Shifting
        - For a *k*-mer size of 50, the segments generated from the 1st identical mRNA sequence will start from the 1st, 51st, 101st positions, and so on. Similarly, 50-mer segments from the 2nd identical mRNA will start from the 2nd, 52nd, 102nd positions, and so forth. This pattern continues for all identical mRNA templates, ensuring that each sequence is uniformly split into *k*-mers.
    - Step 2.2: Consistent *k*-mer Generation from Identical mRNA Templates
        - In this process, the *k*-mer segment sets generated from the 1st, 51st, 101st, and subsequent identical mRNA templates will result in identical 50-mer segments. Likewise, the sets of 50-mers produced from the 2nd, 52nd, 102nd identical mRNA sequences will also be the same. This pattern repeats every 50 mRNA templates, ensuring uniformity in *k*-mer segmentation across all identical mRNAs.
    - Step 2.3: Comprehensive *k*-mer Alignment with Single-Base Shifting
        - Every unique starting position of a 50-mer segment in one mRNA template will yield a distinct set of 50-mers. This distinct set will then repeat with every 50 mRNA templates thereafter. This method achieves consistent and uniform *k*-mer segmentation across the entire transcript template. By capturing every possible *k*-mer within the sequences, this approach maintains the integrity and comprehensiveness of the genomic analysis.
3. Clustering of transcript-specific kmers set using VAE-GMM:
    - Step 3.1: VAE-GMM Clustering Algorithm:
        - Vector Representation: RNA *k*-mers (strings of RNA sequences) are converted into numeric vectors through one-hot encoding. Each base (A, C, G, U) is represented by a four-element vector, with a 1 indicating the presence of the base and 0s elsewhere.
        - Sampling Function for the VAE: The Variational Autoencoder involves sampling from a probability distribution to generate latent variables. The reparameterization trick is used to make this process differentiable, which is essential for training the model using backpropagation.
        - Implementing the Custom VAE Model: The script defines a custom VAE model. The total loss is the sum of the reconstruction loss and the KL divergence loss.

Metrics are tracked for these losses to monitor training progress.
    - Defining and Training the VAE: The encoder architecture includes multiple Conv1D layers to extract sequential features, culminating in dense layers that output the mean and log variance of the latent distribution. The decoder mirrors this process using Conv1DTranspose layers to reconstruct the original input. The model is compiled with the 'adam' optimizer and employs early stopping based on validation loss. Post-training, the encoder transforms *k*-mer data into latent vectors via the reparameterization trick, enabling effective clustering in the latent space.
    - Apply VAE-GMM Clustering: The GMM is initialized with the chosen number of clusters and a full covariance matrix, then fitted using the latent representations of the *k*-mers. Cluster labels are predicted for each data point, and the silhouette score is computed to evaluate clustering consistency. These results are added to the original data frame for further analysis.
4. Modeling Categorization and Data Grouping Based on Cluster Label:
    - Step 4.1: Initial data preparation:
        - For each 50-mer along the transcript, construct an initial table that includes:
            - The 50-mer sequence.
            - The modeling count for each 50-mer (initially set to 1 or global isoform *k*-mer overlapping frequency).
            - Sequencing counts obtained from the cleaned sequencing data.
            - The cluster label assigned to each 50-mer.
    - Step 4.2: Example Dataset:
        - Use a simplified dataset containing 10 example 50-mers for demonstration purposes.
    - Step 4.3: Modeling Data Grouping:
        - Group the 50-mers by their identical cluster labels.
    - Step 4.4: Cluster Distribution:
        - Construct a modeling count aggregate distribution for each group based on their cluster labels.
5. Sequencing Data Categorization and Grouping Based on Cluster Label:
    - Step 5.1: Grouping and Plotting:
        - Sequencing counts from individual k-mers within each read are aggregated based on the cluster label of the 50-mer region they correspond to.
        - Create a plot to visualize the relationship between sequencing count aggregate and cluster labels.
6. Correlation Fitting Between Cluster-Based Aggregated Modeling Count Aggregate and Sequencing Count Aggregate:
    - Step 6.1: Bias identification:
        - Identify discrepancies between the simulated data and actual plotting data for each cluster. These discrepancies indicate 3-D-structured bias effects.
        - Average the structured cluster-label assigned counts across all involved 50-mers to provide a simultaneous correction for sequencing biases.

Step-by-Step Procedure for GC-Based RNA-read Conversion Tracking Model:

1. Transcript mRNA splitting based on the kmer setting (e.g., 50 nt): same as VAE-based model
2. Consistent *k*-mer splitting across all identical mRNAs: same as VAE-based model
3. Procedural description for modeling data fitting:
    - Step 3.1: Initial data preparation:
        - For each 50-mer along the transcript, an initial table is constructed. This table includes the following elements:
            - The 50-mer sequence.

        - The modeling count for each 50-mer (initially set to 1 or global isoform *k*-mer overlapping frequency).
        - Sequencing counts obtained from the cleaned sequencing data.
        - GC-content assigned to each 50-mer.
    - Step 3.2: Example Dataset:
        - For demonstration, we use a simplified dataset containing 10 example 50-mers.
    - Step 3.3: Modeling Data Grouping:
        - We group the 50-mers by identical GC-content.
    - Step 3.4: GC-content Distribution:
        - A Gaussian distribution is constructed for each group, based on their GC-content, following the natural distribution principles.
    - Step 3.5: Parameter Estimation:
        - Key parameters-mean and standard deviation-are derived from the GC-content distribution within the modeling data. These parameters are intrinsic to the distribution and are determined independently of empirical sequencing data.
4. Procedural description for sequencing data fitting:
    - Step 4.1: Grouping and Plotting:
        - Sequencing counts are grouped by the GC-content of each 50-mer.
        - A plot is created to visualize the relationship between sequencing count aggregate and GC-content.
    - Step 4.2: Fitting the Gaussian Distribution:
        - We apply the fixed-parameter Gaussian distribution to the sequencing count aggregate data, highlighting and correcting the discrepancies between actual and simulated aggregates.
    - Step 4.3: Bias identification:
        - Discrepancies in GC-assigned counts are averaged across all 50-mers, providing a simultaneous correction for sequencing biases.

Step-by-Step Procedure for MFE-Based RNA-read Conversion Tracking Model (this model is quite similar as the GC-based model):

1. Transcript mRNA splitting based on the kmer setting (e.g., 50 nt): same as VAE-based model
2. Consistent *k*-mer splitting across all identical mRNAs: same as VAE-based model
3. Procedural description for modeling data fitting:
    - Step 3.1: Initial data preparation:
        - For each 50-mer along the transcript, an initial table is constructed. This table includes the following elements:
            - The 50-mer sequence.
            - The modeling count for each 50-mer (initially set to 1 or global isoform *k*-mer overlapping frequency).
            - Sequencing counts obtained from the cleaned sequencing data.
            - MFE assigned to each 50-mer. For each *k*-mer, the function *rnafold* to compute the RNA secondary structure and its associated energy.
    - Step 3.2: Example Dataset:
        - For demonstration, we use a simplified dataset containing 10 example 50-mers.
    - Step 3.3: Modeling Data Grouping:
        - We group the 50-mers by identical MFE.
    - Step 3.4: MFE Distribution:
        - A Gaussian distribution is constructed for each group, based on their MFE, following the natural distribution principles.

- Step 3.5: Parameter Estimation:
    - Key parameters-mean and standard deviation-are derived from the MFE distribution within the modeling data. These parameters are intrinsic to the distribution and are determined independently of empirical sequencing data.

4. Procedural description for sequencing data fitting:
    - Step 4.1: Grouping and Plotting:
        - Sequencing counts are grouped by the MFE of each 50-mer.
        - A plot is created to visualize the relationship between sequencing count aggregate and MFE.
    - Step 4.2: Fitting the Gaussian Distribution:
        - We apply the fixed-parameter Gaussian distribution to the sequencing count aggregate data, highlighting and correcting the discrepancies between actual and simulated aggregates.
    - Step 4.3: Bias identification:
        - Discrepancies in MFE-assigned counts are averaged across all 50-mers, providing a simultaneous correction for sequencing biases.

Encoding *k*-mer sequence into latent vectors and training the convolutional neural network

We explored various methods to encode RNA *k*-mer sequences into latent vectors for clustering analysis, ultimately adopting a variational autoencoder (VAE) to learn a compressed representation of the data in a lower-dimensional space. First, each *k*-mer sequence was transformed into a one-hot-encoded vector, assigning each nucleotide (A, C, G, T) a distinct binary code. This process generated an input format of (k, 4) per sequence, where k is the sequence length, which was then flattened into a single vector of length 4k.

The VAE was built with an encoder and a decoder. The encoder began by reshaping the flattened vectors back to (k, 4), then applied three one-dimensional convolutional layers to capture local sequence motifs: a Conv1D layer with 64 filters and a kernel size of 3 (output (k, 64)), another with 32 filters (output (k, 32)), and a final one with 16 filters (output (k, 16)). After these convolutional layers, the output was flattened and passed into dense layers that produced a mean vector (z_{mean}) and a log variance vector $(z_{log\sigma^2})$ for the latent space, which had a dimensionality of 2. Sampling of the latent vector z was achieved via the reparameterization trick,

$$z = z_{mean} + \exp(0.5 z_{log\sigma^2}) \odot \varepsilon,$$

with ε drawn from a standard normal distribution. The decoder took these latent vectors and reconstructed the input sequences by first using a dense layer to restore the flattened dimension, then reshaping to (k, 16), and finally applying multiple Conv1DTranspose layers with progressively larger filter sizes (16, 32, and 64), ending with a layer of 4 filters and a sigmoid activation to produce outputs in [0, 1]. The model was trained using Adam, with a combined loss of binary cross-entropy and Kullback-Leibler divergence. Early stopping was employed, monitoring validation loss and halting training after five epochs without improvement.

Once trained, the encoder was used to project all *k*-mer sequences into the learned latent space. A Gaussian Mixture Model (GMM) was then applied to cluster the latent vectors. The silhouette score was calculated to gauge the quality of the resulting clusters, with higher values indicating more coherent groupings. Finally, Uniform Manifold Approximation and Projection (UMAP) was used to visualize the clusters by reducing the latent vectors to two dimensions while preserving global data structure. Each point in the resulting scatter plot represented a *k*-mer sequence, colored according to cluster membership, with the option to highlight particular clusters if desired.

The theoretical consideration of GC- and MFE-organized RNA structural modeling:

The theoretical foundation and empirical evidence supporting the use of GC- and MFE-based algorithms are anchored in the binomial distribution model for analyzing GC-content as a statistical framework. This model excels in estimating the likelihood of witnessing a certain

number of successes across a preset number of independent events. In the realm of genetics, these "successes" are identified as occurrences of guanine (G) or cytosine (C) bases (grouped together as degenerate bases, S), within an RNA sequence of length k. Each nucleotide position is considered a separate trial with possible outcomes: a success (S) when a G or C base is encountered, or a failure (W) for adenine (A) or uracil (U) bases (Fig. 2a). In this model, the probability of finding a degenerate base (S) at any position is denoted as (p), while the chance of coming across a base (W) is (q = 1 - p). For the purpose of simplification, when analyzing RNA sequences, these fragments are regarded as sequences of (k) independent Bernoulli trials, each with a probability of success (p = 1/2), reflecting an equal likelihood of hitting a G or C base. Let (X) represent the random variable that counts the occurrences of GC bases in these (k) trials. To determine the probability mass function (PMF) of (X), it's necessary to calculate the chance of achieving exactly (X) successes. This calculation involves multiplying the probability of (X) successes, (p^X), by the probability of the remaining (k-X) outcomes being failures, ((1-p)^{k-X}). The number of distinct ways in which (X) successes can appear among (k) trials is given by the binomial coefficient calculation. Therefore, the formula for the PMF of (X) in the context of the binomial distribution is given as:

$$P_X = \frac{k!}{X!*(k-X)!} * p^X * q^{(k-X)} = \frac{k!}{X!*(k-X)!} * (\frac{1}{2})^k \quad \text{(Eq.1)}$$

In this equation, (k! / (X!(k-X)!)) calculates the number of ways to select (X) successes out of (k) trials, (p^X) is the probability of these (X) successes, and ((1-p)^{k-X}) provides the probability of the other trials resulting in failures. This approach offers a strong statistical basis for assessing GC-content in RNA sequences. Utilizing the binomial distribution provides a precise and effective method to study the distribution of GC-content across RNA fragments of a defined length, through the pseudo-random arrangement of G and C bases. The binomial distribution is inherently suited for modeling outcomes across a constant number of trials (k), which aligns well with scenarios involving uniform RNA fragment lengths. However, its applicability is limited in situations involving mixed RNA populations or under conditions characterized by variability. To overcome this limitation, the Gaussian function has been adapted based on the principles of the binomial distribution. Specifically, the approximation of the binomial distribution by the Gaussian distribution becomes remarkably accurate when the size of the parameters of the binomial distribution-namely, k (total trials), kp (mean success), and kq (mean failure)-increases significantly. This adaptation follows from the de Moivre-Laplace theorem41, 42 and enhances the analytical versatility, enabling a more comprehensive examination of the distribution of GC-content. This relation is articulated as follows:

$$f_{(x)} = \frac{k!}{X!*(k-X)!} * p^X * (1-p)^{(k-X)} \sim \frac{1}{\sqrt{2\pi kpq}} e^{-\frac{(x-kp)^2}{2kpq}} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \text{ (Eq.2)}$$

The Gaussian distribution function significantly enhances data analysis' flexibility and utility, offering a sophisticated method for understanding a broad array of experimental phenomena beyond the capabilities of binomial distributions. In evaluating the distribution patterns of GC across different RNA fragments, the adaptability of the Gaussian distribution proves exceptionally valuable. It allows for a more nuanced and flexible analytical approach. Building on this foundation, the application of the Gaussian distribution has been extended beyond its traditional role in denoting probabilities to include the concept of frequency or occurrence rates. This important development involves converting the Probability Mass Function (PMF) to an Occurrence Mass Function (OMF). To achieve this, the PMF is multiplied by the total number of observations, denoted as 'N', within a specific transcriptomic dataset. In this context, 'N' refers to the total count of *k*-mers in the dataset, whether it is being modeled or sequenced. For modeling data, each *k*-mer is typically counted once by default, meaning that the total modeling count of all *k*-mers directly equals 'N'. In the case of sequencing data, 'N' also represents the aggregate count of all *k*-mers. This modified formulation can be expressed as:

$$f_{occu\_GC(x)} = A * \frac{1}{\sqrt{2\pi}\sigma\_GC} e^{-\frac{(x-\mu\_GC)^2}{2\sigma\_GC^2}} \text{ (Eq.3)}$$

This equation reflects the frequency or occurrence rate of specific GC *k*-mers within the dataset, broadening the scope of Gaussian distribution application in genomic data analysis.

Developing a model predicated on GC-content employing a Gaussian framework involves an elevated degree of abstraction, taking into account the complex two-dimensional structure of RNA molecules. The MFE metric is pivotal for quantitatively evaluating the stability of RNA secondary structures, which in turn, significantly influences RNA's molecular functions. To construct a model based on MFE, we initiate a sequence of theoretical steps which are anchored in the principles of an GC-based Gaussian model. To encapsulate this intricate relationship in a model, we employ linear modeling targeted at individual RNA sequences. Each sequence, denoted by the subscript "i", undergoes examination to discern how alterations in its GC-content (notated as (GC_i)) impact its corresponding MFE (MFE_i).43 This yields a linear relationship between GC-content and MFE for each sequence, which can be expressed as:

$$\mathrm{MFEi} = \beta 0\mathrm{i} + \beta 1\mathrm{i} * \mathrm{GCi} \quad \text{(Eq.4)}$$

This equation does not function as a predictive model for MFE from GC-content but rather illustrates the correlation between these factors across all RNAs provided. The computational workflow involves sequencing each spike-in, followed by the individual determination of both GC-content and MFE. The term β0i symbolizes the expected MFE value when the GC-content is null, representing the foundational stability of the RNA structure. Conversely, β1i, likely negative, elucidates the decrement in MFE occurring with each incremental rise in GC-content, underscoring that heightened GC-content correlates with a stabler RNA structure, as reflected by a lowered MFE.

Progressing from a GC-centric model to one anchored in MFE necessitates the adoption of Gaussian distribution to model the array of MFE values respective to a particular GC-content. Such distribution is mathematically encapsulated as follows:

$$f_{occu\_MFE(x)} = A * \frac{1}{\sqrt{2\pi}\sigma\_MFE} e^{-\frac{(x-\mu\_MFE)^2}{2\sigma\_MFE^2}} \quad \text{(Eq.5)}$$

Herein, the MFE values of sequences with akin GC-contents cluster around a mean MFE value (μ_MFE), with a specific standard deviation (σ_MFE). This Gaussian framework not only simplifies the complex interplay between GC-content and RNA structural stability but also paves the way for a deeper understanding of the molecular functionalities and stability patterns of RNA through the prism of MFE.

Data analysis:

1. Data processing. The processing pipeline for analyzing raw paired-end sequencing data incorporates a suite of specialized tools tailored for quality control, base trimming, accurate alignment, and comprehensive genomic analysis. The initial quality assessment of the data is conducted using FastQC (v0.11.8) provided by Babraham Bioinformatics. This is followed by the removal of adapter sequences and the trimming of low-quality bases utilizing both Cutadapt (v2.10) and Trimmomatic (v0.39). Subsequent alignment of the trimmed reads to the human reference genome GRCh38.p14 (as per Ensembl version) is achieved through the combined use of HISAT2 (v2.2.1) and STAR (v2.7.3a), ensuring precise and reliable mapping. Post-alignment tasks, including the sorting and indexing of mapped reads, are managed using Samtools (v1.7) and visualized with the Integrative Genomics Viewer (IGV) for in-depth analysis. Further, the quality of the RNA-seq dataset post-alignment is assessed utilizing the RSeQC package, verifying the integrity of the data analysis. Additionally, the alignment of *k*-mer segments to transcript references according to Ensembl annotations enhances the granularity of the genomic insights provided. Collectively, the integrated use of these tools forms a robust pipeline that guarantees meticulous preparation, alignment, and insightful genomic analysis across the datasets. The 3-D structure is predicted using AlphaFold (https://alphafoldserver.com/). Mol* is used for 3-D structure visualization (https://molstar.org/).

2. Cutadapt for adapter removing. Cutadapt is used to remove adapter sequences, which are short fragments of DNA ligated to the ends of the sequenced fragments during library preparation. For different librarys, the specific adaptators are employed and then removed.

2.1 N8 library:

The specific adaptors for the N8 samples are:

5'-CTGTCTCTTATACACATCTCCGAGCCCACGAGAC-3'

5'-CTGTCTCTTATACACATCTGACGCTGCCGACGA-3'

2.2 293T library

The specific adaptors for the 293T samples are:

5'-AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC-3'

5'-AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT-3'

2.3 N1-N6 human tissue library

The specific adaptors for the N1-N6 human tissue samples are:

5'-AAGTCGGAGGCCAAGCGGTCTTAGGAAGAC-3'

5'-AAGTCGGATCGTAGCCATGTCGTTC-3'

3. Trimmomatic for further cleaning data. It is designed to remove adapter sequences, trim low-quality bases, and filter out poor-quality reads from the raw data. Trimmomatic is specifically useful for improving read quality, which is crucial for accurate alignment and downstream analysis.

4. 4.2 Alignment of cleaned reads to a reference

4.2.1 STAR indexing

4.2.2 STAR mapping

4.3 hisat2 aligning the cleaning reads to reference

5. samtools converting .sam to .bam

6. samtools index *.bam

7. samtools extracting transcript-specific sequencing coverage

8. Outputing the length of paired-end determined fragment from .bam

9. *k*-mer counting in *k*-mer counting in kmer_counting_loop.py (https://github.com/QiangSu/N-sequence)

10. UMAP for Dimensionality Reduction and Visualization

10.1 Transcript mRNA Splitting and *K*-mer Generation

We first split each transcript into 50-nt non-overlapping segments. For subsequent identical mRNAs, a single-base shift is applied, generating distinct 50-mer sets. This process ensures consistent and uniform *k*-mer generation across all identical mRNAs, capturing every possible *k*-mer within the sequences.

10.2 *K*-mer Clustering Using VAE-GMM

Each 50-mer is converted into a numerical vector representation. These vectors are then clustered using the VAE-GMM algorithm, grouping sequences based on their 3-D structural characteristics.

10.3 UMAP for Dimensionality Reduction and Visualization

The resulting *k*-mer clusters are represented as a matrix of numerical values, which serves as the input for UMAP. UMAP is then used to reduce the high-dimensional *k*-mer feature space into two dimensions, enabling effective visualization of the structural clustering.

11. Generating RNA-seq simulation data using polyester: This R script polyester_simulated_data.R is designed for simulating RNA-Seq data by using a reference transcriptome, abundance data, and specific simulation parameters. It begins by installing any required packages, then proceeds to load the reference transcriptome file. From this file, the script extracts relevant transcript IDs, aligning them with the supplied abundance data to form an abundance matrix. It then establishes the simulation environment by setting fold changes and specifying an output directory. Throughout these steps, the script also verifies the presence of essential dependencies, ensuring the entire simulation process runs smoothly.

Collection of tissue samples

Human colorectal samples were ethically obtained from Sun Yat-sen University Cancer Center and Shenzhen University General Hospital. Prior to sample collection, all donors provided informed consent, allowing for the acquisition of biopsies and the performance of comprehensive molecular profiling of transcriptomes.

Cell culture:

HEK293T cells were cultured in DMEM high glucose (catalog no. SH30022.01, HyClone) supplemented with 10% FBS (catalog no. 10100147, Thermo Fisher). The cells were maintained at optimal conditions, incubated at 37°C with 5% CO2 and maintained in an environment of saturating humidity. MCF7 cells were cultured in DMEM supplemented with 10% FBS, 5% Penicillin/Streptomycin, and 1 mg/mL Insulin (Gibco). The cells were maintained at 37°C with 5% CO_2 in a humidified incubator.

Library preparation:

The RNA-seq library preparation methods for spike-ins, Hex293T, and colorectal samples share a common protocol based on the VAHTS Universal V8 RNA-seq Library Prep. This protocol encompasses several essential steps, including RNA fragmentation, hexamer-directed cDNA synthesis, end repair and dA-tailing of fragments, adaptor ligation, PCR library amplification, and subsequent sequencing. Notably, when dealing with spike-in samples, an alternative tagmentation step is introduced to replace the step of end repair, dA-tailing, and adaptor ligation steps.

RNA isolation:

Total RNAs were extracted from cells using the RNAiso Plus kit (catalog no. 9109, TaKaRa Biotechnology) following the manufacturer's instructions. The extracted RNA was then resuspended in RNase-free water, which was consistently used throughout all RNA-related assays. To assess the RNA quality, a 2100 Bioanalyzer RNA picochip was employed. Subsequently, the total RNA was aliquoted into 5 μg portions for long-term storage at -80°C, ensuring preservation for future experiments.

rRNA depletion:

To ensure accurate profiling of non-rRNA molecules in the RNA samples, we employed the commercial Ribo-off rRNA Depletion Kit (Human/Mouse/Rat) (kit no. N406-01, Vazyme). This kit offers a well-established and efficient method for removing ribosomal RNA (rRNA) from the total RNA pool. The rRNA depletion process involves several key steps. Firstly, the total RNA sample is mixed with the rRNA removal probes, which are specifically designed to target and bind to rRNA molecules. Subsequently, the mixture undergoes an incubation period, allowing the rRNA removal probes to effectively hybridize with the rRNA molecules. Following incubation, the removal solution is introduced to the mixture, leading to the degradation of the rRNA-probe complexes. The remaining RNA, enriched with non-rRNA molecules, is then purified to obtain the desired RNA pool for downstream applications. By implementing the Ribo-off rRNA Depletion Kit, we can confidently reduce the abundance of rRNA, ensuring a more comprehensive analysis of

the non-rRNA fraction and enhancing the sensitivity of our subsequent RNA-seq experiments. This rRNA depletion step plays a crucial role in obtaining high-quality data, ultimately contributing to the accuracy and reliability of our molecular analyses.

Spike-in circularization

For circular spike-in RNA preparation, a single-stranded RNA oligo with a 5' phosphate and a 3' OH end was ligated to form a circular RNA. The spike-in sequences were used in the experiment:

Spike-in: 5'-phosphate-AAAAAAAAGGUAACUGCGNUUANCACNAGCNCCANGAGNAACNACANGAAUUCUUUAUAAAAAAAA-OH-3'

During RNA synthesis, random nucleotides (N) are incorporated by introducing an equimolar mixture of A, U, C, and G at each “N” position, ensuring all possible sequence variants are represented equally. While slight biases can occur due to synthesis inefficiencies, using high-quality reagents and protocols minimizes these effects, allowing for near-equimolar representation of all possible sequences.

To create the spike-in RNA, 5 μL of a 10 μM spike-in RNA was added to a 20 μL reaction system containing 1 mM ATP, Rnase inhibitor of 2 u/μL, 1 μL T4 RNA ligase 1 (ssRNA Ligase) (no. M0204S, New England Biolabs), and 50% PEG8000 with 15%. The reaction was incubated at 25°C for 1-2 hours and then terminated by boiling for 2 minutes. The cooled mixture was then treated with RNase R (no. R7092L, Beyotime) by adding the enzyme (2 μL) and reaction buffer (2.3 μL 10× buffer), followed by incubation at 37°C for 30 minutes to degrade linear RNA. The reaction was terminated by heating at 70°C for 10 minutes, leaving only the circular RNA intact.

Reverse transcription:

The rRNA-free RNA samples were subjected to reverse transcription using random hexamers and SuperScript™ IV reverse transcriptase (no. 18090200, Invitrogen), following the manufacturer's instructions. In a 20 μL reaction system containing 100 ng of RNA template, 2.5 μM hexamers, dNTP Mix (10 mM each), and 1 μL (200 units) of SuperScript™ IV, the mixture was incubated at 25°C for 10 minutes, followed by 42°C for 50 minutes. Finally, the reaction was heated to 70°C for 15 minutes to inactivate the reverse transcriptase.

Tagmentation:

For tagmentation, 50 ng of DNA was combined with 30 μL reaction system consisting of 1× insertion buffer and 2 μL Tn5-50 adaptor index (10 μM). The mixture was incubated at 55°C for 5 minutes. Subsequently, another 30 μL of 2× Tn5 Digestion Mix (TransNGS® Tn5 DNA Library Prep Kit for Illumina®, no. KP101) was added and incubated together at 55°C for 5 minutes. The resulting tagmented library structure is as follows:

5'-AATGATACGGCGACCACCGAGATCTACAC-i5-TCGTCGGCAGCGTCAGATGTGTATAAGAGACAG-NNNNNN-CTGTCTCTTATACACATCTCCGAGCCCACGAGAC-i7-ATCTCGTATGCCGTCTTCTGCTTG-3'.

PCR amplication:

The library was then amplified by PCR using HIFI KAPA master mix (2×) with the following mixture: 25 μL 2× HIFI KAPA master mix, 10 μL cDNA, 13 μL H2O, and 1 μL of either the universal forward or reverse primer (10 μM):

Universal Forward Primer: 5'-AATGATACGGCGACCACCGAGATCTACACCTCTCTATACACTCTT-3' (:phosphorothioate)

Universal Reverse Primer: 5'-CAAGCAGAAGACGGCATACGAGATGTGACTGGAGTT-3' (:phosphorothioate)

The PCR reaction was then performed in a thermocycler as follows: 95°C for 5 minutes, followed by 10-15 cycles of 95°C for 15 seconds and 60°C for 30 seconds. The amplified library was subsequently purified with 1.8× Ampure XP DNA Beads and resuspended in 20 μL H2O. One

microliter of the library was checked using the Qubit 4 Fluorometer with the dsDNA HS (high sensitivity) Assay Kit (Invitrogen) for quality assessment.

Sequencing:

The purified PCR libraries were submitted for sequencing using the Illumina NovaSeq 6000 (PE150) platform or the MGISEQ-2000 (PE150) platform. The data output was generated based on the specified yield set.