March 2023

# Statistical Analysis of Ribonucleotide Incorporation in Human Cells

Tejasvi Channagiri
*University of South Florida*

# Statistical Analysis of Ribonucleotide Incorporation in Human Cells

by

Tejasvi Channagiri

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Arts
Department of Mathematics & Statistics
College of Arts and Sciences
University of South Florida

Major Professor: Nataša Jonoska, Ph.D.
Lu Lu, Ph.D.
Joel Rosenfeld, Ph.D.

Date of Approval:
March 22, 2023

Keywords: genomics, exploratory analysis, R, Bioconductor

# DEDICATION

To my mother and family. Thank you for your support.

## ACKNOWLEDGMENTS

I thank my advisor Dr. Natasha Jonoska for entrusting me with projects that were beyond my skill level and allowing me to rise to the challenge, as well as the leading the Math-Bio group with Dr. Masahico Saito. I thank my committee members Dr. Lu Lu and Dr. Rosenfeld for being great mentors, instructors, and collaborators. I thank Dr. Seung-Yeop Lee for teaching some of my favorite classes in probability theory. I thank Dr. Sherwin Kouchekian for his great course in real analysis. I thank Dr. Dmytro Savchuk for his support and guidance from the beginning of the program. I thank my fellow group members, Abdulmelik Mohammed, Margherita Ferrari, Lina Fajarado-Gomez, Trevor Ballard, Francisco Martinez, and Van Pham for their friendship and collaboration. I thank my classmate Jhonathan Medri-Cobos for being a great friend and inviting me to take part in an exciting summer project. I thank the Storici Lab, including Francesca Storici, Youngkyu Jeon, Penghao Xu, and Deepali Kundnani for their collaboration and friendship. I thank my good friend Max for being an amazing roommate. I thank the many others who have helped me throughout both my undergraduate and graduate career, though I cannot name them all here. Without the help of all these individuals, this thesis would not have been possible.

The contributors on this project are: Tejasvi Channagiri, Francisco Martinez, Trevor Ballard, Penghao Xu, Deepali Kundnani, Taehwan Yang, Mo Sun, Margherita Ferrari, Abdulmelik Mohammed, Nataša Jonoska, and Francesca Storici.

This research was (partially) supported by a grant from W.M. Keck Foundation awarded to F. Storici and N. Jonoska and was performed under auspices of Southeast Center for Mathematics and Biology, an NSF-Simons Research Center for Mathematics of Complex Biological Systems, under National Science Foundation Grant No. DMS-1764406 and Simons Foundation Grant No. 594594.

## TABLE OF CONTENTS

List Of Tables . . . ii

List Of Figures . . . iii

Abstract . . . v

Chapter 1 Introduction . . . 1
- 1.1 The Human Genome and Sequencing . . . 1
- 1.2 Ribose-Seq and Ribonucleotide Incorporation . . . 2
- 1.3 Data . . . 3
- 1.4 Questions . . . 5

Chapter 2 Literature Review . . . 7
- 2.1 Previous Studies of Ribonucleotide Incorporation . . . 7
- 2.2 Multiple Change-Point Problem . . . 7
- 2.3 Circular Binary Segmentation . . . 8
- 2.4 Sparse Linear Regression . . . 10
- 2.5 Hidden Markov Model Classifier . . . 11
- 2.6 AIC/CV Window Width Selection . . . 13
- 2.7 Smoothing Spline Method . . . 16

Chapter 3 Methods . . . 19
- 3.1 Programming Tools and Packages . . . 19
- 3.2 Preliminary Processing and Binning . . . 19
- 3.3 AIC and CV Windows Width Selection . . . 20
- 3.4 Distribution Fitting . . . 21
- 3.5 Determination of Hotspots With Distributions . . . 23
- 3.6 Smoothing-Spline Windows . . . 24
- 3.7 Similarity Metrics . . . 24
  - 3.7.1 Spearman Correlation . . . 25
  - 3.7.2 $L^p$ Distance . . . 25
  - 3.7.3 Hotspot Similarity . . . 26
  - 3.7.4 Multiple-Sample Hotspot Indicators . . . 26
- 3.8 Permutation Tests . . . 27
- 3.9 DNA-Seq Coverage . . . 29
- 3.10 Kmer Correlation . . . 30
- 3.11 Annotation Correlation . . . 31

Chapter 4 Results . . . 33
- 4.1 Data Summary . . . 33
- 4.2 Window Width Selection . . . 34
- 4.3 Sample Similarity . . . 34
- 4.4 Distribution Fitting . . . 35
- 4.5 Kmer Correlation . . . 35
- 4.6 Annotation Correlation . . . 36
- 4.7 DNA-Seq Coverage . . . 36
- 4.8 Smoothing Spline Hotspots . . . 37

Chapter 5 Conclusions and Future Work . . . 50

References . . . 54

## LIST OF TABLES

Table 1: The ribose-seq samples and cell types used in this study. . . . . . . . . . . . . . . . . . . . . 6

Table 2: Summary of ribonucleotide counts. Chromosome 1. + strand (A). - strand (B). Chromosome 1 size ≈ 249mb. RPB = “ribos per base” (ribonucleotides divided by chromosome length). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

Table 3: Summary of ribonucleotide counts (with repeat regions retained). Chromosome 1. + strand (A). - strand (B). Chromosome 1 size $\approx 249 \times 10^6$bp). RPB = “ribos per base” (ribonucleotides divided by chromosome length). . . . . . . . . . . . . . . . . . . . . . . . . 38

Table 4: Mean ribonucleotides per window for different windows widths. Chromosome 1. + strand (A). - strand (B). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

Table 5: Summary of annotated features. “N” is the total number of features. “Width median” is the median width in bases. Strand “*” indicates a feature that exists on both strand of the chromosome. All promoters in the dataset had width 1000b. Chromosome 1. . . . . . 40

Table 6: Median window widths obtained using spline method. Chromosome 1. + and - strand indicated in column name. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40

Table 7: Summary of the hotspot windows with smoothing-spline window selection. “Chromosome %” is the percentage of the chromosome covered by the hotspot windows. “Ribos %” is the percentage of the samples’ ribonucleotides in the hotspot windows. Chromosome 1. + strand (A). - strand (B). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

## LIST OF FIGURES

Figure 1: Ribonucleotides vs. chromosome position. Chromosome 1. + strand (left). - strand (right). Window width = 1mb. . . . . 39

Figure 2: AIC and CV log-likelihood vs. window width. Optimal width indicated by dotted line. Chromosome 1. + strand (A). - strand (B). . . . . 40

Figure 3: Spearman correlation between pairs of samples (A, B). Two-sided permutation p-values with 1000 permutations (C, D). Chromosome 1. + strand (A, C). - strand (B, D). . . . . 41

Figure 4: Each segment represents a 100kb hotspot window. A hotspot is a window whose ribonucleotide count is in the top 1% of windows in the sample. Chromosome 1. + strand (A). - strand (B). . . . . 42

Figure 5: Hotspot indicators. Each vertical bar is the number of samples with a hotspot in that position. A hotspot is a window whose ribonucleotide count is in the top 1% of windows in the sample (see Figure 4). There are several positions where at least three out of the six samples share a hotspot. Chromosome 1. + strand (A). - strand (B). Window width = 100kb. . . . . 42

Figure 6: Permutation analysis to check the significance of the shared hotspots in Figure 5. We permute the data 1000 times using the adjacent swapping scheme (see Section 3.8) and evaluate the number of positions with $\geq 3$ samples sharing a hotspot. The permuted values are black bars and the observed value is a red line. Chromosome 1. + strand (A). - strand (B). Window width = 100kb. . . . . 43

Figure 7: Histograms of zero-truncated Poisson distributions (green) fit to the data (orange). Window width = 100kb. Chromosome 1. + strand (A). - strand (B). . . . . 43

Figure 8: Histograms of zero-truncated negative-binomial distributions (green) fit to the data (orange). Window width = 100kb. Chromosome 1. + strand (A). - strand (B). . . . . 44

Figure 9: Ranks of kmer counts and ribonucleotide counts. The orange curves show the ranks of A and C counts in each window. The green curves show the ranks of ribonucleotide counts in each window. Chromosome 1 (+ strand). Sample FS185 only. Window width = 1mb. . 44

Figure 10: Spearman correlation between the nucleotide frequencies and the ribonucleotide frequencies (A, B). Two-sided permutation p-values of correlations using adjacent swapping scheme with 1000 permutations (see Section 3.8) (C, D). Chromosome 1. + strand (A, C). - strand (B, D). Window width = 100kb. . . . . 45

Figure 11: Relative frequencies of ribonucleotides (green) vs. annotated features (orange). The annotated features are: CpG islands (A, B), exons (C, D), and promoters (E, F). Sample FS185 only. Chromosome 1. + strand (A, C, E), - strand (B, D, F). Window width = 1mb. 45

Figure 12: Spearman correlations of ribonucleotides with annotated feature counts (A, B). Two-sided permutation p-values of correlations using adjacent swapping scheme (see Section 3.8) (C, D). Chromosome 1. + strand (A, C). - strand (B, D). Window width = 100kb. DNA-seq "depth" on $y$-axis title means "coverage". . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

Figure 13: DNA-seq coverage vs. chromosome position. Chromosome 1. + strand (left). - strand (right). Window width = 1mb. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

Figure 14: Spearman correlations between ribonucleotide counts and Fragmentase (F) DNA-seq coverage (A, B). Two-sided permutation test p-values for correlations using adjacent swapping permutation scheme (see Section 3.8) (C, D). Chromosome 1. + strand (A, C). - strand (B, D). Window width = 100kb. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

Figure 15: Histogram of smoothing-spline window widths. Chromosome 1. + strand (A). - strand (B). 47

Figure 16: Histogram of window densities for smoothing-spline windows. Chromosome 1. + strand (A). - strand (B). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

Figure 17: Hotspot windows using the smoothing-spline method. Hotspots are defined as windows with the top 1% ribonucleotide density ([total ribos in window] / [window width]) in the sample. "Gaps" are positions on the reference genome with incomplete information. Chromosome 1. + strand (A). - strand (B). . . . . . . . . . . . . . . . . . . . . . . . . . . . 49

# ABSTRACT

During the DNA replication process, ribonucleotides, the building blocks of RNA, may be occasionally incorporated in the newly synthesized DNA. DNA is primarily composed of deoxyribonucleotides and there exist cellular mechanisms for removing ribonucleotides from DNA, which may indicate ribonucleotide incorporation being a replication error. Further, an excess of these ribonucleotides in the genome may lead to genomic instability and has been implicated in human diseases. However, there are also hypotheses that suggest that ribonucleotides may be beneficial in certain circumstances. In this study we examine ribonucleotide incorporation in the human genome in several human cell types. While ribonucleotide incorporation has been studied in yeast, there has yet to be a systematic study of this phenomenon in the human genome. We analyze data obtained through a sequencing protocol that detects the positions of ribonucleotide incorporation in genome samples. We use mathematical analysis to detect hotspots and sequence patterns, as well as biologically relevant regions where such ribonucleotide incorporation appears nonrandom. Our analysis shows that the phenomenon is most commonly seen in regions that are GC rich and may be correlated with some gene regulatory segments. Further study will be needed to ascertain whether ribonucleotide incorporation has a specific biological function in the human genome.

# CHAPTER 1:
# INTRODUCTION

In this thesis we analyze a form of genomic data obtained through an experimental protocol know as *ribose-seq* [38]. Ribose-seq was designed to study a phenomenon known as *ribonucelotide incorporation* in genomes. The analyses we describe here form part of the first comprehensive study of ribonucleotide incorporation in human cells. The study is a collaboration between the Storici group (`https://storicilab.gatech.edu/`) at the Georgia Institute of Technology, and the Math-Bio group (`https://knot.math.usf.edu/`) at the University of South Florida. The work that formed the basis of this thesis was begun in spring 2022 and is ongoing.

The remainder of this thesis is organized as follows. We start with a brief description of sequencing technology, ribonucleotide incorporation, and our datasets in this chapter. Then Chapter 2 gives an overview of some of the major statistical tools that have been developed to analyze the type of genomic data studied here. Chapter 3 describes the methods we use in our own analysis. Chapter 4 presents the results of our analyses. Finally, Chapter 5 discusses our conclusions and proposes questions for further study.

## 1.1 The Human Genome and Sequencing

Physically, the human genome is a long, complicated molecule called DNA (deoxyribonucleic acid) that resides in cell nuclei. Though our understanding is far from complete, we know that the genome carries information for creating proteins that perform vital functions within cells. A fundamental characteristic of DNA is that it is a chain composed of four basic nucleotides (a type of molecule): A (adenine), C (cytosine), G (guanine), and T (thymine). Through a technology known as *DNA-sequencing* (or *DNA-seq*) we can take DNA molecules in biological samples and turn them into strings over the letters $\{$A, C, G, T$\}$, representing the chain of nucleotides in the molecules. This technology allows us to view the human genome as a long string, roughly three billion letters long. When referring to the length of genome segments, we interchangeably use the terms *nucleotides*, *base pairs*, or *bases*. For example, the previous statement about the length of the human genome may be equivalently stated as “three billion nucleotides”, “three billion base pairs”, or “three billion bases”.

In 2001, the first complete human reference genome was sequenced as the culmination of the Human Genome Project, which was started in 1990 [40]. In this study, we use an updated reference genome know as *GrCh38* or *hg38* [63]. Although the genome has been sequenced, the process of studying, annotating, and discovering the biological function of different parts of the genome is an ongoing process. Aiding this process have been the advances in sequencing technology. In particular, *high-throughput* sequencing technology was introduced in the mid 2000s [58], and allowed large amounts of genomic material to be sequenced from cell samples. The type of high-throughput sequencing technology studied here works by first physically fragmenting DNA samples into short strands, then using a sequencing machine to translate the strands into strings (i.e., a sequence of letters) known as *reads*. Thus, the raw output of the sequencing technology is not the fully assembled genome of the specimen, but a long list (usually millions) of relatively short reads (e.g., around 150 bases long for Illumina short-read sequencing [34]). The next stage is usually to *map* the reads to the reference genome using alignment tools, such as Bowtie2 [41], which try to find the most likely position in the genome that each read originated from. Although we use different cell types that may have genetic differences from the reference genome, this process mapping to the reference is still generally valid since any two human genomes are highly similar (less than 1% difference [1]). The output of this mapping process is a sequence of genome positions that each read maps to.

## 1.2 Ribose-Seq and Ribonucleotide Incorporation

Though we described DNA sequencing above, there are several related sequencing technologies that are similar and capture different genomic features of interest. We refer the reader to [58] for a review of high-throughput sequencing technologies. The ribose-seq [38] [2] technology that we study here was developed by the Storici Lab at Georgia Tech, and other collaborators. This protocol allows us to infer positions in the cell sample's genome that have been subject to ribonucletoide incorporation, which we describe in the following. Thus, each read sequenced in the ribose-seq protocol, when mapped to the genome, gives us the position of a single ribonucleotide.

A ribonucleotide is a molecule similar to a deoxyribonucleotide. Ribonucleotides are predominantly found in RNA (ribonucleic acid) while deoxyribonucleotides are predominantly found in DNA (deoxyribonucleic acid). Ribonucleotides are chemically different from deoxyribonucleotides, though the details are beyond the scope of this exposition. During DNA replication, either ribonucleotides or deoxyribonucleotide can be incorporated into the newly synthesized DNA strands. Ribonucleotides are far more infrequently incorporated than deoxyribonucleotides; several hundred or thousand time less frequently depending on the process involved [62] [80]. There are several hypotheses as to why ribonucleotides are incorporated into the genome and whether they are beneficial or detrimental to cells. An excess of ribonucleotides in the genome may lead to genome instability [62] [80]. In fact, ribonucleotides have been implicated in the human disease Aicardi-Goutières Syndrome, a serious genetic disorder with no known cure [62]. Further, there are mechanisms in cells for removing ribonucleotides incorporated during replication [62] [80], suggesting that they may simply be replication errors. However, there are also hypothesis of how ribonucleotides in the genome may serve a beneficial role [83] [80]. Ribonucleotide incorporation has been comprehensively studied in species such as yeast [3] [38] [14] [18] [57]. However, it has never been studied systematically in the human nuclear genome (i.e., the primary genome consisting of the 23 chromosomes). This thesis is part of the attempt to fill this gap in the literature by performing a study of several ribose-seq libraries of human nuclear DNA.

### 1.3 Data

The wet-lab part of the ribose-seq protocol, read mapping, and preprocessing is performed by the Storici group. For the detailed protocol, we refer the reader to the original paper [38] introducing the method. Simply stated, human cells are cultured in the laboratory and several steps are performed so that the sequenced reads allow us to infer the location of ribonucleotides. As part of this process a substance known as an *enzyme* is administered to the cell sample, which fragments the DNA into relatively short fragments of average length 350bp [38]. Due to the nature of the process, each DNA fragment in the sample will only be sequenced if it has a ribonucleotide at one of its ends. Once the DNA is fragmented, the samples are sequenced using Illumina sequencing technology, resulting in a table of strings representing the sequenced reads. As this is a *bulk* sequencing protocol, the reads, and thus the detected ribonucleotides, may come from different cells in the sample (by constrast *single-cell* technologies allow grouping reads by the individual cells they originated from). When the reads are mapped to the reference human genome, we get an indication of the relative number of ribonucleotides at each position on the genome, within all cells in the sample.

We can abstract the output of this process as a single vector of integers $\mathbf{y} = \{y_i\}_{i=1}^{N}$, where each $i$ indexes a distinct position on the genome, $y_i$ is the number of reads mapping to position $i$ (alternatively, the number of ribonucleotides detected at position $i$), and $N$ is the size of the genome. Note, $\mathbf{y}$ represents the output for a single sample. If there are multiple samples, we denote them with superscripts such as $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)}$.

To reduce the computational expense, we focus on a subset of the genome here known as *chromosome 1*. The human genome in cells is physically separated into 23 molecules known as *chromosomes*, of which chromosome 1 is the largest at approximately 249 million bases. The GRCh38 reference genome is likewise separated into 23 separate strings. Further, a DNA molecule is a double helix with two physically connected strings of nucleotides called the *positive* (or +) strand and the *negative* (or -) strand. Either strand contains exactly the same information since they are *reverse complements* of each other. That is, the negative strand is obtained from the positive strand by swapping A with T (and vice versa), C with G (and vice versa), and reversing the orientation of the strand. Due to this duality, only the positive strand of the human genome need be recorded in the reference genome. If a given read maps without modification to the reference genome, it is considered to originate from the positive strand. However, if the reverse complement of the read maps to the reference, then it is considered to originate from the negative strand. In addition to focusing on just a single chromosome, we also focus on each strand separately since they have different biological functions (e.g., different genes). Thus, each vector, $\mathbf{y}$, will usually refer to the data for a single sample, chromosome, and strand.

By itself, the vector $\mathbf{y}$ is a very general representation of the data. However, because $i$ indexes a genomic position, we have much additional information about this vector. For example, we may expect $\mathbf{y}$ to exhibit spatial dependence, which means that values $y_i$ and $y_j$ will have some dependence if $i$ and $j$ are close. Since we assume that $\mathbf{y}$ are observed counts for positions on a contiguous DNA molecule, the closeness of $i$ and $j$ should be related to their physical distance on the molecule. Since we expect the biological properties of physically near positions to be similar, it is reasonable to assume that ribonucleotide incorporation also follows this rule. A further property of the genome position, $i$, is the nucleotide sequence around it in the reference genome. Let $\mathbf{s} = \{s_i\}_{i=1}^N$ be the string of nucleotides in the reference genome corresponding to the positions of $\{y_i\}_{i=1}^N$. We may be interested in quantifying the distribution of nucleotides in the multiset $\{s_i | i = 1, \ldots, N \text{ and } y_i > 0\}$, which are the nucleotides that occur at position with at least one ribonucleotide. Similarly, we may be interested in whether there are certain nucleotide *motifs* that occur in the neighborhood of ribonucleotide positions, since biological function may be related to these motifs. Finally, there is a wealth of metadata associated with the GRCh38 reference genome known as *annotations*. Annotations may include information such as regions on the genome that are *genes* (segments the genome that are used as information for building proteins). Having such data will allow us to answer questions such as whether the occurrence of such functional elements are associated with the positions of ribonucleotides.

Although more than twenty combinations of cell types and fragmenting enzymes have been used to obtain ribose-seq libraries, we only focus on a subset of these here. Namely, we focus on six libraries that come from five different cell types, but have all been prepared with the *dsDNA Fragmentase* enzyme. The rationale behind focusing on Fragmentase is that it “provides random fragmentation, similar to mechanical methods”, according to the manufacturer [9]. Thus we expect it to provide more complete read coverage across the genome and reduce biases in downstream analyses. The libraries are listed in Table 1. Of note are the two *knockout* (KO) samples: FS329 and FS327. These samples have been genetically engineered so that their RNASEH2A gene is nonfunctional. RNASEH2A is one of the genes that code for the enzyme RNASE H2, which helps to remove ribonucleotides incorporated in the genome [16]. Mutations in this gene have been implicated in Aicardi-Goutières syndrome (AGS) [16]. Thus, we expect the KO samples to have significantly higher occurrence of ribonucleotides than the other samples.

### 1.4 Questions

There are several questions that guide our analyses:

- Is ribonucleotide abundance associated with the local nucleotide context?
- Can we identify *hotspots* where ribonucleotide abundance is relatively high?
- What statistical methods can we use to detect ribonucleotide hotspots on the genome?
- Can we identify any biological properties of the identified hotspots?
- What statistical methods can we use to verify the strength of our findings?

These questions are all aimed at revealing the biological function, or lack thereof, of ribonucleotide incorporation in human cells. In Chapter 2, we survey some of the answers to these questions in other organisms. The use of statistically rigorous methods will be important in this study because genomic datasets tend to be large and noisy, and have a danger of generating false positive findings.

**Table 1.** The ribose-seq samples and cell types used in this study.

| Sample | Cell type | Description |
|---|---|---|
| FS185 | CD4T | White blood cells (T lymphocyte) |
| FS197 | hESC-H9 | Human embryonic stem cell |
| FS198 | hESC-H9 | Human embryonic stem cell |
| FS326 | HEK293T-WT | Human embryonic kidney cell |
| FS329 | HEK293T-RNASEH2A-KO-T3-17 | Human embryonic kidney cell (RNASEH2A gene knockout) |
| FS327 | HEK293T-RNASEH2A-KO-T3-8 | Human embryonic kidney cell (RNASEH2A gene knockout) |

# CHAPTER 2:
# LITERATURE REVIEW

## 2.1 Previous Studies of Ribonucleotide Incorporation

As reviewed by Zhou et al. (2021) [83], ribonucleotide incorporation in whole genomes have previously been previously mapped and studied in Clausen et al. (2015) [14], Daigaku et al. (2015) [18], Koh et al. [38], and Reijns et al. (2015) [57]. These studies mapped ribonucleotide incorporation in different yeast species. They each used different ribonucleotide-sequencing protocols introduced by their respective publications (respectively, HydEnSeq, Pu-seq, ribose-seq, and emRiboSeq). The Python tool *Ribose-Map* [25] was introduced to streamline the process of mapping and analyzing ribonucleotide sequencing data from several protocols.

Previous studies have shown that ribonucleotide incorporation may have a biological preference for certain nucleotide motifs. In a study of the yeast genome by Balachander et al. (2020) [3], it was shown that ribonucleotide incorporation in yeast genomes has a preference for positions that have nucleotide C or G rather than A or T. Likewise, other nucleotide motifs were found to be associated with ribonucleotide incorporation, such as sequences made from short repeats, and dinucleotide (a sequence of two nucleotides) motifs immediately adjacent to the ribonucleotide. It has also been shown that different mechanisms for incorporating ribonucleotides may result in different nucleotide motifs being preferred for incorporation [83].

## 2.2 Multiple Change-Point Problem

The type of univariate, sequentially-ordered data that we study here has been studied before in other contexts, such as time series. As described in Section 1.3, we assume that our data is a vector of the form $\mathbf{y} = \{y_i\}_{i=1}^{N}$, where $y_i$ is a real number and $N$ is the number of measured data points in the sample. Importantly, we assume that the index $i$ is ordered in a meaningful way and that $y_i$ are not exchangeable. This is a general form that encompasses, for example, time series data (where $i$ is a discrete time index), genomics data (where $i$ is the position along the reference genome), and others. More specifically, we may want to account for *spatial dependence* between the $y_i$, meaning that values with nearby indices should be similar to each other. Since $\mathbf{y}$ is experimentally measured, we may consider it as a realization of a random vector $\mathbf{Y} = \{Y_i\}_{i=1}^N$ that describes the distribution of $\mathbf{y}$ under repeated data collection. One question that arises when studying such data is whether we can partition the index set $\{1, \ldots, N\}$ into segments $\{1, 2, \ldots, i_1\}, \{i_1 + 1, i_1 + 2, \ldots, i_2\}, \ldots, \{i_{n-1}, i_{n-1} + 1, i_{n-1} + 2, i_n\}$, such that on the $k^{\text{th}}$ segment the values $\{y_{i_k+1}, y_{i_k+2}, \ldots, y_{i_{k+1}}\}$ are relatively similar. In our case, we are interested in finding segments that are ribonucleotide hotspots. The problem can equivalently be stated as finding the indices, $\{i_1, i_2, \ldots, i_n\}$, where the distribution of $Y_i$ changes. This is known as the *multiple change-point problem* and approaches to solving this have been surveyed by Niu et al (2016) [48], which we follow here.

In the general multiple change-point problem, we are given a sequence of random variables $\{Y_i\}_{i=1}^N$ with distribution functions $\{F_i\}_{i=1}^N$. That is,

$$\Pr(Y_i \leq y) = F_i(y) \text{ for } y \in \mathbb{R}$$

The goal is to discover the *change points*, $i$, such that the distributions changes. That is, we search for a set of indices $\{i_k\}_{k=1}^K$ such that

$$\begin{gathered} i_1 = 1, i_K = N, \\ F_{i_k} = F_{i_k+1} = \cdots = F_{i_{k+1}-1} \text{ for } k = 1, \ldots, K-1 \\ F_{i_k} \neq F_{i_{k+1}} \text{ for } k = 1, \ldots, K-1 \end{gathered}$$

A common special case is to assume that the distributions are normal with equal variances but unknown nonrandom means $\boldsymbol{\mu} = \{\mu_i\}_{i=1}^N$. That is, $\forall i, Y_i \sim \mathrm{N}(\mu_i, \sigma^2)$. Two of the methods formulated in terms of the multiple change-point problem are *circular binary segmentation* and *sparse linear regression* (also surveyed in [48]).

### 2.3 Circular Binary Segmentation

Circular binary segmentation was introduced by Olshen et al. (2004) [49] to study genomic data and is an extension of the binary segmentation algorithm introduced by Sen and Srivastava (1975) [64]. The binary segmentation algorithm is a top-down procedure for recursively identifying change points until a stopping criteria is met. The method assumes that the random variables $\{Y_i\}_{i=1}^N$ are independent and normally distributed with constant known variance $\sigma^2$. For any subinterval of indices, $\{a, \ldots, b\} \subseteq \{1, \ldots, N\}$, we try to determine whether a change point exists in $\{a, \ldots, b\}$ with the following hypothesis test:

$$\mathrm{H}_0 : \mu_a = \cdots = \mu_b,$$
$$\mathrm{H}_1 : \exists i \in \{a, \ldots, b-1\}, \mu_a = \cdots = \mu_i \neq \mu_{i+1} = \cdots = \mu_b$$

Because the alternative is a union of $b - a$ hypotheses, we first compute the individual Z-test statistics as

$$Z_i = \frac{\frac{1}{i-a+1}\left(\sum_{j=i}^{a} Y_j\right) - \frac{1}{b-i}\left(\sum_{j=i+1}^{b} Y_j\right)}{\sigma^2\left(\frac{1}{i-a+1} + \frac{1}{b-i}\right)} \text{ for } i = a, a+1, \ldots, b-1$$

The final test statistic is then the maximum of the $b - a$ individual statistics:

$$Z_{(a,b)}^{\max} = \max_{i=a}^{b-1} Z_i$$

The null-hypothesis distribution of $Z_{(a,b)}^{\max}$ is estimated through Monte-Carlo simulation or other methods [64]. If the null hypothesis is rejected, we identify $i$ such that $Z_i = Z_{(a,b)}^{\max}$ is the change point. Then we apply the hypothesis test recursively to the index sets $\{a, \ldots, i\}$ and $\{i+1, \ldots, b\}$. The process ends when we have identified indices $1 \leq i_1 < \cdots < i_K < n$ as change points but fail to reject $\mathrm{H}_0$ for any of the subintervals $\{1, \ldots, i_1\}$, $\{i_1 + 1, \ldots, i_2\}$, ..., $\{i_K + 1, \ldots, n\}$.

Circular binary segmentation extends the binary segmentation procedure by testing additional alternative hypotheses at each stage of the change-point selection. Particularly, we test the union of the $\binom{b-a}{2}$ alternative hypotheses of the form

$$\mathrm{H}_1 : \mu_a = \cdots = \mu_{i-1} \neq \mu_i = \cdots = \mu_j \neq \mu_{j+1} = \cdots = \mu_b = \mu_a \text{ for } a \leq i < j \leq b$$

Note, that the "circular" part of the test comes from the fact that the alternative hypothesis includes the "wrap around" equality $\mu_b = \mu_a$. The motivation for this modification is that it allows greater sensitivity in finding small segments (indices $\{i, \ldots, j\}$ above) that are nested within the overall segment (indices $\{a, \ldots, b\}$), but may have a different mean. There are additional heuristics specified by Olshen et al. [49] to improve the practical usefulness of the algorithm.

### 2.4 Sparse Linear Regression

The next method, introduced by Huang et al. (2005) [32], is based on the sparse linear regression method know as the *Lasso*, introduced by Tibshirani (1996) [69]. The Lasso is a modification of the standard ordinary least-squares fitting process, which allows the procedure to simultaneously perform variable selection. The Lasso method seeks to minimize the loss function

$$L_\lambda(\boldsymbol{\beta}|\mathbf{y}) = \underbrace{\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2}_{L^2 \text{ error}} + \underbrace{\lambda\|\boldsymbol{\beta}\|_1}_{L^1 \text{ penalization}} ,$$

where $\mathbf{y} = \{y_1, \ldots, y_N\}$ is the observed response vector, $\mathbf{X} = \{x_{ij}\}_{i=1,j=1}^{N,M}$ is the model matrix, $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_M\}$ is the parameter vector, $\lambda$ is a hyperparameter controlling the smoothness (or variance) of the fitting procedure, and $\|\cdot\|_p$ is the $L^p$ norm defined by $\|\mathbf{u}\|_p = \left(\sum_{i=1}^N |u_i|^p\right)^{1/p}$ for $\mathbf{u} = \{u_1, \ldots, u_N\} \in \mathbb{R}^N$. The $L^1$ penalization term constrains the optimal parameter vector $\hat{\boldsymbol{\beta}}_\lambda = \operatorname{argmin} L_\lambda(\boldsymbol{\beta}|\mathbf{Y})$ by shrinking it towards $\mathbf{0}$ (the $M$-dimensional zero vector). The larger the hyperparameter $\lambda$, the more dominant the penalization term becomes and the more $\hat{\boldsymbol{\beta}}_\lambda$ is shrunk towards $\mathbf{0}$. The Lasso not only shrinks variables towards 0, but also performs variable selection by shrinking a subset of parameters to exactly 0. The size of this subset will again depend on $\lambda$, and thus $\lambda$ can also be interpreted as controlling the sparsity of the solution, with larger $\lambda$ resulting in sparser solutions. Because the Lasso can be formulated as a convex optimization problem, efficient algorithms exists for finding the solution.

To apply the Lasso for change-point detection with genomic data, the authors of [32] formulate the problem first in terms of a linear model. We again let $\mathbf{Y} = \{Y_1, \ldots, Y_N\}$ be the random vector for the observed signal at each position on the genome. We assume that $Y_i$ is distributed as

$$Y_i = \mu_i + \varepsilon_i$$

with unknown nonrandom mean $\mu_i$ and random error term $\varepsilon_i$ ([32] does not specify assumptions on $\varepsilon_i$). Note that this problem is trivial as a standard linear regression problem since the number of parameters equals the sample size and so the ordinary least squares solution (with standard assumptions on $\{\varepsilon_i\}_{i=1}^N$) is simply $\hat{\mu}_i = Y_i$. However, to incorporate spatial dependence, we constrain the smoothness of the means, where smoothness is measured by $\sum_{i=1}^{N-1} |\mu_{i+1} - \mu_i|$. Thus, the linear regression problem becomes

$$\hat{\boldsymbol{\mu}} = \operatorname{argmin}_{\boldsymbol{\mu}} \left[\sum_{i=1}^{N} (Y_i - \mu_i)^2 + \lambda \sum_{i=1}^{N-1} |\mu_{i+1} - \mu_i|\right]$$

This can be put into a more standard form for the Lasso by using the change of parameters given by

$$\beta_1 = \mu_1,$$
$$\beta_i = \mu_i - \mu_{i-1} \text{ for } 2 \le i \le N$$

Then we get

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \left[ \sum_{i=1}^{N} \left( Y_i - \sum_{j=1}^{i} \beta_j \right)^2 + \lambda \sum_{i=2}^{N} |\beta_i| \right]$$

This form with the penalization term, $\lambda \sum_{i=2}^{n} |\beta_i|$, is known as the *Lagrangian form* [29, p. 68]. However, it is equivalently expressed in [32] as the constrained regression problem

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left( Y_i - \sum_{j=1}^{i} \beta_j \right)^2 \text{ subject to } \sum_{i=2}^{N} |\beta_i| \le s,$$

where $s$ is the hyperparameter that controls the sparsity of the solution rather than $\lambda$. In this form, smaller $s$ gives a sparser solution. Each term $\beta_i$ can be interpreted as the change in the mean from position $i-1$ to $i$. The $L^1$ penalization shrinks some $\beta_i$ to 0 and thus the indices $i$ such that $\hat{\beta}_i \neq 0$ may be interpreted as the change points. The sparsity of the solution induced by the hyperparameter $\lambda$ or $s$ can also be interpreted as controlling the amount of spatial dependence between adjacent positions, with sparser solutions associated with greater spatial dependence. This method was originally introduced for detecting DNA copy-number variation and contains additional post-processing steps to fine tune the results and evaluate their statistical significance [32].

### 2.5 Hidden Markov Model Classifier

Yet another approach to account for the spatial dependence is to use a *hidden Markov model* (HMM) as proposed by Fridyland et al. (2004) [23]. As in Section 2.4, this approach was also developed for analyzing DNA copy-number variation data and uses as input the vector of log2 fold changes of copy number between a test and reference sample, which we denote $\mathbf{y} = \{y_i\}_{i=1}^{N}$. Note, these are continuous values, which contrasts with our ribonucleotide count values.

A HMM is a parametric probability distribution on a set $\mathcal{Y}^N \times \mathcal{S}^N$, where $\mathcal{Y}$ is any set and $\mathcal{S} = \{1, \ldots, K\}$ is a set of $K$ states for some $K > 0$. Potential choices for $\mathcal{Y}$ may be $\mathbb{R}^p$ with a multivariate normal distribution, $\mathbb{R}$ with a continuous distribution, $\{0, 1, \ldots, \}$ with a count distribution, or a finite set with a categorical distribution. Let $(\mathbf{Y}, \mathbf{S}) = \left(\{Y_i\}_{i=1}^N, \{S_i\}_{i=1}^N\right)$ denote a random vector representing this space. The components of $\mathbf{Y}$ take on values in $\in \mathcal{Y}^N$, and are known as the *emissions* of the HMM and are the observed values of the model. The components of $\mathbf{S}$ take on values in $\mathcal{S}^n$, and are unobserved latent variables or *hidden states*. A key feature of the HMM is that it models dependence between the $Y_i$. As we show below, this is indirectly defined through the dependence between adjacent states $S_i, S_{i+1}$, and the dependence of each emission $Y_i$ on the state $S_i$ that it is emitted from. We often want to use the observed emissions $\mathbf{y}$ to make inferences about the most likely hidden states $\mathbf{s}$.

Here, we follow the description of HMMs given by Rabiner (1989) [56]. The parameters of a HMM may be specified as $\boldsymbol{\lambda} = \{K, N, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$, described as follows.

- $K$: the number of distinct hidden states.
- $N$: the number of emissions observed. That is, the length of both $\mathbf{y}$ and $\mathbf{s}$.
- $\mathbf{A} = \{A_{ij}\}_{i=1,j=1}^{K,K}$: a $K \times K$ *transition matrix* such that $\forall t \in \{1, \ldots, n-1\}, \forall \mathbf{s} \in \mathcal{S}^n, A_{ij} = \Pr(S_{t+1} = j | S_t = i)$. $\mathbf{A}$ is properly defined if and only if $\forall (i,j), A_{ij} \geq 0$ and $\forall i, \sum_{j=1}^K A_{ij} = 1$.
- $\mathbf{B} = \{B_1, \ldots, B_K\}$: emission distributions. Each $B_k : Y \to [0, \infty)$ is a probability density function, and is interpreted as the emission distribution when the HMM is in state $k$. That is, $\Pr(Y_i = x | S_i = k) = B_k(x)$.
- $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$: the initial probabilities for each state. That is, $\pi_k = \Pr(S_1 = k)$. $\boldsymbol{\pi}$ is properly defined if and only if $\forall k, \pi_k \geq 0$ and $\sum_{k=1}^K \pi_k = 1$.

Two fundamental properties of HMMs are described in Yoon (2009) [82]. One is the *Markov property* of HMMs, which states

$$\Pr(S_{i+1} = s_{i+1} | S_i = s_i, S_{i-1} = s_{i-1}, \ldots, S_1 = s_1) = \Pr(S_{i+1} = s_{i+1} | S_i = s_i) = A_{s_i s_{i+1}}$$

This means that all the information for determining the value of $S_{i+1}$ is contained in the value of $S_i$. Another property, is the conditional independence of the emissions, stated as

$$\Pr(Y_i = y_i | S_i = s_i, S_{i-1} = s_{i-1}, Y_{i-1} = y_{i-1}, \ldots, S_1 = s_1, Y_1 = y_1) = \Pr(Y_i = y_i | S_i = s_i) = B_{s_i}(y_i)$$

This means that all the information for determining the value of $Y_i$ is contained in the value of $S_i$. Using these properties, we can derive the probability density function of the HMM as

$$\Pr(\mathbf{Y} = \mathbf{y}, \mathbf{S} = \mathbf{s}) =$$
$$\Pr(S_1 = s_1) \prod_{i=1}^{n-1} \Pr(S_{i+1} = s_{i+1} | S_i = s_i) \Pr(Y_i = y_i | S_i = s_i) =$$
$$\pi_{s_1} \left( \prod_{i=1}^{n-1} A_{s_i s_{i+1}} \right) \left( \prod_{i=1}^{n} B_{s_i}(y_i) \right)$$

As a generative process, the HMM can be thought of as first choosing an initial state according to $\boldsymbol{\pi}$, transitioning to $n - 1$ new states according to $\mathbf{A}$, and emitting elements according to $\mathbf{B}$ at each state.

In the application of HMMs to model DNA copy-number variation by [23], the states are interpreted as different levels of variation in the log2 fold change $y_i$ between the test and control copy number. For example, one state may capture copy numbers that are similar to the reference ($y_i \approx 0$), other states may capture copy numbers that are less than the reference ($y_i < 0$), and other states may capture copy numbers that are greater than the reference ($y_i > 0$). Several heuristics are used by the authors for model selection (that is, choosing the number of states) and initializing the model parameters. For emission distributions, the authors make use of normal distributions since their observations are continuous values. However, as we model count data, we may employ distributions such as the Poisson and negative-binomial.

HMMs may be fit to observed data using an *expectation-maximization* (EM) algorithm. Once the fitted models is obtained, we can use it to perform inferences, such as determining the most likely sequence of states $\hat{s} = \{\hat{s}_1, \ldots, \hat{s}_N\}$. This allow us to infer the change points, where $\hat{s}_i \neq \hat{s}_{i+1}$. It may also allow us to assign an interpretable class to the segmented regions if the states themselves are interpretable. However, a potential downside to the HMM approach is the need to specify the number $K$ of hidden states beforehand. However, procedures for selecting the initial number of hidden states, as well as pruning excess states from a fitted model is discussed further in [23].

### 2.6 AIC/CV Window Width Selection

Another method proposed by Gusnanto et al. (2014) [28] seeks to determine a uniform window size to partition a chromosome for aggregating (or *binning*) data and determining outlier (or hotspot) regions. This method approaches the problem using the statistical model selection tools *AIC* (Akaike's information criterion) and *CV* (cross validation). Both criteria are ways to estimate the model log-likelihood on a new dataset (rather than the dataset used to fit the model). Again, let $\mathbf{y} = \{y_i\}_{i=1}^{N}$ be the observed data. Also, assume that $\mathbf{y}$ is count data: $\forall i, y_i \in \{0, 1, \ldots\}$. Let $Y = \sum_{i=1}^{N} y_i$. The model assumption is that

$\mathbf{y}$ represents $Y$ independent samples from a categorical distribution with $N$ categories and probabilities $\mathbf{p} = \{p_i\}_{i=1}^N$. That is,

$$\Pr(\mathbf{y}|\mathbf{p}) = \prod_{i=1}^{N} p_i^{y_i}$$

In our case, the $i^{\text{th}}$ category represents the $i^{\text{th}}$ position on the genome segment being considered. This gives the following formulas for the likelihood, $L$, and log-likelihood, $LL$, given the probabilities $\mathbf{p} = \{p_i\}_{i=1}^N$:

$$L(\mathbf{p}|\mathbf{y}) = \prod_{i=1}^{N} p_i^{y_i}, \tag{2.1}$$

$$LL(\mathbf{p}|\mathbf{y}) = \sum_{i=1}^{N} y_i \log(p_i) \tag{2.2}$$

Since $\mathbf{p}$ are probabilities we also have the constraint

$$\forall i, p_i \geq 0 \text{ and } \sum_{i=1}^{N} p_i = 1 \tag{2.3}$$

The probability vector $\mathbf{p}$ are the parameters of our model that we make inferences about. To do so, an additional assumption is made that a *histogram* will be a good approximation to $\mathbf{p}$. This assumption can be though of as a means to account for spatial dependence in the model, since histograms are piecewise constant functions.

For a given window width $w \in \{1, \dots, N\}$, a histogram is simply a sequence that is constant on each consecutive window of width $w$. More formally, it is the set of sequences $\mathcal{H}_w$ such that for all $\mathbf{p} = \{p_1, \dots, p_N\} \in \mathcal{H}_w$, we have $\forall i, p_i = p_{\lceil i/w \rceil}$. Note, $w$ controls the complexity of our model. In fact, there are exactly $\lceil N/w \rceil$ parameters since the values $\{p_w, p_{2w}, \dots, p_{\min(\lceil N/w \rceil w, n)}\}$ fully determine $\mathbf{p}$. At one extreme, $w = 1$, we have $N$ parameters, one for every index. At the other extreme, $w = N$, we have only one parameter. To find the optimal window width, denoted $\hat{w}$, that avoids overfitting or underfitting the data, we use the AIC and CV log-likelihoods.

First, for any fixed $w$, we can determine the unique maximum-likelihood estimate (MLE) of $\mathbf{p}$, denoted $\hat{\mathbf{p}}_w$. This estimate may be found by optimizing the right-hand side of Equation 2.2 with the constraint 2.3. For ease of notation, we define the windows by $W_i = \{w(i-1)+1, \dots, wi\}$ for $i \in \{1, 2, \dots\}$. Let the count in the $i^{\text{th}}$ window be given by

$$v_i = \sum_{\substack{j \\ j \in W_i \\ j \leq N}} y_j$$

Then the explicit MLE, $\hat{\mathbf{p}}_w$, is given by

$$\forall i, \hat{p}_{wi} = \frac{v_i}{Yw}$$

Note, this solution is exact when $N$ is a multiple of $w$ and is an approximation otherwise. The AIC is defined in terms of the log-likelihood (2.2) and number of parameters ($\lceil N/w \rceil$) in the model:

$$AIC_w = LL(\hat{\mathbf{p}}_w | \mathbf{y}) - \lceil N/w \rceil$$

Note, this definition differs from that given by [28], since we use the form that estimates the log-likelihood on new data rather than the conventional definition of AIC. To define the CV, we must introduce additional notation. Let $\left\{\mathbf{y}^{(i)}\right\}_{i=1}^{Y}$ be the decomposition of $\mathbf{y}$ into $Y$ independent observations. That is, for $j = 1, \ldots, N$, for $i = \left(\sum_{k=1}^{j-1} y_k\right) + 1, \ldots, \left(\sum_{k=1}^{j} y_k\right)$, define

$$y_l^{(i)} = \mathbf{1}\{l = j\} \text{ for } l = 1, \ldots, N$$

Let $\hat{\mathbf{p}}_w^{(-i)}$ be the MLE for the observations $\mathbf{y} - \mathbf{y}^{(i)}$ (all observations except the $i^{\text{th}}$). Then the CV is defined by

$$CV_w = \sum_{i=1}^{Y} \log \Pr(\mathbf{y}^{(i)} | \hat{\mathbf{p}}_w^{(-i)})$$

The AIC and CV can be algebraically simplified into

$$AIC_w = \left( \sum_{k=1}^{\lceil N/w \rceil} y_k \log(y_k) \right) - N \log(Nw) - \lceil N/w \rceil \tag{2.4}$$

$$CV_w = \left( \sum_{k=1}^{\lceil N/w \rceil} y_k \log(y_k - 1) \right) - N \log((N-1)w) \tag{2.5}$$

However, the implemented formula is slightly different (see Section 3.3) to account for $i$ such that $y_i \leq 2$, which may make some values in the definitions above undefined. The corresponding optimal window width is given by minimizing these quantities over $\hat{w}$:

$$\hat{w} = \text{argmin}_w AIC_w$$

or

$$\hat{w} = \text{argmin}_w CV_w$$

Gusnanto et al. (2014) [28] provide an implementation, which we referred to in our own implementation, of their method in the R package *NGSoptwin* available at `http://www1.maths.leeds.ac.uk/~arief/R/win/`.

### 2.7 Smoothing Spline Method

A final method that we consider here, introduced by Beissinger et al. (2015) [5] is based on the idea of smoothing the data using *cubic smoothing splines*. This method may also be considered a way of dealing with spatial dependence by smoothing the data in a way that accounts for spatial position. In this method, we assume that we observe points $\{(x_i, y_i)\}_{i=1}^N$ from the model

$$y_i = f(x_i) + \varepsilon_i$$

where $f \in C^2([a,b])$ is a function with continuous second derivative, $-\infty < a < b < \infty$, and $\boldsymbol{\varepsilon} = \{\varepsilon_i\}_{i=1}^N$ are uncorrelated random errors with equal variance $\sigma^2$. That is,

$$\forall i, \text{E}(\varepsilon_i) = 0,$$
$$\forall (i,j), \text{E}(\varepsilon_i \varepsilon_j) = \sigma^2 \mathbf{1}\{i = j\}$$

We also assume $\forall i, x_i \in [a,b]$ and $\forall i, x_i < x_{i+1}$. Unlike the other methods used here, the smoothing spline approach explicitly accounts for potentially unequal spacing between the observed values, $\mathbf{y} = \{y_i\}_{i=1}^N$, by taking as input the position of each observation, $\mathbf{x} = \{x_i\}_{i=1}^N$.

The goal of this approach is to use a good approximation to the unknown function $f$ using the observed points. To do so, a class of functions known as *cubic splines* are used. A cubic spline is a piecewise cubic polynomial on an interval $[a,b]$. More precisely, it is a function $g : [a,b] \to \mathbb{R}$ satisfying the following.

- There exists a partition, $\{a_i\}_{i=0}^K$, such that $a = a_0 < a_1 < \cdots < a_K = b$, and $g$ is a cubic polynomial on $[a_i, a_{i+1}]$ for all $i$.

- $g'$ and $g''$ exist and are continuous on $[a,b]$.

For any $\lambda > 0$, the solution, $\hat{f}_\lambda$, to the optimization problem

$$\hat{f}_\lambda = \text{argmin}_g \left[ \sum_{i=1}^{N} (g(x_i) - y_i)^2 + \lambda \int_a^b (g''(t))^2 \, dt \right]$$
$$\text{for } g \in C^2([a, b]),$$

is a cubic spline, where $C^2([a, b])$ is the set of all functions with continuous second derivative. The loss function being minimized above has a natural interpretation. The term $\sum_{i=1}^{N} (g(x_i) - y_i)^2$ captures how much $g$ deviates from the observed points $\{(x_i, y_i)\}_{i=1}^{N}$. The term $\lambda \int_a^b (g''(t))^2 \, dt$ captures the *nonsmoothness* of $g$. That is, large values of $g''$ indicates that $g$ has higher curvature and deviates more significantly from a straight line. The term $\lambda$ is a hyperparameter that controls the tradeoff between how smooth the solution is versus how close it comes to interpolating the points. In fact, as $\lambda \to +\infty$, $\hat{f}_\lambda$ approaches the least-squares line through the points $\{(x_i, y_i)\}_{i=1}^{N}$. On the other hand, as $\lambda \to 0$, $\hat{f}_\lambda$ approaches the unique natural spline that interpolates the points $\{(x_i, y_i)\}_{i=1}^{N}$ [26]. Thus, $\lambda$ must be chosen to balance overfitting and underfitting.

An efficient method for selecting the optimal $\lambda$ is based on the idea of cross-validation, which uses the data to both fit the model and evaluate the quality of the fit. For each $i \in \{1, \ldots, N\}$, let $\hat{f}_\lambda^{(-i)}$ denote the fitted cubic-spline with the observation $i$ deleted. Formally, this is

$$\hat{f}_\lambda^{(-i)} = \text{argmin}_g \left[ \sum_{\substack{j=1 \\ j \neq i}}^{N} (g(x_j) - y_j)^2 + \lambda \int_a^b (g''(t))^2 \, dt \right]$$

Then the cross-validation error is defined as

$$V_\lambda = \frac{1}{N} \sum_{i=1}^{N} \left( \hat{f}_\lambda^{(-i)}(x_i) - y_i \right)^2$$

Since $\hat{f}_\lambda^{(-i)}$ has been fit without using the point $(x_i, y_i)$, each of the terms $\left( \hat{f}_\lambda^{(-i)}(x_i) - y_i \right)^2$ is an estimate of the squared prediction error in the cubic spline fit. Then, $V_\lambda$ is the average of these estimates. Thus, to pick the values of $\lambda$ with the smallest prediction error, we use $\hat{\lambda} = \text{argmin}_\lambda V_\lambda$. This procedure for selecting $\lambda$ was introduced by Wahba and Wold (1975) [71] [72].

To obtain stronger conditions of convergence, Craven and Wahba (1978) [15] introduced *generalized cross-validation* (GCV) for selecting $\lambda$. To defined GCV, we first express the fitted values, $\left\{ \hat{f}_\lambda(x_i) \right\}_{i=1}^{N}$, as

$$\left\{ \hat{f}_\lambda(x_i) \right\}_{i=1}^{N} = \mathbf{A}_\lambda(\mathbf{x})\mathbf{y}$$

where $\mathbf{A}_\lambda(\mathbf{x}) = \{A_{\lambda,ij}(\mathbf{x})\}_{i=1,j=1}^{N,N}$ is the $N \times N$ matrix of coefficients, dependent on $\mathbf{x}$ and $\lambda$, that determines $\left\{\hat{f}_\lambda(x_i)\right\}_{i=1}^{N}$ as a linear function of $\mathbf{y}$. Such a matrix always exists [26, sec. 2.3]. Then the GCV error $G_\lambda$ is defined as

$$G_\lambda = \frac{1}{N} \left\|(\mathbf{I} - \mathbf{A}_\lambda(\mathbf{x})\mathbf{y}\right\|^2 \Bigg/ \left[\frac{1}{N}\mathrm{Tr}(\mathbf{I} - \mathbf{A}_\lambda(\mathbf{x}))\right]^2 ,$$

where $\mathbf{I}$ is the $N \times N$ identity matrix and $\mathrm{Tr}(\cdot)$ is the trace of a matrix. The GCV error can also be expressed as

$$G_\lambda = \sum_{i=1}^{N} w_{\lambda,i}(\mathbf{x}) \left(\hat{f}_\lambda^{(-i)}(x_i) - y_i\right)^2$$

where

$$w_{\lambda,i} = \left[(1 - A_{\lambda,ii}(\mathbf{x})) \Bigg/ \frac{1}{N}\mathrm{Tr}(\mathbf{I} - \mathbf{A}_\lambda(\mathbf{x}))\right]^2$$

Thus, the GCV error is a weighted version of the CV error with weights $\mathbf{w}_{\lambda,i}(\mathbf{x}) = \{w_{\lambda,i}(\mathbf{x})\}_{i=1}^{N}$. The GCV has been shown by [15] to have stronger convergence guarantees than CV, under certain assumptions.

Once we choose $\hat{\lambda}$ and fit the smoothing spline to obtain $\hat{f}_{\hat{\lambda}}$, we can use the fitted spline to segment the interval $[a, b]$. To do so, we simply take the set of point $\{a, b\} \cup F$, where $F$ is the set of inflection points of $\hat{f}_\lambda$. That is,

$$F = \left\{x \middle| \hat{f}''_{\hat{\lambda}} \text{ changes signs at } x\right\}$$

The inflection points are used because the authors have observed that there is usually local minima or maxima between two consecutive inflection points (although this is not guaranteed). Conversely, let $a_1, b_1$ be local extrema of $\hat{f}_\lambda$ such that $a \leq a_1 < b_1 \leq b$. Then there must always be an inflection point in $(a_1, b_1)$, unless $\hat{f}_\lambda$ is a straight line on $[a_1, b_1]$. The rationale is that segments capturing a local maxima will be of interest, since a high peak could indicate a hotspot.

The authors of [5] implement their method in the R package *GenWin* [4], which we based our own implementation on.

# CHAPTER 3:
# METHODS

To detect ribonucleotide hotspots and study their association with nucleotide composition, we employ several of the methods outlined in Chapter 2, as well as other custom approaches. For the methods used from the literature, we describe our custom implementations and how they may differ from that of the original authors.

## 3.1 Programming Tools and Packages

For all the analyses described in this thesis, we used the R programming language [54], the suite of bioinformatics packages known as *Bioconductor* [33] (including *Biostrings* [50], *GenomicRanges* [43], *IRanges* [43], *annotatr* [13], *plyranges* [45], and *rtracklayer* [42]) and the suite of data processing packages known as *tidyverse* [79] (including *lubridate* [27], *ggplot2* [74], *forcats* [73], *dplyr* [78], *stringr* [75], *tidyr* [76], *readr* [77], *purrr* [31], and *tibble* [47]). Other miscellaneous packages include *xtable* [17] for generating tables. Additional packages are cited in the following sections where appropriate.

## 3.2 Preliminary Processing and Binning

As described in Section 1.3, our raw data after mapping and filtering can be thought of as a sequence $\left\{\left(x_i^{(k)}, y_i^{(k)}\right)\right\}_{i=1}^{N^{(k)}}$, where $x_i^{(k)} \in \{1, 2, \dots\}$ are unique genome coordinates, $y_i^{(k)} \in \{1, 2, \dots\}$ is the number of reads mapping to coordinate $x_i^{(k)}$, $N^{(k)}$ is the number of unique coordinates that have at least one read, and $k$ is the index for each unique combination of sample, chromosome, and strand. In the following, we drop the index $k$ and assume that the sequence $\{(x_i, y_i)\}_{i=1}^{N}$ represents the data for a single sample, chromosome, and strand.

A complication of having both positions $x_i$ and read counts $y_i$ is the potentially uneven spacing of the coordinates $x_i$. These is also a lack of information of the explicit zeros in the data (i.e., positions on the genome that are not mapped to by any reads) since $\forall i, y_i > 0$. As we will see in Section 4.1, some of the datasets were very sparse, with $10^{-3}$ or less ribonucleotides per base, meaning the vast majority of the genome has zero reads mapping. To mitigate both of these concerns, we perform the following binning on the data:

- Pick a window size $w \in \{1, 2, \dots, C\}$, where $C$ is the length of the chromosome.

- Compute the binned counts, $\tilde{\mathbf{y}} = \{\tilde{y}_i\}_{i=1}^{\lceil C/w \rceil}$, as

$$\tilde{y}_i = \sum_{\substack{j, \\ w(i-1)+1 \leq x_j \leq wi}} y_j$$

- Define the corresponding coordinates, $\tilde{\mathbf{x}} = \{\tilde{x}_i\}_{i=1}^{\lceil C/w \rceil}$, as

$$\tilde{x}_i = (i-1)w + 1$$

  We usually do not explicitly use $\tilde{\mathbf{x}}$ in our analyses.

Each $\tilde{y}_i$ can be interpreted as the aggregate count in the window $\{(i-1)w+1, \dots, iw\}$. Thus each $\tilde{y}_i$ represents the number of ribonucleotides in equally spaced windows, and windows with zero reads are automatically accounted for. By choosing a sufficiently large $w$, we may reduce the noise in our data by pooling nearby observations, at the expense of loosing precision in the $x$-coordinates. As suggested by Gusnanto et al. (2014) [28], the choice of window size should be balanced to improve the accuracy of read density in each window without sacrificing too much precision in the downstream inferences. In Section 4.2, we implement their method for estimating the optimal window size. In the following sections, the vector $\mathbf{y} = \{y_i\}_{i=1}^{N}$ usually represents binned data (i.e., $\tilde{\mathbf{y}}$ above), unless specifically mentioned.

Before binning, we also remove ribonucleotides that are in GRCh38 *repeat regions*. Repeats regions are regions on the reference genome that are highly repetitive, making it difficult to uniquely identify the position of reads originating from such positions (since such a read may map equally well to multiple regions). The repeat region files, which have been created using the *RepeatMasker* [65] software, are obtained from the University of California Santa Cruz website (https://genome.ucsc.edu/).

### 3.3 AIC and CV Windows Width Selection

The process for determining the optimal window width using the AIC and CV is implemented in the R package *NGSoptwin* [28], which differs slightly from the method described by Gusnanto et al. (2014) [28] (see Section 2.6). Assume the following definitions.

- Let $w$ be the window width.

- Let $N_w = \lceil C/w \rceil$ be the number of windows of width $w$, where $C$ is the length of the chromosome.

- Let $\mathbf{y} = \{y_i\}_{i=1}^{N_w}$ be the binned ribonucleotide counts using $w$-width windows (see Section 3.2).

- Let $N_w^* = |\{i|y_i \geq 2\}|$ be the number of entries of $\mathbf{y}$ with value greater than 1.
- Let $\mathbf{y}^* = \{y_k^*\}_{k=1}^{N_w^*} = \{y_{i(k)}\}_{k=1}^{N_w^*}$ be the subvector of $\mathbf{y}$ consisting of indices $\{i(k)\}_{k=1}^{N_w^*}$ such that $\forall k, y_{i(k)} \geq 2$.
- Let $Y^* = \sum_{i=1}^{N_w^*} y_i^*$ be the total number of ribonucleotides in windows with at least 2 ribonucleotides.
- Let $h = 1/N_w$ be approximately proportional to the window width.

Then the AIC and CV log-likelihood are defined as

$$\begin{aligned} AIC_w &= \left[\sum_{i=1}^{N_w^*} y_i^*(\log(y_i^*) - \log(Y^*h))\right] - N_w \\ CV_w &= \sum_{i=1}^{N_w^*} y_i^*(\log(y_i^* - 1) - \log((Y^* - 1)h)) \end{aligned}$$

We require the auxiliary definitions, $N_w^*$ and $\mathbf{y}_w^*$, because the definition of CV is undefined if any entries have value less than 2. (Author's note: It is not clear why the definition of AIC also uses $N_w^*$ in *NGSoptwin*, since AIC is well defined even if $y_i = 1$ for any $i$). It is also not clear why $h$ is used in place of $w$, though, presumably, this should not affect the results too much.)

### 3.4 Distribution Fitting

One of our central question is whether the occurrence of ribonucleotides in the genome is "random" or there is some "nonrandom" pattern to their occurrence. More precisely, we may ask whether the ribonucleotides detected in our samples is modeled well by a simple random generative process, which may indicate that there is no pattern. A basic null hypothesis we may make is that the ribonucleotides are uniformly distributed throughout the genome. Though simple to state, such a hypothesis is difficult to test due to various confounding factors that may occur in the data-gathering process. The first is that the GRCh38 reference genome is incomplete and contains *gaps* such as in the centromeres (the center of the "X" shape a chromosome takes when coiled up in the nucleus) of nuclear chromosomes. These gaps are regions in the reference genome that are filled with the letter N (indicating an unknown nucleotide) instead of A, C, G, or T, which means that no reads can map to these regions. The windows that are wholly in a gap will always have read count of zero. This may be partially resolved by ignoring such windows or simply ignoring windows with zero reads. Another problem is that the enzymes used in the ribose-seq protocol to fragment the genome do not fragment DNA uniformly, but select specific nucleotide patterns [81]. This may result in biased estimates of the ribonucleotides distribution at different locations, though this may be less of an issue for the Fragmentase enzyme used in this study, since it is supposed to fragment DNA uniformly [9]. Finally, as we explain in more detail in Section 3.9, the sequencing protocol may itself introduce biases for certain regions of the genome. Despite these issues, we attempt to analyze the distribution of the raw binned reads counts as a form of exploratory data analysis.

There are several types of distributions that we consider here. The first are simple parametric distributions, where were assume that each position is independent and identically distributed (iid). These include the Poisson and negative-binomial distributions. These are common count distributions that have been used to model genomic data in different contexts [22]. The Poisson distribution is especially appealing because it has a simple interpretation: it approximates the expected distribution generated by uniform random sampling across all positions. On the other hand, the negative-binomial is a generalization of the Poisson that has an additional parameter to allow more flexibility. While this iid assumption may serve as a starting point, in practice it may be highly unrealistic, even if we omit the gap regions.

Because of the difficulty in modeling the zeros in the data, we tried different variations of the distributions, such as zero-truncated (nonzero values only) and zero-inflated. To define these distributions, assume that $f_{\boldsymbol{\theta}} : \{0, 1, \dots\} \to [0, \infty)$ is the Poisson or negative-binomial probability mass function (PMF) with parameter vector $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_K\}$. That is, $f_{\boldsymbol{\theta}}(x) = \Pr(X = x)$, for a random variable $X$ that follows the selected distribution. The zero-truncated variant is the distribution with PMF

$$f^{(\mathrm{ZT})}_{\boldsymbol{\theta}}(x) = \begin{cases} \frac{f_{\boldsymbol{\theta}}(x)}{1-f_{\boldsymbol{\theta}}(0)} & \text{if } x > 0, \\ 0 & \text{if } x = 0 \end{cases}$$

The zero-truncated Poisson distribution and an application has been described by Plackett (1953) [53]. This distribution may be described as the standard Poisson conditioned on $X > 0$. Thus, by using it, we assume that we are only able to observe nonzero values from the distribution and can only model the conditional part.

The zero-inflated version is a mixture of a point-mass at zero and $f_{\boldsymbol{\theta}}$. If $p$ is the probability of observing the zero point-mass, the zero-inflated distribution has PMF

$$f^{(\mathrm{ZI})}_{\boldsymbol{\theta},p}(x) = \begin{cases} p + (1-p)f_{\boldsymbol{\theta}}(0) & \text{if } x = 0, \\ (1-p)f_{\boldsymbol{\theta}}(x) & \text{if } x > 0 \end{cases}$$

This zero-inflated Poisson has been described by Lambert (1992) [39]. The motivation behind using this variant is that there may be a different process that governs the generation of the zeros due to the point-mass component and those due to $f_{\boldsymbol{\theta}}$. For instance, there may be regions on the genome for which it is impossible to observe ribonucleotides (zero point-mass component), while for the remaining regions there may be a small constant probability of observing a ribonucleotide (Poisson component). This contrasts with the zero-truncated distribution where we assume no knowledge of the zeros.

Finally, although we do not show the results here, a more complicated distribution is the hidden Markov model (HMM). This model allows us to model each position on the genome as a mixture of distributions, using latent states (or *hidden states*) that determine which distribution is observed. Moreover, the distributions of the hidden states form a Markov chain. HMMs are described in more detail in Section 2.5.

To fit the distributions we use two R packages: *gamlss* [59] [67] [66] (non-HMM distributions) and *depmixS4* [70] (HMM distributions). We use the `gamlssML` function in the *gamlss* package to fit the Poisson and negative-binomial models (or their zero-inflated or zero-truncated variants). We use the `makeDepmix`, `fit`, and `posterior` functions from the *depmixS4* package to fit HMM models and infer the most likely hidden states in the fitted model.

### 3.5 Determination of Hotspots With Distributions

The determination of ribonucleotide hotspots in our data can be thought of as a hypothesis testing problem. Our null hypothesis is that our data originated through a random process described by one of the parametric iid models described in Section 3.4. Let $f_{\hat{\boldsymbol{\theta}}} : \{0, 1, 2, \dots\} \to [0, \infty)$ be a parametric distribution, where $\hat{\boldsymbol{\theta}}$ denotes the fitted parameters using the observed values $\{y_i\}_{i=1}^N$. We may then obtain p-values, $p_i = \Pr(Y_i \geq y_i | \hat{\boldsymbol{\theta}}) = \sum_{x=y_i}^{\infty} f_{\hat{\boldsymbol{\theta}}}(x)$, giving the probability of observing a value equal to or more extreme than $y_i$ under the null hypothesis. To determine p-value cut-offs for classifying positions as hotspots (those that would be highly unlikely under the null hypothesis) we may use a criteria for multiple hypothesis testing. Let $\alpha$ be the chosen level of significance (e.g., $\alpha = 0.05$). To control the familywise error rate (FWER) at $\alpha$, we may use the Bonferroni criteria $p_i < \alpha/N$. Alternatively, we could control the false-discovery rate (FDR) by using the Benjamini-Hochberg [6] criteria $p_i \leq p_{(i^*)}$, where $\{p_{(i)}\}_{i=1}^N$ are the order statistics of $\{p_i\}_{i=1}^N$ and $i^* = \max\{i | p_{(i)} < \frac{i}{N}\alpha\}$.

### 3.6 Smoothing-Spline Windows

To implement the smoothing-spline method introduced by Beissinger et al. (2015) [5], we start with the R implementation provided by the authors in the R package *GenWin* [4]. In this package, the smoothing spline is fit with the function `smooth.Pspline` from the R package *pspline* [35]. We use the argument `method = 3` in this function to use the GCV criterion (see Section 2.7) for estimating the optimal smoothing parameter. We use the argument `norder = 2` to fit a cubic spline.

The raw input to the `smooth.Pspline` function is the vector of ribnucleotide counts, $\{y_i\}_{i=1}^N$, and positions, $\{x_i\}_{i=1}^N$. The initial window width, $w$, used to create these inputs is determined in terms of the total number of ribonucleotides detected in the sample, $Y = \sum_{i=1}^N y_i$. We define it by $w = \left\lceil \frac{rC}{Y} \right\rceil$, where $C$ is the length of the chromosome and $r > 0$. Equally-spaced windows of width $w$ will contain $r$ ribonucleotides on average. We arbitrarily choose $r = 5$ in the actual analysis since it appeared to give reasonably-sized windows. Since the windows are fixed width, we have $\forall i, x_i = w(i-1)+1$.

The output from `smooth.Pspline` can be described as two vectors $\{v_i\}_{i=1}^N$, $\{v_i''\}_{i=1}^N$, which are the fitted value of the spline and its second derivative, respectively, at $\{x_i\}_{i=1}^N$. We then determine the roots of the second derivative as the set $R = \{x_i | i = 1, \ldots, N \text{ and } \operatorname{sign}(v_i'') \neq \operatorname{sign}(v_{i+1}'')\}$, where

$$\operatorname{sign}(a) = \begin{cases} -1 & \text{if } a < 0, \\ 0 & \text{if } a = 0, \\ 1 & \text{if } a > 0 \end{cases}$$

The boundaries of the windows are then the set $W = \{1\} \cup R \cup \{C\}$.

### 3.7 Similarity Metrics

We are also interested in whether there are features of the ribonucleotides counts that are conserved across different samples and cell types. To make this question more precise, let $\mathbf{y}^{(s)} = \left\{ y_i^{(s)} \right\}_{i=1}^N$ denote the binned ribonucleotide counts, where the superscript $s$ denotes the sample. We are interested in how similar are a pair of samples $\left\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}\right\}$ are or even a larger collection of $K$ samples $\left\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(K)}\right\}$. To quantify this, let $T : \mathbb{R}^{K \times N} \to \mathbb{R}$ be a function that maps the matrix $\left[\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(K)}\right]$ (here we assume each $\mathbf{y}^(i)$ is a column vector) to a real number. We define the different similarity (or dissimilarity) measures in the following subsections.

### 3.7.1 Spearman Correlation

This follows the definition of the function `cor` with `method = "spearman"` in the base R language [55]. We first define the rank function, $R : \mathbb{R}^N \to \mathbb{R}^N$, by

$$R(\mathbf{y})_i = \left( \sum_{j=1}^{N} \mathbf{1}\{y_j < y_i\} \right) + \frac{1}{2} + \frac{1}{2} \left( \sum_{j=1}^{N} \mathbf{1}\{y_j = y_i\} \right)$$

That is, $R(\mathbf{y})_i$ is rank of $y_i$ in the overall vector $\mathbf{y}$, where ties are given the midpoint of the tied ranks (e.g., if ranks 3, 4, and 5 have the same value, they are all given a rank of 4). Then the Spearman correlation is defined as

$$\begin{aligned} \mathbf{r}^{(1)} &= R(\mathbf{y}^{(1)}) = \left\{ r_i^{(1)} \right\}_{i=1}^{N}, \\ \mathbf{r}^{(2)} &= R(\mathbf{y}^{(2)}) = \left\{ r_i^{(2)} \right\}_{i=1}^{N}, \\ T_{\text{spear}}\left(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}\right) &= \frac{\sum_{i=1}^{N} \left( r_i^{(1)} - \overline{r}_{\cdot}^{(1)} \right) \left( r_i^{(2)} - \overline{r}_{\cdot}^{(2)} \right)}{\sqrt{\left( \sum_{i=1}^{N} \left( r_i^{(1)} - \overline{r}_{\cdot}^{(1)} \right)^2 \right) \left( \sum_{i=1}^{N} \left( r_i^{(2)} - \overline{r}_{\cdot}^{(2)} \right)^2 \right)}} \end{aligned}$$

where $\bar{a}_{\cdot} = \frac{1}{N} \sum_{i=1}^{N} a_i$ denotes the vector mean for any vector $\mathbf{a} = \{a_i\}_{i=1}^{N}$. The Spearman correlation is equivalent to the Pearson correlation between the vectors $\left\{ r_i^{(1)} \right\}_{i=1}^{N}$ and $\left\{ r_i^{(2)} \right\}_{i=1}^{N}$.

### 3.7.2 $L^p$ Distance

The $L^p$ distance, denoted $\|\cdot\|_p$, where $p > 0$, is a pairwise dissimilarity measure defined by

$$\begin{aligned} \|\mathbf{y}\|_p &= \left( \sum_{i=1}^{N} |y_i|^p \right)^{1/p}, \\ T(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}) &= \left\| \frac{\mathbf{y}^{(1)}}{\|\mathbf{y}^{(1)}\|_p} - \frac{\mathbf{y}^{(2)}}{\|\mathbf{y}^{(2)}\|_p} \right\|_p \end{aligned}$$

We normalize the vectors before subtracting to account for differences in the number of ribonucleotides detected in different libraries. For example, if $p = 1$, normalizing the vectors ensures that each sums to 1, making them probability mass functions.

### 3.7.3 Hotspot Similarity

The purpose of the *hotspot similarity* measure is to check whether the regions of relatively high ribonucleotides in each sample occur in similar locations on the genome. It can be considered as a means of smoothing the data by ignoring the low-count regions, which may have lower signal-to-noise ratios. For a given $p \in [0,1]$, the hotspot indicator of $\mathbf{y} = \{y_i\}_{i=1}^N$ is defined by

$$O_p(\mathbf{y})_i = \mathbf{1}\{y_i \geq \text{quantile}(\mathbf{y}, 1-p)\} \tag{3.1}$$

The empirical quantile above is computed using the base R function `quantile` with the argument `type = 7` (default). This function sets the empirical quantile of the $p = \frac{k-1}{N-1}$ to the $k^{\text{th}}$ order statistic of $\{y_1, \dots, y_N\}$ for $k = 1, \dots, N$, and linearly interpolates between these values for $p \in \left(\frac{k-1}{N-1}, \frac{k}{N-1}\right)$. Then the $p$-hotspot similarity between $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(1)}$ is defined as the Jaccard similarity of $O_p\left(\mathbf{y}^{(1)}\right)$ and $O_p\left(\mathbf{y}^{(2)}\right)$, which is

$$T\left(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}\right) = \frac{\sum_{i=1}^N \mathbf{1}\left\{O_p\left(\mathbf{y}^{(1)}\right)_i > 0 \text{ and } O_p\left(\mathbf{y}^{(2)}\right)_i > 0\right\}}{\sum_{i=1}^N \mathbf{1}\left\{O_p\left(\mathbf{y}^{(1)}\right)_i > 0 \text{ or } O_p\left(\mathbf{y}^{(2)}\right)_i > 0\right\}}$$

This value is 1 if the two indicator functions are positive on identical sets and 0 if their positive sets are disjoint.

### 3.7.4 Multiple-Sample Hotspot Indicators

In addition to the pairwise similarity measure, we also consider a measure of the overall similarity between an arbitrary number of samples $\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(K)}$ for $K \geq 2$. We define the individual hotspot sets $O_p(\mathbf{y})$ as in Equation 3.1 and define the joint hotspot indicator by

$$O\left(\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(K)}\right)_i = \sum_{k=1}^K O_p\left(\mathbf{y}^{(k)}\right)_i \tag{3.2}$$

Thus, $O\left(\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(K)}\right)_i$ indicates the number of samples that have a common hotspot at position $i$. To summarize the overall degree of overlap, we may consider statistics such as the maximum value of this function,

$$T\left(\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(K)}\right) = \max_{i=1}^N O\left((\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(K)}\right)_i,$$

or the number of indices where a value greater than a threshold, $h$, is reached,

$$T\left(\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(K)}\right)=\sum_{i=1}^{N}\mathbf{1}\{O\left(\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(K)}\right)_i\geq h\}$$

### 3.8 Permutation Tests

To quantify the significance of the various properties, such as similarity between the sample, we use Monte Carlo permutation tests. Permutation tests have been employed by tools such as the regioneR [24] R package and others reviewed in [21]. While we have referred to these sources for the methodology, our implementations are custom written.

In the traditional permutation test, we are given a vector of observations $\mathbf{y}=\{y_1,\ldots,y_N\}$, a test statistic $T:\mathbb{R}^N\to\mathbb{R}$, and the set $\mathcal{S}_N$ of all permutations of $\{1,\ldots,N\}$. Then, as shown by Phipson and Smyth (2010) [51], we may compute a one-sided p-value of the null hypothesis

$$H_0: y_i \text{ are independent and identically distributed (iid)}$$

by

$$p=\frac{\sum_{\sigma\in S_N}\mathbf{1}\{T(\mathbf{y}\circ\sigma)\geq T(\mathbf{y})\}}{|S_N|}$$

where $y\circ\sigma=\{y_{\sigma(1)},\ldots,y_{\sigma(N)}\}$ and $|S_N|=N!$ is the number of elements in $S_N$. In situations where it may be infeasible to try every permutation in $S_N$, they show that we may perform the test on a random subsample of $S_N$ with $K<|S_N|$ elements. The subsample, $S'=\{\sigma_1,\ldots,\sigma_K\}$, may be either chosen with or without replacement from $S_N$. Then the one-sided p-value can similarly be computed by

$$p=\frac{1+\sum_{k=1}^{K}\mathbf{1}\{T(\mathbf{y}\circ\sigma_k)\geq T(\mathbf{y})\}}{1+K}$$

In all cases, [51] shows that this expression for the p-value properly controls the type-I error, though the test may be overly conservative under certain conditions.

The work of Hemerik and Goeman (2018) [30] extends the permutation test to a more general setting that we use here. Instead of considering all permutations of the indices, we consider only a subgroup $G\subseteq S_N$. That is, $G$ is a subset of $S_N$ such that

- $\sigma_{\text{id}}\in G$ (where $\sigma_{\text{id}}$ is the identity permutation);

- for any $\sigma_1, \sigma_2 \in G$ we have $\sigma_1 \circ \sigma_2 \in G$; and
- for any $\sigma \in G$ we have $\sigma^{-1} \in G$.

Our motivation for using a restricted subgroup is to make our null hypothesis more restricted and thus more conservative. Hemerik and Goeman (2018) [30] prove analogous expressions for the permutation p-values when using permutations only in $G$. We assume $K > 0$ is fixed and $G' = \{\sigma_1, \ldots, \sigma_K\}$ is a subsample of $G$ chosen with replacement. Then the one-sided p-value can be calculated by

$$p = \frac{1 + \sum_{i=1}^{K} \mathbf{1}\{T(\mathbf{y} \circ \sigma_k) \geq T(\mathbf{y})\}}{1 + K} \tag{3.3}$$

The null hypothesis is the more general

$$H_0 : \forall \sigma \in G, T(\mathbf{y} \circ \sigma) \overset{d}{=} T(\mathbf{y}) \tag{3.4}$$

where $\overset{d}{=}$ means identically distributed. Though we have shown only one-sided p-value formulas, the analogous two-sided p-value to (3.3) can be calculated by

$$p = \frac{2\left[1 + \min\left(\sum_{k=1}^{K} \mathbf{1}\{T(\mathbf{y} \circ \sigma_k) \geq T(\mathbf{y})\}, \sum_{k=1}^{K} \mathbf{1}\{T(\mathbf{y} \circ \sigma_k) \leq T(\mathbf{y})\}\right)\right]}{1 + K} \tag{3.5}$$

We use these type of permutation tests to judge how strong the association between samples is. For the multiple sample statistics, $T : \mathbb{R}^{N \times S} \to \mathbb{R}$, explained in Section 3.7, we consider our observed values to be the matrix $\mathbf{y} = \left[\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(S)}\right]$ (each $\mathbf{y}^{(s)}$ is assumed to be a column vector). Then we may consider the group of permutations $G$ that are invariant on the indices of each column. That is, for all $\sigma \in G$ and all $s = 1, \ldots, S$, we have

$$\sigma\left(\{(i, s)\}_{i=1}^{N}\right) = \{(i, s)\}_{i=1}^{N}$$

The assumption is that for each sample $s$ the observations, $\mathbf{y}^{(s)} = \left\{y_i^{(s)}\right\}_{i=1}^{N}$, are identically distributed. We may also consider more conservative tests that leave finer partitions of the index set invariant. For example, we may find a partition $\{P_q\}_{q=1}^{Q}$ of the set $\{1, 2, \ldots, N\}$ and let $G$ be the permutations that are invariant on each $P_q \times \{s\} = \{(p, s) | p \in P_q\}$ for $q = 1, \ldots, Q$ and $s = 1, \ldots, S$. A simple way to form the partition is to select a positive integer $h$ and let $P_q$ be the consecutive indices $\{(q-1)w + 1, (q-1)h + 2, \ldots, \min(qh, N)\}$ for $q = 1, \ldots, \left\lceil \frac{N}{h} \right\rceil$. Note, in all the permutation results reported in Chapter 4, we use $h = 2$, which we sometimes refer to as the *adjacent swapping* scheme. This means, for all $s$, for $q = 1, \ldots, \left\lceil \frac{N}{2} \right\rceil$, we allow each pair $\left\{y^{(s)}_{2q-1}, y^{(s)}_{\min(2q,N)}\right\}$ to be randomly swapped. The null hypothesis is then that for all $s$ and all $q$, we have $y^{(s)}_{2q-1}$ and $y^{(s)}_{2q}$ are identically distributed.

Another way to restrict the permutations may be to only allow the nonzero elements of each sample to be permuted, since we may be unable to judge whether the zeros are due to artifacts in the reference genome such as gaps (i.e., the unknown regions on the genome) or variability in the data collection. However, the latter methods selects the permutations in a data-dependent manner, which may invalidate the permutation test. Thus, an alternative way may be to exclude the known gaps or repeat regions from consideration. In some analyses we remove indices $i$ such that the $i^{\text{th}}$ binning window (see Section 3.2), $\{w(i-1)+1, \ldots, \min(wi, C)\}$, lies fully in a gap.

After selecting the subgroup $G$ we perform a one-sided or two-sided permutation test to determine whether the observed similarity measure between the samples is consistent with what would be observed under the null-hypothesis (3.4). If the test is rejected, we hope to establish that there is a position-dependent pattern in the observed ribonucleotides that holds across multiple independent cell types. However, it is important to note that the validity of this conclusion relies on the null hypothesis (or, equivalently, the chosen permutations) being a biologically realistic assumption. Here, we do not attempt to justify the permutation approach on biological grounds but simply report the results. However, for future study it may be important to use a permutation strategy that is more grounded in theory (see Chapter 5 for a discussion).

### 3.9 DNA-Seq Coverage

A potential bias that may affect our analyses is unequal read coverage of different parts of the genome. When analyzing a sample, the *coverage* of a position on the genome refers to how many reads overlap that position. Ross et al. (2013) [60] have observed that some sequencing technologies may exhibit bias for certain regions of the genome. For example, this may occur due to a systematic bias for regions with an optimal GC content. These biases in the DNA sequence coverage may translate into biases in the ribonucleotide counts, since ribonucleotides are detected by sequencing DNA reads (after performing several additional steps to ensure that the read represents a ribonucleotide). Thus, we would like to ensure that the regions detected as ribonucleotide hotspots are not simply due to biased coverage in that part of the genome, but truly reflect a higher quantity of ribonucleotides in that area. To determine whether there is biased coverage in the DNA reads, we perform DNA sequencing experiments without the additional steps for detecting ribonucleotides. When we map these reads to the genome, we obtain a vector of background coverage values, $\mathbf{z} = \{z_i\}_{i=1}^{N}$, where each $z_i$ is the average number of reads overlapping window $i$ (or the average coverage). Then we visually compare this with the ribonucleotide vector $\mathbf{y} = \{y_i\}_{i=1}^{N}$ to determine whether appears to be any systematic association between the two vectors. This may be a linear relationship, where we expect $\mathbf{y} \approx \alpha\mathbf{z}$ for some scaling constant $\alpha$, or it may be nonlinear, where we would expect there would be some other strictly monotonic function $f : \mathbb{R} \to \mathbb{R}$ such that $\forall i, y_i \approx f(z_i)$. We quantify the strength of the relationship in the same way we handled the pairwise comparisons of ribonucleotide vectors, by using a correlation coefficient and testing the significance using a permutation test.

An alternative method to visualize the correlation between $\mathbf{y}$ and $\mathbf{z}$ is to plot the vector of fold changes in the empirical densities. We first define the empirical densities by $\mathbf{y}^* = \left\{y_i/\sum_{j=1}^N y_j\right\}_{i=1}^N$ and $\mathbf{z}^* = \left\{z_i/\sum_{j=1}^N z_j\right\}_{i=1}^N$. Then we define the fold change of the densities by $\{y_i^*/z_i^*\}_{i=1}^N$. Plotting these values would then give us an indication of how similar the empirical densities $\mathbf{y}^*$ and $\mathbf{z}^*$ are at each position.

As some of raw values $z_i$ are outliers, we perform additional outlier removal. First, define $\hat{\mu}$ to be the sample median and $\hat{\sigma}$ to be the median absolute deviation:

$$\hat{\mu} = \text{median}_{i=1}^N z_i,$$
$$\hat{\sigma} = \frac{1}{\Phi^{-1}(3/4)} \text{median}_{i=1}^n |z_i - \hat{\mu}|$$

The definition for $\hat{\sigma}$ above is the default used by the `mad` function in R [55]. The scaling constant $\frac{1}{\Phi^{-1}(3/4)}$ is used to make it a consistent estimator for the standard deviation if $\mathbf{z}$ iid normally distributed. Outliers are defined as windows $i$ such that

$$\frac{z_i - \hat{\mu}}{\hat{\sigma}} > 4$$

where the threshold 4 was chosen heuristically by trial and error as it removed the outliers when binning with 100kb windows.

### 3.10 Kmer Correlation

We are also interested in examining the association of the ribonucleotides with the nucleotide composition of the genome. From a computational standpoint, nucleotides are simply the string of letters over A, C, G, and T that make up the genome. These letter do no occur with uniform frequency across the reference genome GRCh38. For example, the GC content (frequency of the letters G and C) is around 40.9% [52] in the human genome (making the AT content around 59.1%). Further, certain nucleotide motifs may be associated with biological functions, such as to indicate regions that code for proteins.

To study the association of ribonucleotides with nucleotide motifs, we focus on motifs known as *kmers*. Given a positive integer $k$, the length $k$ kmers are the $4^k$ strings of length $k$ over the alphabet {A, C, G, T}.

For example, the length 1 kmers are {A, C, G, T} and the length 2 kmers are {AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT, TA, TC, TG, TT}. Any nucleotide sequence $S = s_1 s_2 \cdots s_n$ of length $n$ can be decomposed into $n - k + 1$ overlapping kmers. For example, for $k = 3$ we get $s_1 s_2 s_3$, $s_2 s_3 s_4$, ..., $s_{n-2} s_{n-1} s_n$. A simple way to summarize the kmers in $S$ is to map $S$ to a vector of counts of each kmer of length $k$. For any kmer $u$, we define $T^{(u)}(S) = \sum_{i=1}^{n-k+1} \mathbf{1}\{s_i s_{i+1} \cdots s_n = u\}$, which counts the number of times $u$ appears in $S$.

Since we study the binned counts of ribonucleotides $\mathbf{y} = \{y_i\}_{i=1}^N$, where $w$ is the binning window size, $N = \lceil C/w \rceil$, and $C$ is the chromosome length (see Section 3.2), we form an analogous sequence for the kmers. Let the nucleotide sequence of the chromosome be $S = s_1 s_2 \cdots s_C$. Then for a kmer $u$, we define the count vector of $u$ by $\mathbf{z}^{(u)} = \left\{ z_i^{(u)} \right\}_{i=1}^N$ by

$$z_i^{(u)} = T^{(u)}(s_{w(i-1)+1} s_{w(i-1)+2} \cdots s_{wi})$$

Just as we examine the correlation between the ribonucleotides of different samples, we can also examine the correlation of the ribonucleotides vectors with the kmer count vectors. We may also apply similar permutation tests as described in Section 3.8, although in this case we only permute the sample $\mathbf{y}$ while keeping $\mathbf{z}^{(u)}$ fixed.

### 3.11 Annotation Correlation

Similarily to the kmer correlation analysis of Section 3.10, we also study the correlation of ribonucleotides with existing annotations of the genome. For our purpose, an *annotation* can be considered a sequence of genome intervals with labels. That is, a sequence of triples $A = \{(s_i, e_i, l_i)\}_{i=1}^M$, where $s_i$ is a start position, $e_i$ is an end position, and $l_i$ is a label. The label will usually indicate something of biological importance such as “gene”. These annotations have been previously discovered through a combination of computational and experimental means and are now available on publicly available databases. The annotations we use are available in the R packages *TxDb.Hsapiens.UCSC.hg38.knownGene* [68] and *org.Hs.eg.db* [11], and are processed using the R package *annotatr* [12]. The annotations used from these databases are the *CpG islands*, *exons*, and *promoters*, though we do not describe their biological meaning here.

We summarize the annotations in a similar way to the ribonucleotide and kmer data, by counting the number of distinct annotations in windows of a fixed width (see Section 3.2). We obtain a vector of counts $\mathbf{z} = \{z_i\}_{i=1}^N$, where each $z_i$ is the number of the elements of $A$ that overlap with window $i$. An annotated interval $(s_i, e_i, l_i)$ is considered to overlap with a window if any of the coordinates in the interval $\{s_i, \ldots, e_i\}$ overlap with the window. This allows an annotated interval to overlap with multiple windows. We analyze

**a** in an analogous way to the kmer vectors: by visualization, pairwise correlations with the ribonucleotide vectors, and permutation tests.

# CHAPTER 4:
# RESULTS

## 4.1 Data Summary

In this exposition we focus on the six ribose-seq libraries described in Section 1.3. Although our project has involved additional libraries, we focus on these for brevity in the computations and exposition. We also focus on the data for chromosome 1 in the GRCh38 reference genome. Again, this is to reduce the computational expense although we plan to analyze the other chromosomes in the future. However, chromosome 1 is the largest chromosome in the human genome with approximately 249 million base pairs.

Table 2 shows the summary statistics of each of the samples. We see that our datasets are generally quite sparse, with between $10^{-4}$ and $10^{-2}$ reads per base. This motivates the use of binning with large windows as described in Section 3.2. Table 4 shows the number of reads per window for different binning window widths, though we mostly use the widths 100kb and 1mb in this study. Throughout this section we use the abbreviations: “b” (bases), “kb” (kilobases or 1,000 bases), and “mb” (megabases or 1,000,000 bases).

One major distinction in the samples is the number of reads in the knockout (KO) samples versus the other samples. Due to their nonfunctional RNASEH2A gene, the ability of these cells to remove ribonucleotides from the genome is impaired. Thus, they are expected to have more ribonucleotides than the other samples.

As explained in Section 3.2, we have removed the repeat regions. For comparison, we also show the data summaries with the repeat regions retained in Table 3. A substantial portion of the detected ribonucleotides are removed when ignoring repeat regions because the repeat regions account for roughly half of the genome (see Section 3.2).

Figure 1 shows the ribonucleotide distribution across the chromosome. We see that the distributions look similar across each of the samples, indicating that ribonucleotide occurrence may be similar across different human cell types.

### 4.2 Window Width Selection

As described in Section 3.3, we use the AIC and CV criterion to determine reasonable window widths. We use this method only as a suggestion, since our choice of window width is also based on other factors such as computational cost (smaller window widths result in more data points), and ease of presentation and interpretation. Figure 2 shows the plot of AIC and CV log-likelihood versus window width. The optimal window width is indicated by the dotted line. We see that the AIC selects window widths of 1kb to 100kb, with at least half of the samples selecting 100kb as optimal for both the + and - strands. The CV results do not appear to be quite as informative since the curves are monotonically decreasing in most samples and usually the smallest window width of 1kb is selected. This may indicate that the CV criteria would choose much smaller window widths or that it is unsuitable for these datasets. Though we originally chose 100kb windows arbitrarily to perform statistical analyses, the AIC results seem to suggest that 100kb is indeed an appropriate choice. However, when presenting figures with chromosome position on the $x$-axis, we usually use 1mb for greater clarity, since with 100kb windows there are too many data points. To get a rough idea of the scale of these window widths, the average size of a gene on chromosome 1 is around 36kb (obtained by author using the *TxDb.Hsapiens.UCSC.hg38.knownGene* [68] R package) and chromosome 1 is around 249mb.

### 4.3 Sample Similarity

Figure 3 shows the pairwise Spearman correlation between the samples. We choose to show the Spearman correlation since it may better indicate nonlinear monotonic relationships between vectors compared to the Pearson correlation. With the exception of sample FS326, all the coefficients are greater than 0.75 indicating a strong positive correlation. Sample FS326 had the smallest number of ribonucleotides detected and thus may also have the smallest signal-to-noise ratio. To check the significance of the correlations, we use the adjacent swapping permutation scheme described in Section 3.8. One thousand permutations were used for each pair. All correlations tested significant at at 1% level. In fact, in all pairs, none of the permuted samples had correlations as large as the observed value.

Next, we look at the similarity of all six sample simultaneously using the hotspot indicators. In Figure 4 we see the 1%-hotspots (windows that are in the top 1% of ribonucleotides counts within a sample) along chromosome 1. There are several regions where several samples have hotspots in the same position. To quantify this, we show the number of samples ($y$-axis) that have a hotspot at each position ($x$-axis) in Figure 5. There are several positions where more than half the samples ($\geq 3$) have a hotspot in the same position.

We again perform a permutation test to check the significance of the results in Figure 6. The test statistic is the number of positions (in 100kb windows) such that at least half the samples ($\geq 3$) had a hotspot the same position. The figure shows both the permutation distribution and the observed value. The adjacent swapping permutation scheme is used (see Section 3.8). The p-value for the + strand is 0.09 and for - strand is 0.03. This is moderate evidence that such shared hotspots could not have occurred by chance, though further evidence may be needed.

### 4.4 Distribution Fitting

Figure 7 shows the zero-truncated Poisson distribution and Figure 8 shows the zero-truncated negative-binomial fit to the data. As discussed in Section 3.4, we show the zero-truncated versions of the distributions because of the challenge in determining the true frequency of zeros in the empirical distribution. We clearly see that the negative-binomial provides a significantly better fit to the data, which is expected because the negative-binomial is more flexible than the Poisson. Although the distributions appear to fit well for some of the samples, formal goodness-of-fit tests (not shown here) indicate that the fit is poor. This may be because the sample sizes are relatively large ($> 2000$ when binned into 100kb windows) and small deviations may be statistically significant. However, a formal statistical analysis of the empirical distributions within each sample may not be relevant as it is not clear whether the ribonucleotide counts in different windows can be considered independent observations, since they come from the same physical specimen.

### 4.5 Kmer Correlation

Figure 9 shows the ribonucleotide frequency of sample FS185 versus the frequency of kmers A and C along chromosome 1 (+ strand). We show only a single sample since, as observed in Section 4.3, most of the samples are highly correlated. We only show A and C because A and T are highly correlated, and C and G are highly correlated. Instead of showing the raw count of ribonucleotides and kmers, we use the rank transformation on the $y$-axis, as described in Section 3.7.1. This means the window with the smallest count is assigned a value of 1, next smallest 2, and so on (ties are assigned the median value of the tied ranks). This allows us to more clearly see the similarity between the curves, such as when a nonlinear monotonic relationship exists between them. Moreover, the Spearman correlation coefficient is a function of these rank-transformed curves (see Section 3.7.1). The most striking feature is the strong positive correlation between the C curve and the ribonucleotide curve, and the strong negative correlation between the A curve and the ribonucleotide curve. This is confirmed by the pairwise Spearman correlations between the kmer vectors and ribonucleotide vectors in Figure 10. We see that the ribonucleotides are positively correlated with C and

G, and negatively correlated with A and T. This can be summarized as saying that the ribonucleotides are positively correlated with the *GC content*, the sum of the frequencies of C and G.

### 4.6 Annotation Correlation

We first summarize the three types of annotated regions used in this analysis (CpG islands, exons, and promoters) in Table 5. Figure 11 shows the line graphs of the ribonucleotides alongside the three annotation count vectors. Like the kmer correlation analysis in Section 4.5, there appears to be a correlation between the ribonucleotides and each of the annotated features. This is confirmed by Figure 12, which shows the Spearman correlation between the ribonucleotide and annotation vectors. We again performed a permutation test using the adjacent swapping permutation scheme (see Section 3.8) to obtain significant p-values at a 1% significance level.

### 4.7 DNA-Seq Coverage

We obtained DNA-seq libraries for the four types of enzymes used in the ribose-seq library preparation [38]: F (Fragmentase), RE1 (restriction enzyme set 1), RE2 (restriction enzyme set 2), and RE3 (restriction enzyme set 3). For some figures and tables, we only show the Fragmentase DNA-seq data, since we only use the six ribonucleotide libraries prepared with Fragmentase (see Section 1.3). Figure 13 shows the coverage of all four of the DNA-seq samples across chromosome 1. Despite using different enzymes to fragment the DNA, they are all remarkably similar, indicating that the DNA-seq coverage estimates have low sampling variation (at least when using appropriately-sized windows). Figure 13 shows the DNA-seq coverage versus the ribonucleotide samples. The figures are binned using 1mb windows. Visually, we see that there does not appear to be a strong correlation between the ribonucleotides and the coverage. However, when we bin with 100kb windows and compute the Spearman correlation, we see that the correlations are in the range 0.35-0.55 (Figure 14). Additionally, applying the adjacent-swapping permutation test (see Section 3.8) shows that the correlations are significant. Assuming this permutation strategy is reasonable, this may suggest that the ribonucleotide counts are biased by the nonuniform coverage in the sequencing process.

### 4.8 Smoothing Spline Hotspots

Since the spline method for determining window boundaries results in unequally-sized windows, we summarize the window widths for each sample in Figure 15 and Table 6. Notably, for many samples it appears that the median spline window widths have comparable magnitudes to the window widths selected by AIC (Section 4.2). Since the window widths are unequally sized, we compute the density by

$$density = \frac{\text{total ribonucleotides in window}}{\text{width of window}}$$

The histogram of window densities are shown in Figure 16. We see some of the distributions have long right tails, indicating possible hotspots with relatively high ribonucleotide densities. We define hotspots quantitatively as the windows whose density falls in the top 1% for that sample. Again, since the windows cover different widths, we summarize the number of hotspot windows in each window and the fraction of the chromosome that they cover in Table 7. Not surprisingly, the percentage of the chromosome covered by the hotspots is generally between 0.75% and 1.5% for all the libraries (except FS326, the outlier dataset). However, we see that the corresponding percentage of ribonucleotides in the sample is around 2 to 7 times the chromosome percentage, indicating that the hotpsot windows contain more ribonucleotides than would expected based on their size. Finally, in Figure 17 we show the hotspot windows along the chromosome. There appear to be certain regions, especially towards the 0mb end of the chromosome, that are shared hotspots for multiple samples.

**Table 2.** Summary of ribonucleotide counts. Chromosome 1. + strand (A). - strand (B). Chromosome 1 size ≈ 249mb. RPB = "ribos per base" (ribonucleotides divided by chromosome length).

A

| Sample | Cell | Enzyme | Ribos | RPB |
|---|---|---|---|---|
| FS185 | CD4T | F | 26,950 | 1.08e-04 |
| FS197 | hESC-H9 | F | 19,589 | 7.87e-05 |
| FS198 | hESC-H9 | F | 28,097 | 1.13e-04 |
| FS326 | HEK293T-WT | F | 6,140 | 2.47e-05 |
| FS327 | HEK293T-RNASEH2A-KO-T3-17 | F | 145,266 | 5.83e-04 |
| FS329 | HEK293T-RNASEH2A-KO-T3-8 | F | 512,358 | 2.06e-03 |

B

| Sample | Cell | Enzyme | Ribos | RPB |
|---|---|---|---|---|
| FS185 | CD4T | F | 26,496 | 1.06e-04 |
| FS197 | hESC-H9 | F | 19,620 | 7.88e-05 |
| FS198 | hESC-H9 | F | 28,397 | 1.14e-04 |
| FS326 | HEK293T-WT | F | 6,228 | 2.50e-05 |
| FS327 | HEK293T-RNASEH2A-KO-T3-17 | F | 145,649 | 5.85e-04 |
| FS329 | HEK293T-RNASEH2A-KO-T3-8 | F | 510,854 | 2.05e-03 |

**Table 3.** Summary of ribonucleotide counts (with repeat regions retained). Chromosome 1. + strand (A). - strand (B). Chromosome 1 size ≈ 249 × $10^6$bp). RPB = "ribos per base" (ribonucleotides divided by chromosome length).

A

| Sample | Cell | Enzyme | Ribos | RPB |
|---|---|---|---|---|
| FS185 | CD4T | F | 54,281 | 2.18e-04 |
| FS197 | hESC-H9 | F | 40,686 | 1.63e-04 |
| FS198 | hESC-H9 | F | 62,011 | 2.49e-04 |
| FS326 | HEK293T-WT | F | 13,219 | 5.31e-05 |
| FS327 | HEK293T-RNASEH2A-KO-T3-17 | F | 282,862 | 1.14e-03 |
| FS329 | HEK293T-RNASEH2A-KO-T3-8 | F | 925,309 | 3.72e-03 |

B

| Sample | Cell | Enzyme | Ribos | RPB |
|---|---|---|---|---|
| FS185 | CD4T | F | 53,919 | 2.17e-04 |
| FS197 | hESC-H9 | F | 40,798 | 1.64e-04 |
| FS198 | hESC-H9 | F | 62,163 | 2.50e-04 |
| FS326 | HEK293T-WT | F | 13,406 | 5.38e-05 |
| FS327 | HEK293T-RNASEH2A-KO-T3-17 | F | 282,745 | 1.14e-03 |
| FS329 | HEK293T-RNASEH2A-KO-T3-8 | F | 925,503 | 3.72e-03 |

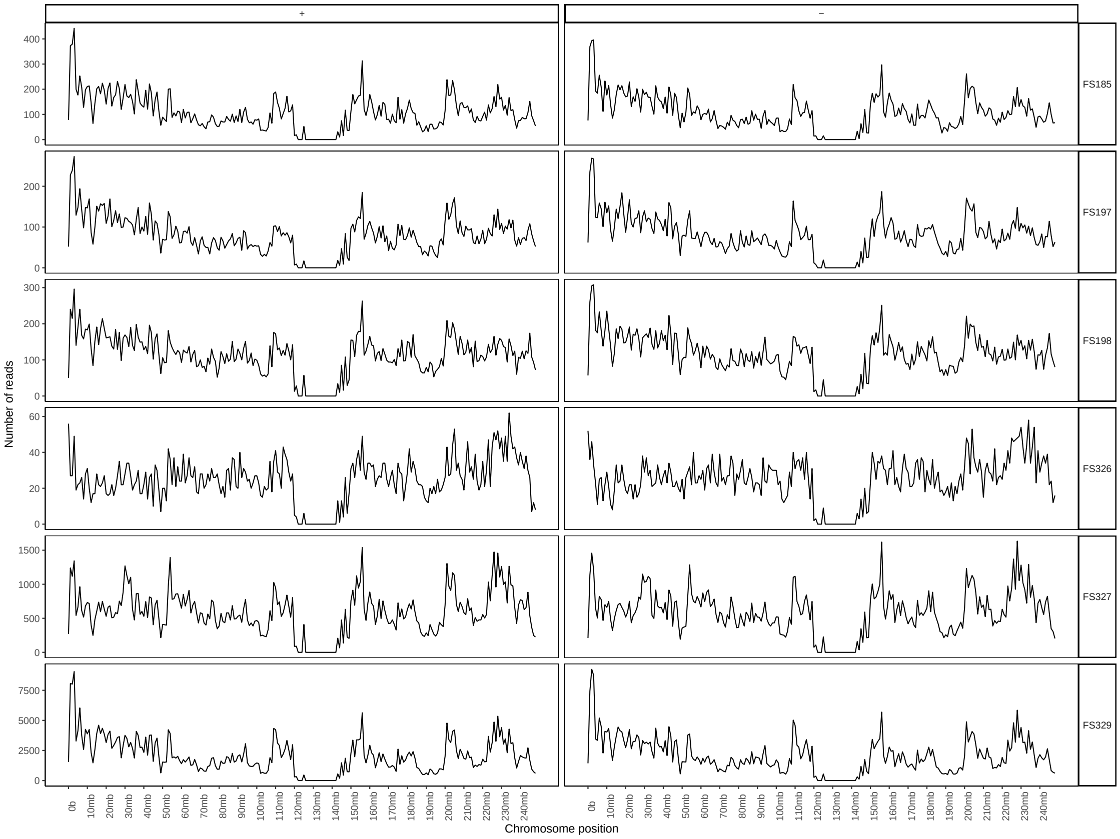

**Figure 1.** Ribonucleotides vs. chromosome position. Chromosome 1. + strand (left). - strand (right). Window width = 1mb.

**Table 4.** Mean ribonucleotides per window for different windows widths. Chromosome 1. + strand (A). - strand (B).

A

| Sample | 1kb | 10kb | 100kb | 1mb | 10mb |
|---|---|---|---|---|---|
| FS185 | 0.12 | 1.17 | 11.66 | 116.16 | 1122.92 |
| FS197 | 0.08 | 0.85 | 8.48 | 84.44 | 816.21 |
| FS198 | 0.12 | 1.22 | 12.16 | 121.11 | 1170.71 |
| FS326 | 0.03 | 0.27 | 2.66 | 26.47 | 255.83 |
| FS327 | 0.63 | 6.30 | 62.86 | 626.15 | 6052.75 |
| FS329 | 2.22 | 22.22 | 221.70 | 2208.44 | 21348.25 |

B

| Sample | 1kb | 10kb | 100kb | 1mb | 10mb |
|---|---|---|---|---|---|
| FS185 | 0.11 | 1.15 | 11.47 | 114.21 | 1104.00 |
| FS197 | 0.09 | 0.85 | 8.49 | 84.57 | 817.50 |
| FS198 | 0.12 | 1.23 | 12.29 | 122.40 | 1183.21 |
| FS326 | 0.03 | 0.27 | 2.69 | 26.84 | 259.50 |
| FS327 | 0.63 | 6.32 | 63.02 | 627.80 | 6068.71 |
| FS329 | 2.22 | 22.15 | 221.05 | 2201.96 | 21285.58 |

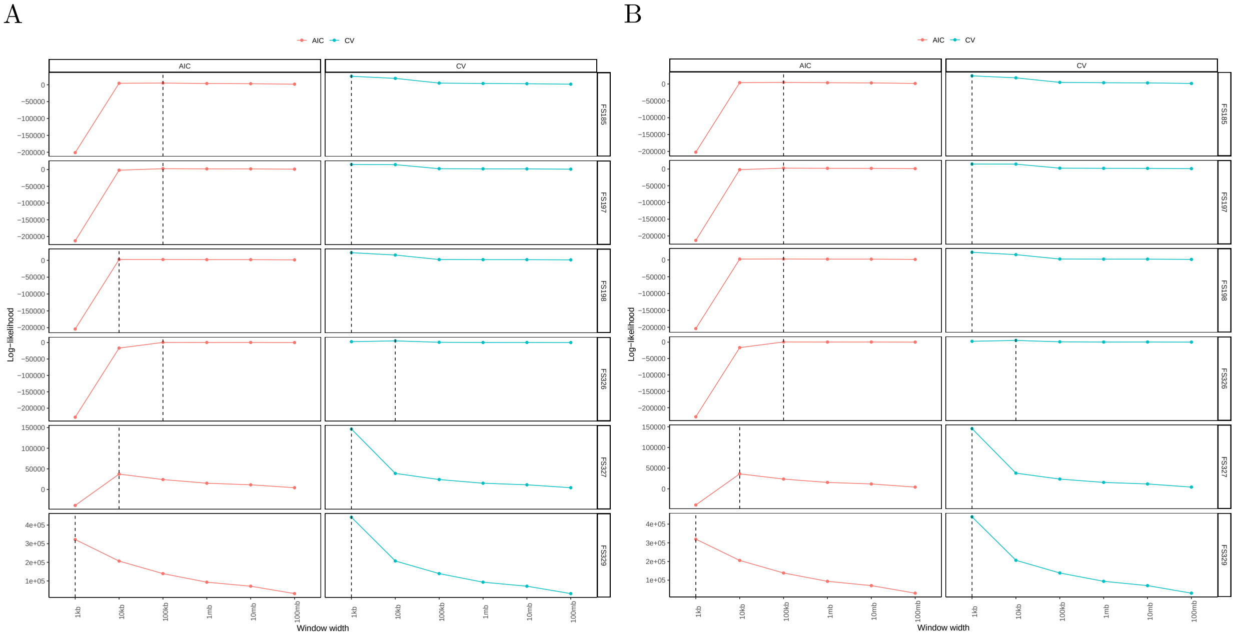

**Figure 2.** AIC and CV log-likelihood vs. window width. Optimal width indicated by dotted line. Chromosome 1. + strand (A). - strand (B).

**Table 5.** Summary of annotated features. "N" is the total number of features. "Width median" is the median width in bases. Strand "*" indicates a feature that exists on both strand of the chromosome. All promoters in the dataset had width 1000b. Chromosome 1.

| Strand | Feature | N | Width median |
|---|---|---:|---:|
| * | CpG islands | 2535 | 583 |
| + | Exons | 79786 | 133 |
| + | Introns | 68169 | 1524 |
| + | Promoters | 11617 | 1000 |
| - | Exons | 72089 | 130 |
| - | Introns | 61313 | 1345 |
| - | Promoters | 10776 | 1000 |

**Table 6.** Median window widths obtained using spline method. Chromosome 1. + and - strand indicated in column name.

| Sample | Median width (+) | AIC (+) | Median width (-) | AIC (-) |
|---|---|---|---|---|
| FS185 | 231kb | 100kb | 235kb | 100kb |
| FS197 | 318kb | 100kb | 254kb | 100kb |
| FS198 | 266kb | 10kb | 263kb | 100kb |
| FS326 | 7.5mb | 100kb | 4.8mb | 100kb |
| FS327 | 51kb | 10kb | 43kb | 10kb |
| FS329 | 9.7kb | 1kb | 9.7kb | 1kb |

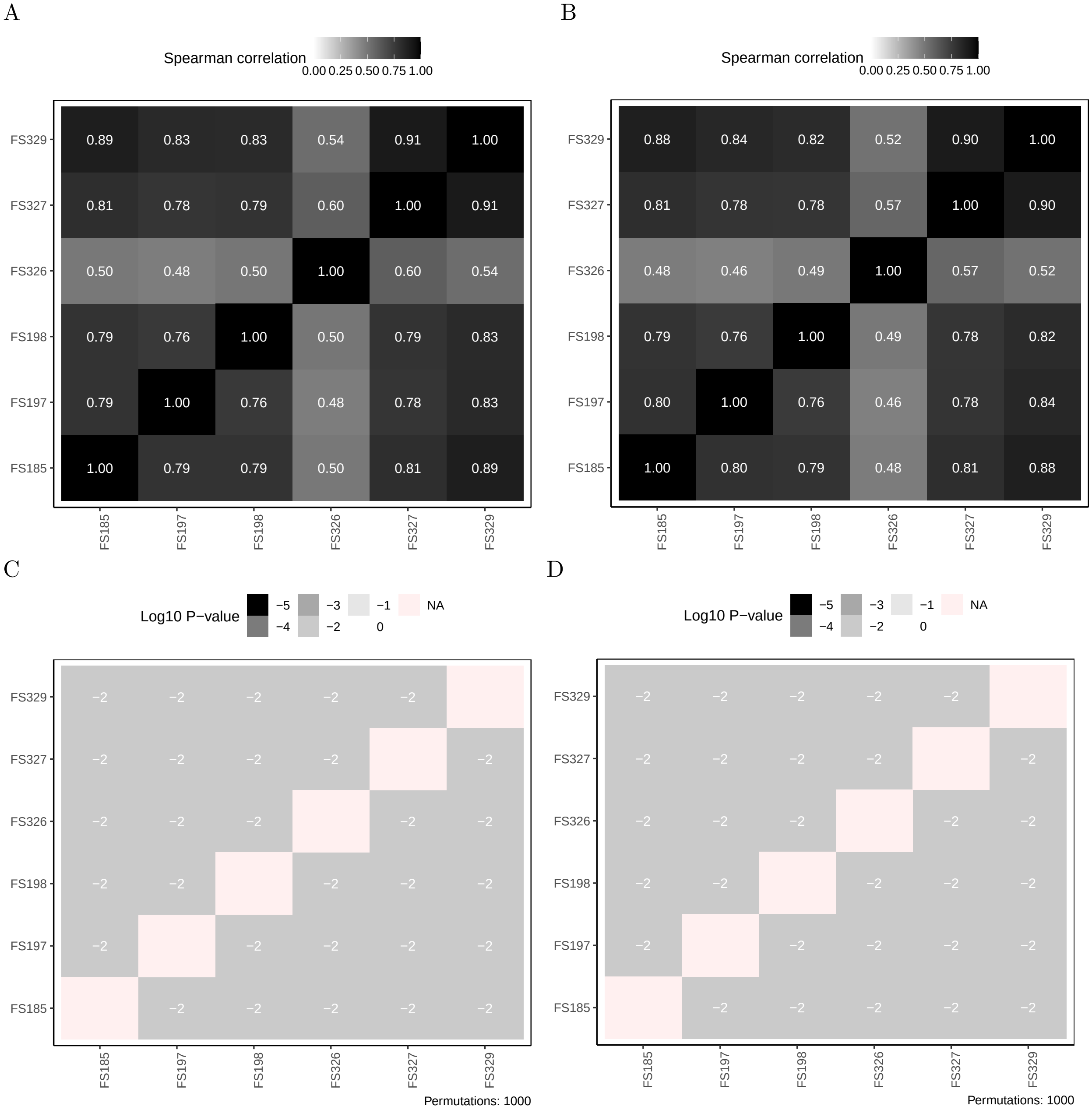

**Figure 3.** Spearman correlation between pairs of samples (A, B). Two-sided permutation p-values with 1000 permutations (C, D). Chromosome 1. + strand (A, C). - strand (B, D).

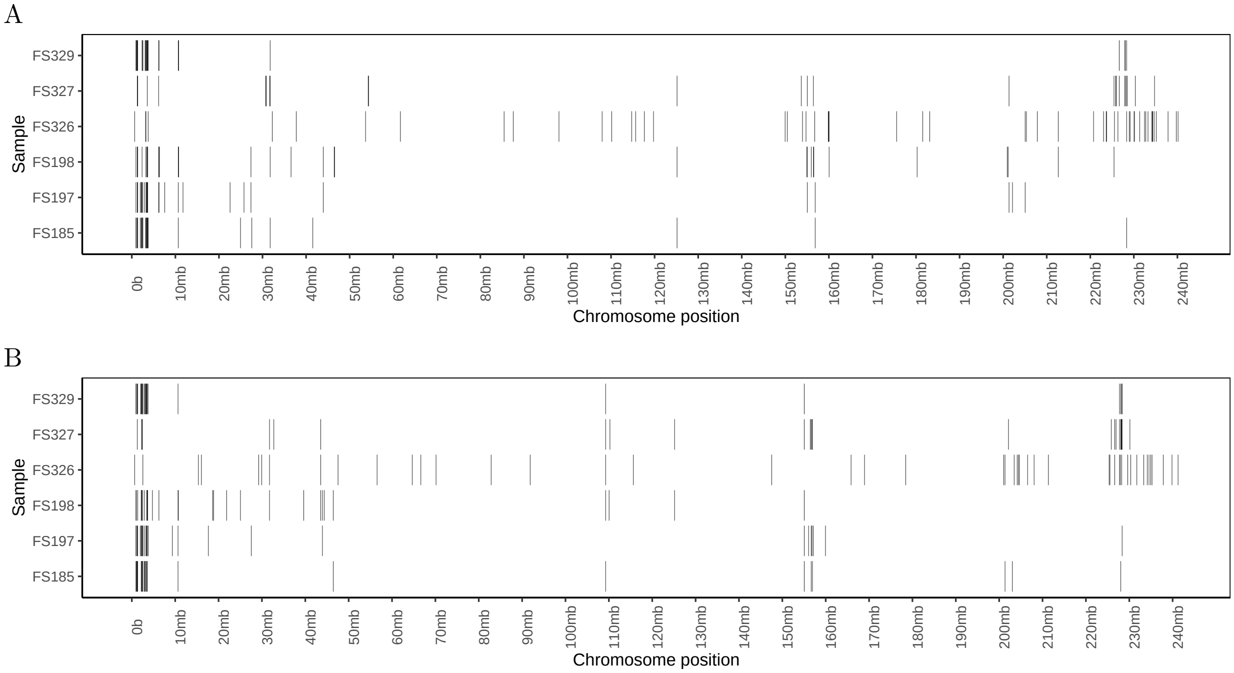

**Figure 4.** Each segment represents a 100kb hotspot window. A hotspot is a window whose ribonucleotide count is in the top 1% of windows in the sample. Chromosome 1. + strand (A). - strand (B).

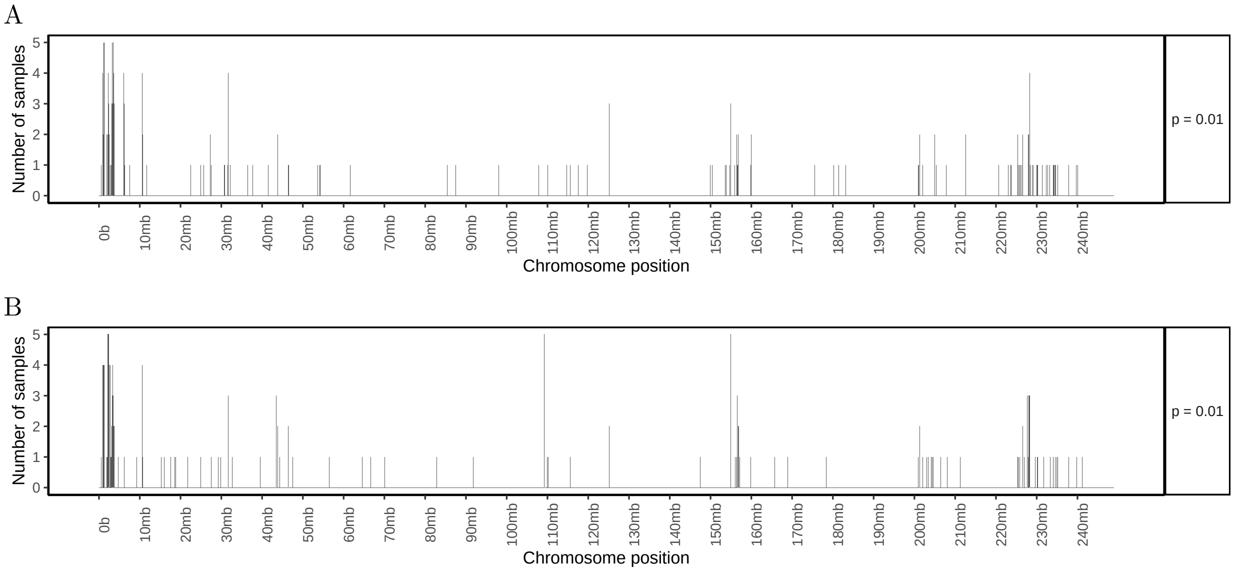

**Figure 5.** Hotspot indicators. Each vertical bar is the number of samples with a hotspot in that position. A hotspot is a window whose ribonucleotide count is in the top 1% of windows in the sample (see Figure 4). There are several positions where at least three out of the six samples share a hotspot. Chromosome 1. + strand (A). - strand (B). Window width = 100kb.

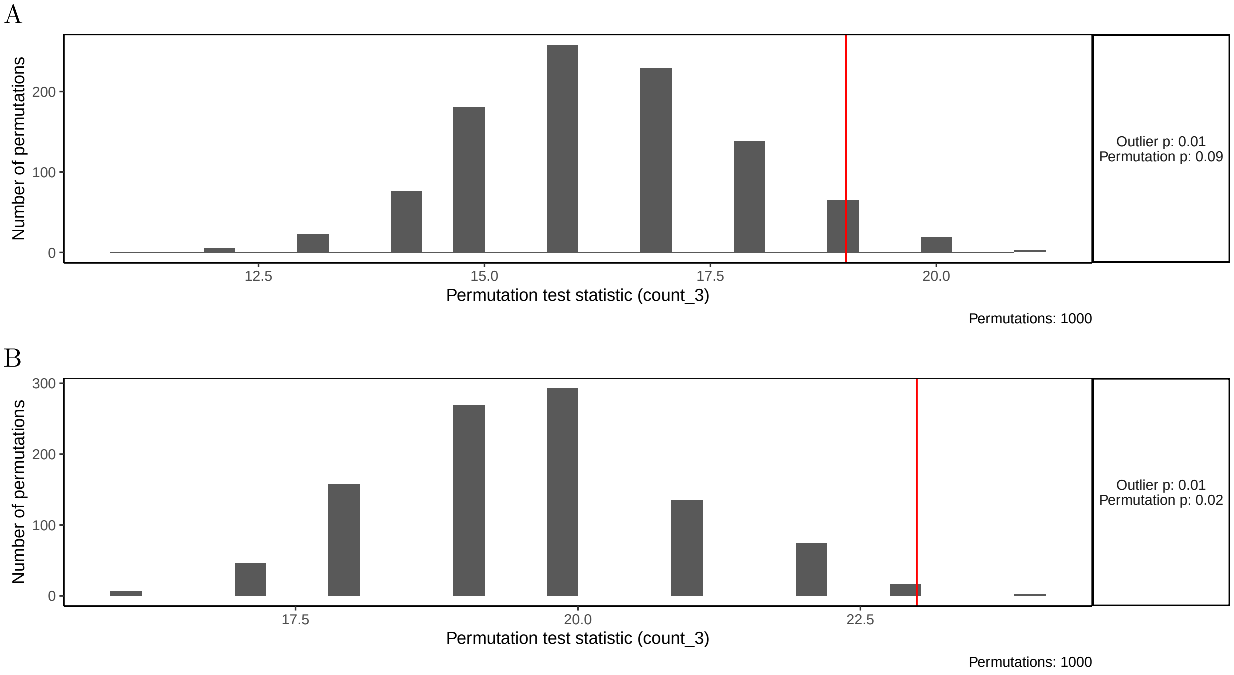

**Figure 6.** Permutation analysis to check the significance of the shared hotspots in Figure 5. We permute the data 1000 times using the adjacent swapping scheme (see Section 3.8) and evaluate the number of positions with $\geq 3$ samples sharing a hotspot. The permuted values are black bars and the observed value is a red line. Chromosome 1. + strand (A). - strand (B). Window width = 100kb.

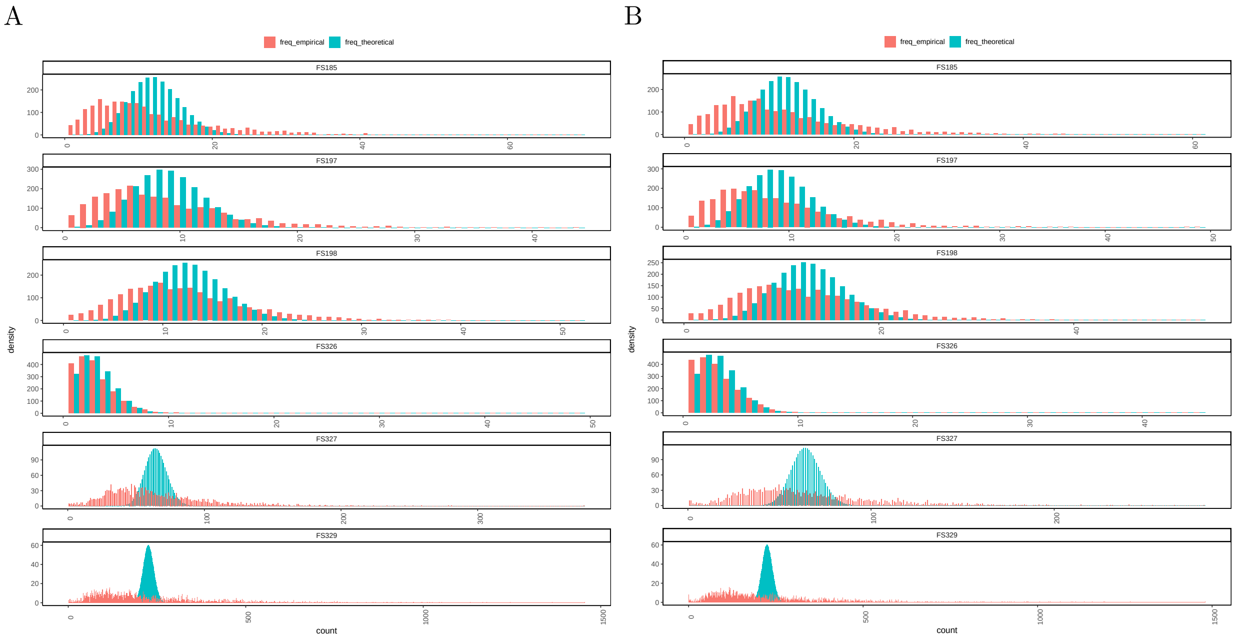

**Figure 7.** Histograms of zero-truncated Poisson distributions (green) fit to the data (orange). Window width = 100kb. Chromosome 1. + strand (A). - strand (B).

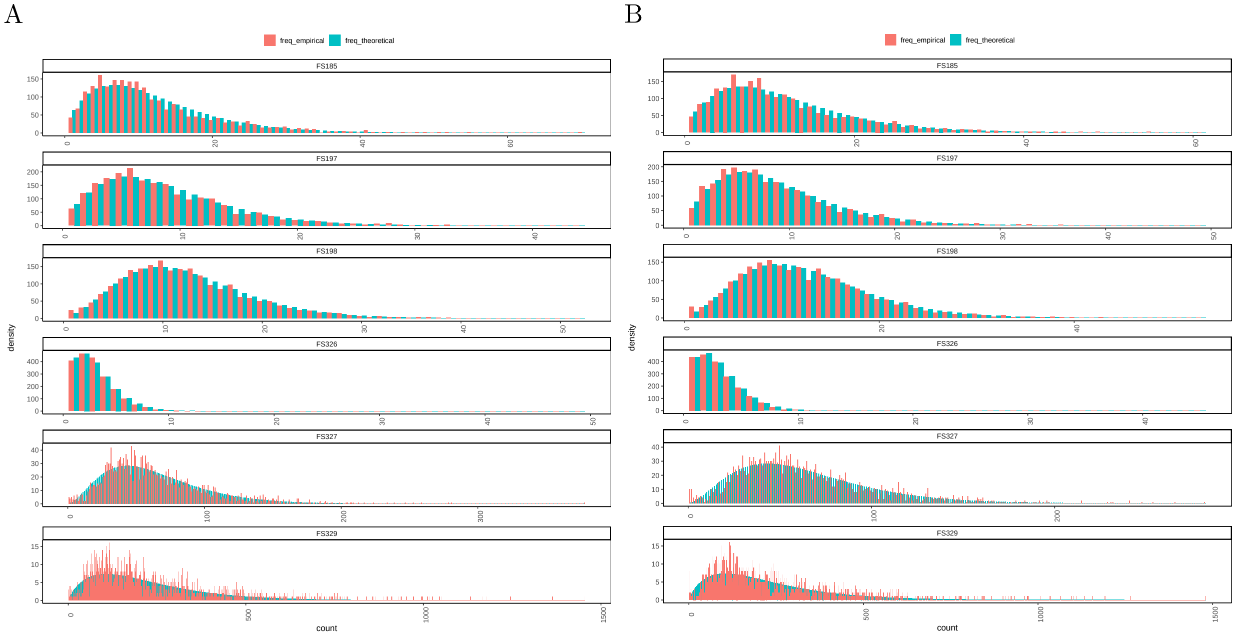

**Figure 8.** Histograms of zero-truncated negative-binomial distributions (green) fit to the data (orange). Window width = 100kb. Chromosome 1. + strand (A). - strand (B).

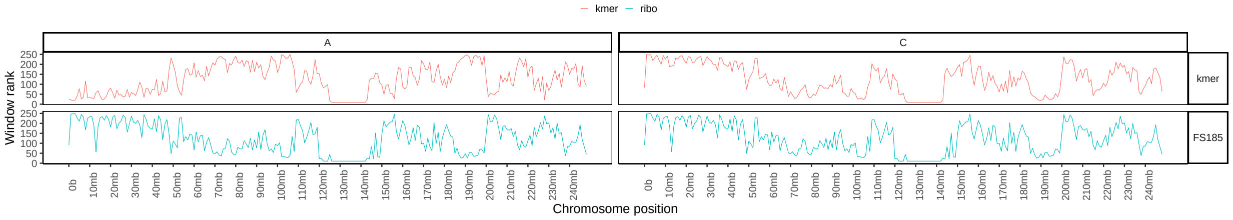

**Figure 9.** Ranks of kmer counts and ribonucleotide counts. The orange curves show the ranks of A and C counts in each window. The green curves show the ranks of ribonucleotide counts in each window. Chromosome 1 (+ strand). Sample FS185 only. Window width = 1mb.

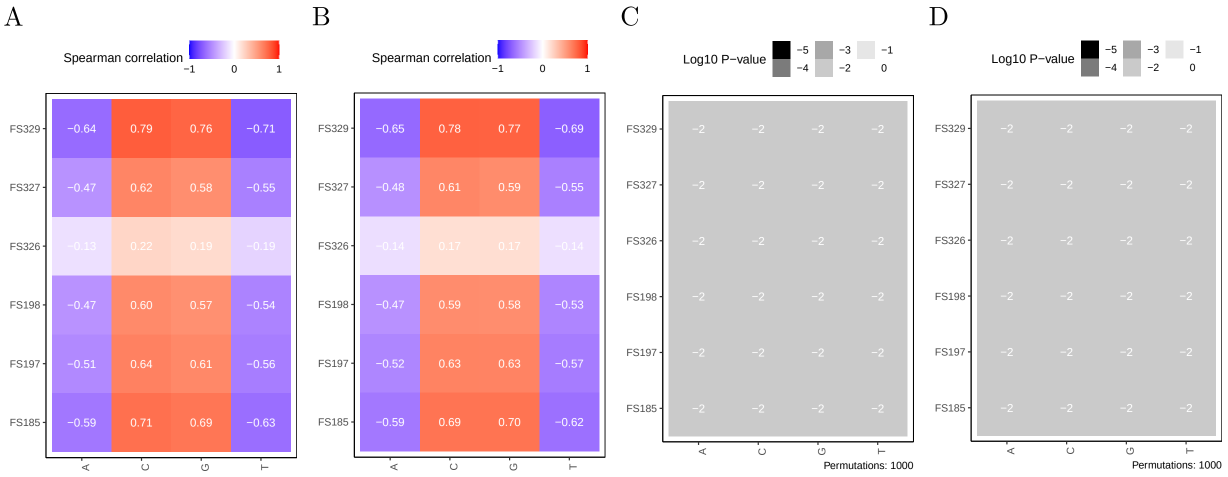

**Figure 10.** Spearman correlation between the nucleotide frequencies and the ribonucleotide frequencies (A, B). Two-sided permutation p-values of correlations using adjacent swapping scheme with 1000 permutations (see Section 3.8) (C, D). Chromosome 1. + strand (A, C). - strand (B, D). Window width = 100kb.

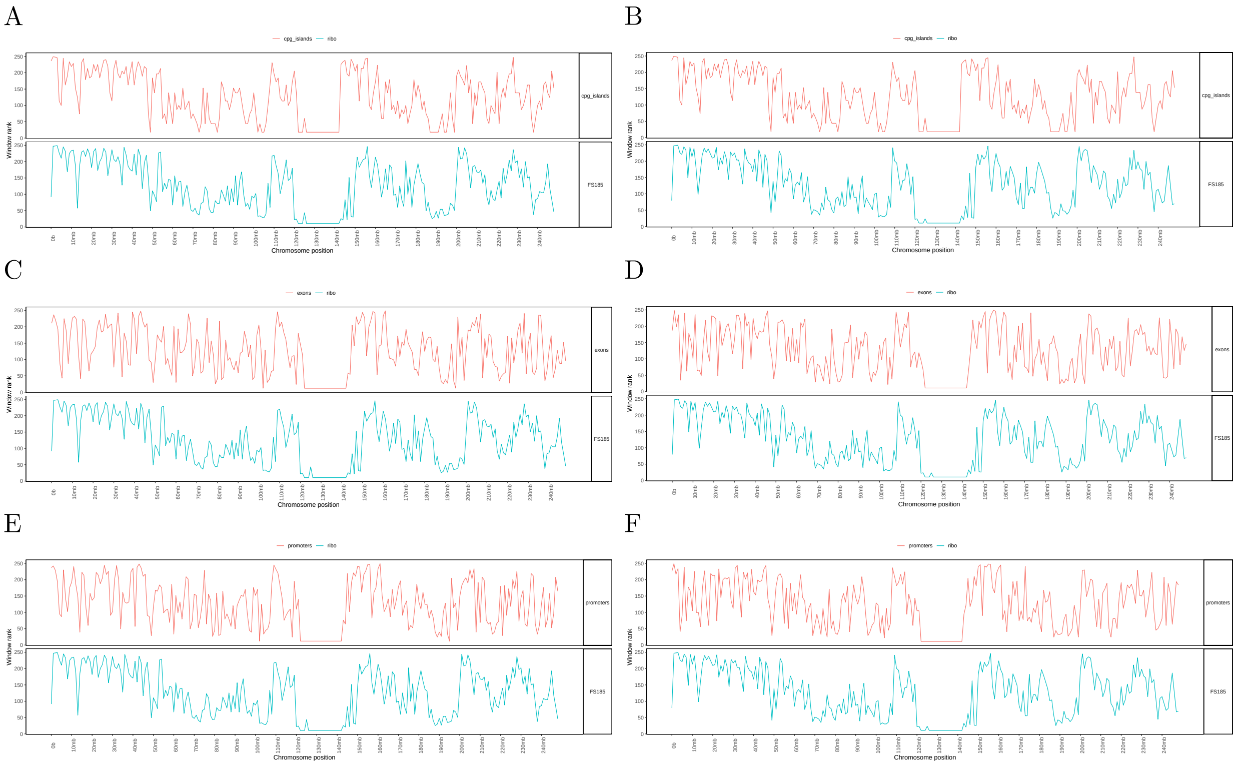

**Figure 11.** Relative frequencies of ribonucleotides (green) vs. annotated features (orange). The annotated features are: CpG islands (A, B), exons (C, D), and promoters (E, F). Sample FS185 only. Chromosome 1. + strand (A, C, E), - strand (B, D, F). Window width = 1mb.

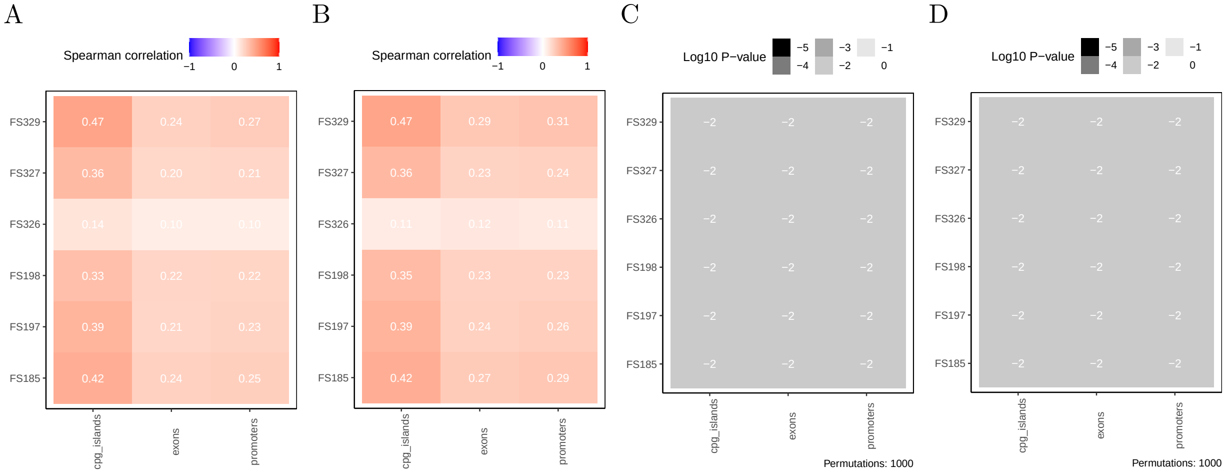

**Figure 12.** Spearman correlations of ribonucleotides with annotated feature counts (A, B). Two-sided permutation p-values of correlations using adjacent swapping scheme (see Section 3.8) (C, D). Chromosome 1. + strand (A, C). - strand (B, D). Window width = 100kb. DNA-seq "depth" on $y$-axis title means "coverage".

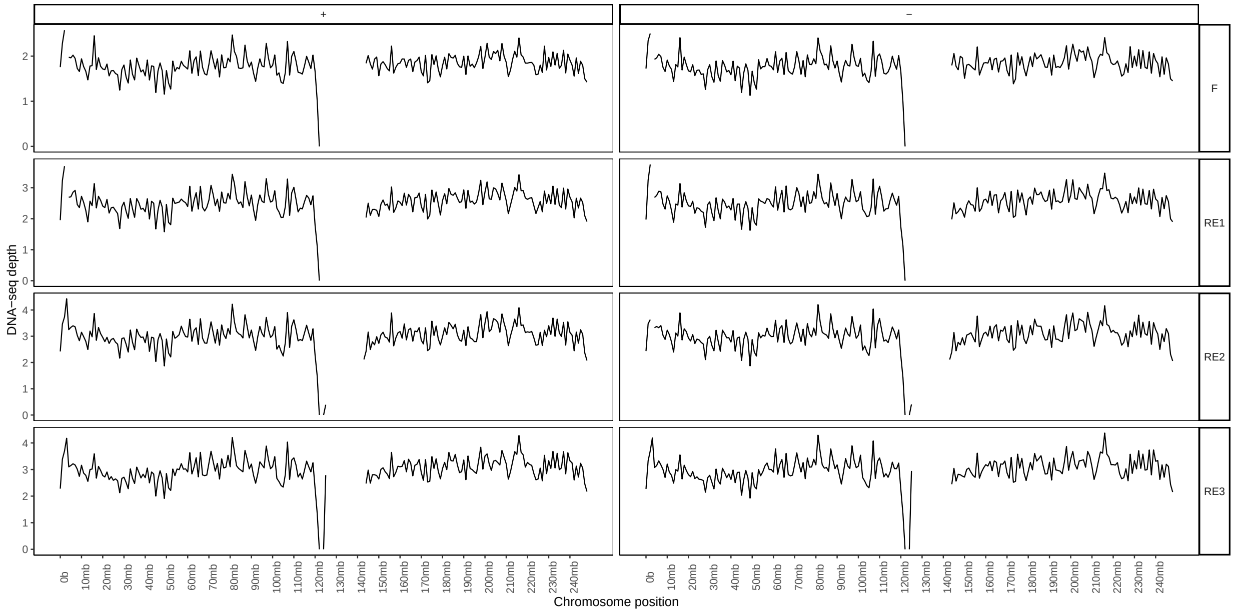

**Figure 13.** DNA-seq coverage vs. chromosome position. Chromosome 1. + strand (left). - strand (right). Window width = 1mb.

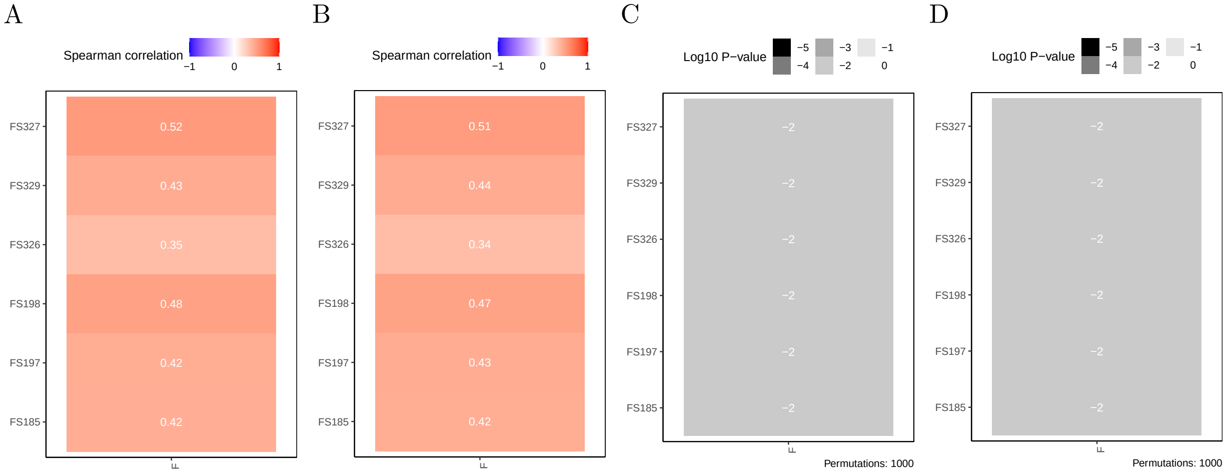

**Figure 14.** Spearman correlations between ribonucleotide counts and Fragmentase (F) DNA-seq coverage (A, B). Two-sided permutation test p-values for correlations using adjacent swapping permutation scheme (see Section 3.8) (C, D). Chromosome 1. + strand (A, C). - strand (B, D). Window width = 100kb.

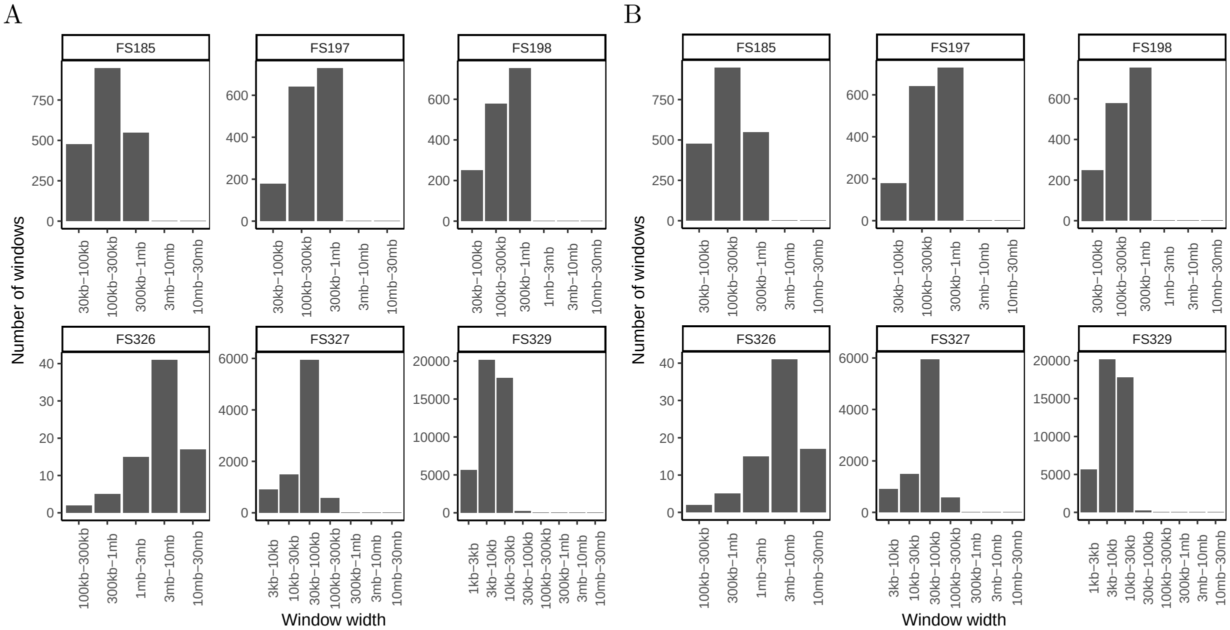

**Figure 15.** Histogram of smoothing-spline window widths. Chromosome 1. + strand (A). - strand (B).

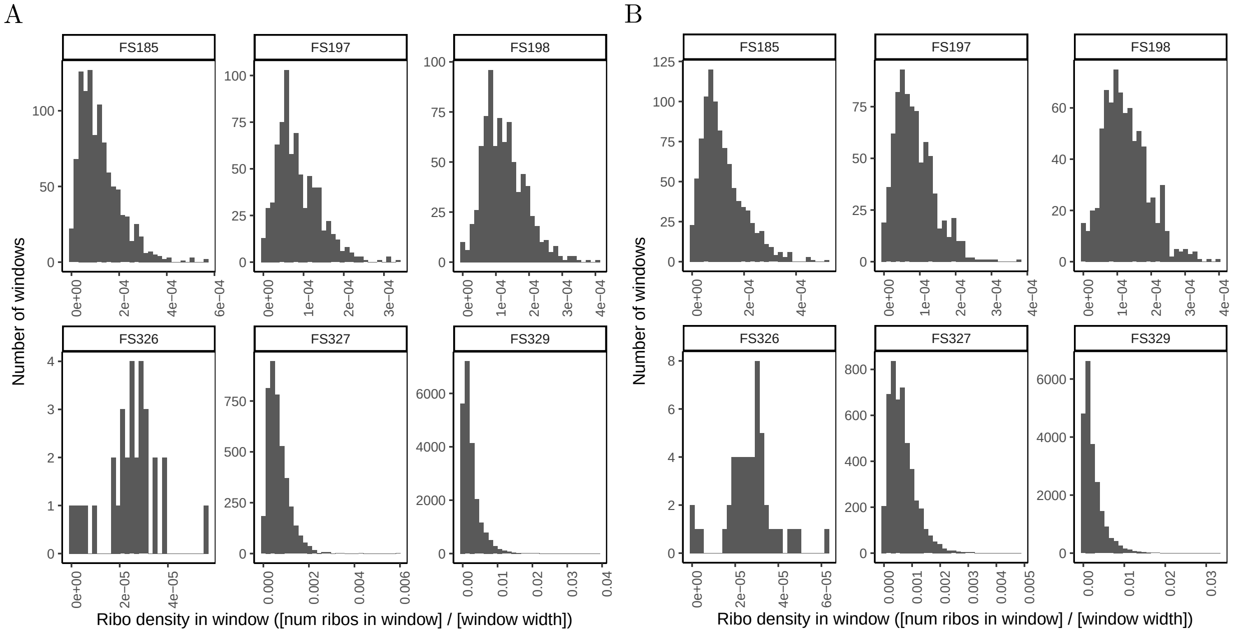

**Figure 16.** Histogram of window densities for smoothing-spline windows. Chromosome 1. + strand (A). - strand (B).

**Table 7.** Summary of the hotspot windows with smoothing-spline window selection. "Chromosome %" is the percentage of the chromosome covered by the hotspot windows. "Ribos %" is the percentage of the samples' ribonucleotides in the hotspot windows. Chromosome 1. + strand (A). - strand (B).

A

| Sample | Chromosome % | Windows | Ribos mean | Ribos % | [Ribos %] / [Chrom. %] |
|---|---|---|---|---|---|
| FS185 | 0.89% | 11 | 100 | 4.10% | 4.61 |
| FS197 | 1.25% | 8 | 115 | 4.70% | 3.76 |
| FS198 | 0.68% | 9 | 64 | 2.06% | 3.05 |
| FS326 | 0.49% | 1 | 68 | 1.11% | 2.27 |
| FS327 | 0.98% | 48 | 127 | 4.21% | 4.29 |
| FS329 | 0.90% | 224 | 147 | 6.45% | 7.15 |

B

| Sample | Chromosome % | Windows | Ribos mean | Ribos % | [Ribos %] / [Chrom. %] |
|---|---|---|---|---|---|
| FS185 | 1.04% | 11 | 105 | 4.36% | 4.20 |
| FS197 | 1.17% | 9 | 95 | 4.37% | 3.73 |
| FS198 | 1.06% | 9 | 100 | 3.19% | 3.02 |
| FS326 | 0.16% | 1 | 25 | 0.40% | 2.50 |
| FS327 | 0.75% | 48 | 104 | 3.44% | 4.60 |
| FS329 | 0.89% | 227 | 144 | 6.41% | 7.21 |

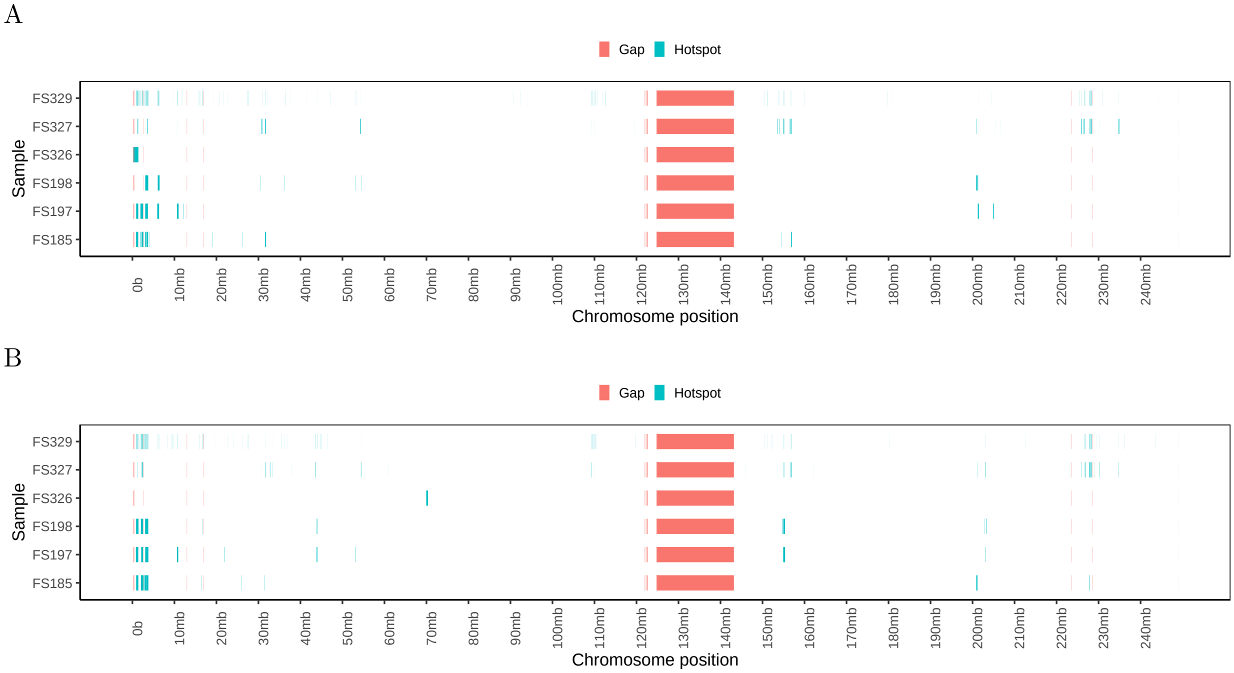

**Figure 17.** Hotspot windows using the smoothing-spline method. Hotspots are defined as windows with the top 1% ribonucleotide density ([total ribos in window] / [window width]) in the sample. "Gaps" are positions on the reference genome with incomplete information. Chromosome 1. + strand (A). - strand (B).

# CHAPTER 5:
# CONCLUSIONS AND FUTURE WORK

In this study, we have analyzed a novel dataset obtained through the ribose-seq protocol in human cells. Our goal was to determine biologically meaningful characteristics of ribonucleotide incorporation in human cells. To do so, we employed a variety of exploratory data analysis techniques such as visualization, statistical modeling, and hypothesis testing with permutations. We also studied the ribose-seq data across multiple samples from different cell types to quantify the variability among genetically different cells. Broadly, we identified the following characteristics of our datasets:

1. The ribonucleotide distribution across the genome appears to be highly conserved among the different cell types. In particular, there appears to be ribonucleotide *hotspots* where multiple samples have an abundance of ribonucleotides in the same genome position.

2. The local ribonucleotide frequency appears to be highly correlated with the local GC content on the chromosome.

3. The ribonucleotide frequency appears to be moderately correlated with the occurrence of functional elements of the genome such as promoters and genes.

These characteristics appear to point towards the systematic incorporation of ribonucleotides in the human genome, rather than it occuring uniformly at random. However, as this study has been purely observational, further study will be required to determine specific mechanisms that contribute to ribonucleotide incorporation. In the following, we interpret our results in more detail and indicate areas for future study.

A important finding of this study was that all the samples analyzed appear highly correlated with each other. This is interesting given that they are different cell types obtained from different sources. This may indicate that ribonucleotide incorporation is a highly conserved aspect of the human genome. Further study will be needed to ascertain whether this is not simply a bias in our data preparation protocols. In the future, the results obtained here by ribose-seq could be checked against other protocols for detecting ribonucleotides such as *emRibo-seq* [57], *HydEn-seq* [14] [84], and *Pu-seq* [18] [37]. Another clear followup analysis would be to examine all twenty-plus samples at our disposal. Since the samples prepared with the Fragmentase enzyme looked very similar to each other, a natural question is whether there is significant variation when different enzymes are used. We may also repeat the analyses with our current datasets for the remaining chromosomes (chromosome 2, 3, ..., 22, and X). Finally, since we have obtained control DNA-seq datasets that describe the background distribution of DNA reads from the genome, we may use these to perform parallel analyses to our ribonucleotide datasets. Then, statistical tests may be used to determine whether the results from the ribonucleotide datasets are significantly different from the control DNA-seq datasets.

Another central goal of our analysis was to determine criteria for classifying regions as ribonucleotide hotspots. The main method employed here was to subdivide chromosome 1 into either equally spaced windows (Section 4.3) or smoothing-spline inflection-point windows (Section 4.8), and take the windows with top 1% of density. For an alternative approach using hypothesis testing, we explored potential null distributions to describe ribonucleotide incorporation, such as the Poisson, negative binomial, and variants thereof. These null models assume that the number of ribonucleotides, $y_i$, detected at each position, $i$, are independent and identically distributed (iid) and come from a specified parametric distribution. The distributions were fit using using maximum likelihood estimation (MLE). For future work, the fitted null distribution may be used to determine p-values, $p_i = \Pr(Y \geq y_i)$, for each $i$, where $Y$ is a random variable with the null distribution. The p-values may then allow us to determine a set of hotspot indices that are so large as to be unlikely under the null distribution. To determine specific p-value thresholds, we may use two criteria: Bonferroni, which is more conservative (less loci detected) and controls the familywise error rate (FWER); and Benjamini-Hochberg [6], which is less conservative (more loci detected) and controls the false-discovery rate (FDR). For the parametric distributions, the most interpretable choice would be a Poisson distribution, since then the null hypothesis has the simple interpretation that the counts are generated by uniform random sampling on the genome. The next step would be to study these loci more finely, such as by examining their nucleotide motifs or their overlaps with annotated regions. Also, since we have used multiple methods for detecting hotspots, we may compare the results across methods to determine whether they are consistent. Another avenue for further modeling would be to use the more complicated hidden Markov model (HMM), which would allow modeling dependent observations as described in Section 2.5.

One of our original questions was whether we could identify nucleotide motifs that were correlated with ribonucleotide abundance. We showed that indeed there appears to be a striking correlation between the local GC content and the ribonucleotide abundance. However, it has been found that certain sequencing technologies may be prone to biases, especially due to local GC content [60] [7]. We may be able to determine whether the GC correlation is due to a sequencing artifact by performing a similar correlation analysis of ribose-seq data in other organisms. If the same pattern holds consistently across multiple organisms, it may indicate that the GC correlation in a sequencing artifact. Conversely, if the correlation is observed only in the human nuclear genome, it may indicate that the correlation is a characteristic of ribonucleotides in the human nuclear genome. If we determine that the correlation is an artifact of the sequencing process, it will be important to mitigate this bias by controlling for the GC content in future analyses. For example, when determining hotspots via parametric distributions, we may add a covariate to the model quantifying the local GC content. Further study could also involve looking at correlations with larger kmers such a di- or tri-nucleotides. An alternative approach to quantifying how well nucleotide motifs are associated with ribonucleotide incorporation would be to use a machine learning (ML) model. For example, Bonidia et al. (2021) [10] outline several methods for extracting numerical features from DNA strings. This could be used with supervised learning to see if a ML model could accurately classify DNA strings as containing or not containing a ribonucleotide. Discovering such a classifier may indicate that nucleotide motifs are indeed associated with ribonuleotide incorporation. However, reverse engineering the classifier to determine the nature of the association may be difficult if the ML model is a "black box".

One of the major drawbacks of our analysis was that we analyzed the whole of chromosome 1 with either 1mb windows for the visualizations or 100kb windows for the statistical analyses. This may be too coarse-grained for detecting finer patterns within our data. For example, in the annotation analysis we showed that there may be potential correlations of the ribonucleotides with genetic elements such as exons and promoters. However, promoter are usually between 100-1000b [44] and exons are usually less than 200b [61]. Thus, we will not be able to precisely detect hotspots at such scales. To improve the precision, we may simply use smaller window widths to bin the ribonucleotide counts. This may be effective for the knockout (KO) samples, since their datasets were relatively dense compared to non-KO samples. We have also relied almost entirely on correlation coefficients (Section 3.7) to assess the strength of relationships. However, more sophisticated analyses can be done, some of which are outlined in Kanduri et al. (2019) [36] and De et al. (2014) [20]. These may involve using different test statistics and permutation strategies. The *regioneR* [24] R package implements some of these strategies, and could be used in future work.

Throughout this study we used permutation tests for testing the statistical significance of our conclusions. However, the validity of these conclusions rests on the validity of the permutation test. All permutation tests made use of the adjacent value swapping scheme described in Section 3.8. Since our window width is 100kb, this means that, roughly, we are testing the sensitivity of the correlations to a random shift of 100kb in the ribonucleotide positions. Changing the window width or the permutation strategy may change the conclusions. For future work we may explore the sensitivity of the correlations to smaller shifts. We also note that the permutation strategy used here is a heuristic that, as far we know, does not have a precedent in the literature or a biological basis. A more reasonable approach may be to use existing methods for the same purpose, such as the *bootRanges* [46] R software based on the *block bootstrap* method introduced by Bickel et al. (2010) [8]. A related method is the *matchRanges* [19] R package, which we may use to generate bootstrapped datasets from the original, while controlling specific characteristics in the bootstrap datasets such as correlation with local GC content.

In summary, we have begun the work of exploring ribonucleotide incorporation in the human nuclear genome. However, there is considerable work to do in terms of refining our analyses and applying them to more datasets. The most important challenge will be to synthesize our observations into a biologically plausible and testable hypothesis as to why ribonucleotides are distributed on the genome the way they are. Ultimately, we hope this will lead to a better understanding of DNA replication and genome stability.

# REFERENCES

[1] "A global reference for human genetic variation". In: *Nature* 526.7571 (2015), pp. 68–74. ISSN: 0028-0836. DOI: 10.1038/nature15393. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4750478/ (visited on 02/24/2023).

[2] Sathya Balachander et al. "Capture of Ribonucleotides in Yeast Genomic DNA Using Ribose-Seq". en. In: *Yeast Systems Biology: Methods and Protocols*. Ed. by Stephen G. Oliver and Juan I. Castrillo. Methods in Molecular Biology. New York, NY: Springer, 2019, pp. 17–37. ISBN: 978-1-4939-9736-7. DOI: 10.1007/978-1-4939-9736-7_2. URL: https://doi.org/10.1007/978-1-4939-9736-7_2 (visited on 03/02/2023).

[3] Sathya Balachander et al. "Ribonucleotide incorporation in yeast genomic DNA shows preference for cytosine and guanosine preceded by deoxyadenosine". en. In: *Nature Communications* 11.1 (Dec. 2020), p. 2447. ISSN: 2041-1723. DOI: 10.1038/s41467-020-16152-5. URL: http://www.nature.com/articles/s41467-020-16152-5 (visited on 05/23/2022).

[4] Timothy M. Beissinger. *GenWin: Spline Based Window Boundaries for Genomic Analyses*. R package version 1.0. 2022. URL: https://CRAN.R-project.org/package=GenWin.

[5] Timothy M. Beissinger et al. "Defining window-boundaries for genomic analyses using smoothing spline techniques". In: *Genetics Selection Evolution* 47.1 (Apr. 2015), p. 30. ISSN: 1297-9686. DOI: 10.1186/s12711-015-0105-9. URL: https://doi.org/10.1186/s12711-015-0105-9 (visited on 12/29/2022).

[6] Yoav Benjamini and Yosef Hochberg. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1 (1995). Publisher: [Royal Statistical Society, Wiley], pp. 289–300. ISSN: 0035-9246. URL: https://www.jstor.org/stable/2346101 (visited on 03/02/2023).

[7] Yuval Benjamini and Terence P. Speed. "Summarizing and correcting the GC content bias in high-throughput sequencing". In: *Nucleic Acids Research* 40.10 (May 2012), e72. ISSN: 0305-1048. DOI: 10.1093/nar/gks001. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3378858/ (visited on 03/23/2023).

[8] Peter J. Bickel et al. “Subsampling methods for genomic inference”. In: *The Annals of Applied Statistics* 4.4 (Dec. 2010). Publisher: Institute of Mathematical Statistics, pp. 1660–1697. ISSN: 1932-6157, 1941-7330. DOI: 10.1214/10-AOAS363. URL: https://projecteuclid.org/journals/annals-of-applied-statistics/volume-4/issue-4/Subsampling-methods-for-genomic-inference/10.1214/10-AOAS363.full (visited on 02/02/2023).

[9] New England Biolabs. *NEBNext® dsDNA Fragmentase® | NEB*. commercial. 2023. URL: https://www.neb.com/products/m0348-nebnext-dsdna-fragmentase (visited on 02/25/2023).

[10] Robson P Bonidia et al. “Feature extraction approaches for biological sequences: a comparative study of mathematical features”. In: *Briefings in Bioinformatics* 22.5 (Sept. 2021), bbab011. ISSN: 1477-4054. DOI: 10.1093/bib/bbab011. URL: https://doi.org/10.1093/bib/bbab011 (visited on 03/16/2023).

[11] Marc Carlson. *org.Hs.eg.db: Genome wide annotation for Human*. R package version 3.16.0. 2022.

[12] Raymond G Cavalcante and Maureen A Sartor. “annotatr: genomic regions in context”. In: *Bioinformatics* 33.15 (Aug. 2017), pp. 2381–2383. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx183. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5860117/ (visited on 02/07/2023).

[13] Raymond G Cavalcante and Maureen A Sartor. “annotatr: genomic regions in context.” In: *Bioinformatics* (2017). R package version 1.24.0.

[14] Anders R. Clausen et al. “Tracking replication enzymology in vivo by genome-wide mapping of ribonucleotide incorporation”. In: *Nature structural & molecular biology* 22.3 (Mar. 2015), pp. 185–191. ISSN: 1545-9993. DOI: 10.1038/nsmb.2957. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4351163/ (visited on 02/25/2023).

[15] Peter Craven and Grace Wahba. “Smoothing noisy data with spline functions”. en. In: *Numerische Mathematik* 31.4 (Dec. 1978), pp. 377–403. ISSN: 0945-3245. DOI: 10.1007/BF01404567. URL: https://doi.org/10.1007/BF01404567 (visited on 12/29/2022).

[16] Yanick J. Crow et al. “Mutations in genes encoding ribonuclease H2 subunits cause Aicardi-Goutières syndrome and mimic congenital viral brain infection”. en. In: *Nature Genetics* 38.8 (Aug. 2006). Number: 8 Publisher: Nature Publishing Group, pp. 910–916. ISSN: 1546-1718. DOI: 10.1038/ng1842. URL: https://www.nature.com/articles/ng1842 (visited on 03/03/2023).

[17] David B. Dahl et al. *xtable: Export Tables to LaTeX or HTML*. R package version 1.8-4. 2019. URL: https://CRAN.R-project.org/package=xtable.

[18] Yasukazu Daigaku et al. “A global profile of replicative polymerase usage”. In: *Nature structural & molecular biology* 22.3 (Mar. 2015), pp. 192–198. ISSN: 1545-9993. DOI: 10.1038/nsmb.2962. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4789492/ (visited on 02/25/2023).

[19] Eric S. Davis et al. *matchRanges: Generating null hypothesis genomic ranges via covariate-matched sampling*. en. Pages: 2022.08.05.502985 Section: New Results. Aug. 2022. DOI: 10.1101/2022.08.05.502985. URL: https://www.biorxiv.org/content/10.1101/2022.08.05.502985v1 (visited on 02/03/2023).

[20] Subhajyoti De, Brent S. Pedersen, and Katerina Kechris. “The dilemma of choosing the ideal permutation strategy while estimating statistical significance of genome-wide enrichment”. In: *Briefings in Bioinformatics* 15.6 (Nov. 2014), pp. 919–928. ISSN: 1467-5463. DOI: 10.1093/bib/bbt053. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4271068/ (visited on 02/02/2023).

[21] Mikhail G Dozmorov. “Epigenomic annotation-based interpretation of genomic data: from enrichment analysis to machine learning”. In: *Bioinformatics* 33.20 (Oct. 2017), pp. 3323–3330. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx414. URL: https://doi.org/10.1093/bioinformatics/btx414 (visited on 01/08/2023).

[22] Zhide Fang, Jeffrey Martin, and Zhong Wang. “Statistical methods for identifying differentially expressed genes in RNA-Seq experiments”. In: *Cell & Bioscience* 2.1 (July 2012), p. 26. ISSN: 2045-3701. DOI: 10.1186/2045-3701-2-26. URL: https://doi.org/10.1186/2045-3701-2-26 (visited on 01/25/2023).

[23] Jane Fridlyand et al. “Hidden Markov models approach to the analysis of array CGH data”. en. In: *Journal of Multivariate Analysis*. Special Issue on Multivariate Methods in Genomic Data Analysis 90.1 (July 2004), pp. 132–153. ISSN: 0047-259X. DOI: 10.1016/j.jmva.2004.02.008. URL: https://www.sciencedirect.com/science/article/pii/S0047259X04000260 (visited on 02/01/2023).

[24] Bernat Gel et al. “regioneR: an R/Bioconductor package for the association analysis of genomic regions based on permutation tests”. In: *Bioinformatics* 32.2 (Jan. 2016), pp. 289–291. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btv562. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4708104/ (visited on 01/10/2023).

[25] Alli L Gombolay, Fredrik O Vannberg, and Francesca Storici. “Ribose-Map: a bioinformatics toolkit to map ribonucleotides embedded in genomic DNA”. In: *Nucleic Acids Research* 47.1 (Jan. 2019), e5. ISSN: 0305-1048. DOI: 10.1093/nar/gky874. URL: https://doi.org/10.1093/nar/gky874 (visited on 02/25/2023).

[26] Peter J Green and Bernard W Silverman. “Generalized linear models”. In: *Nonparametric Regression and Generalized Linear Models*. Boston, MA: Springer US, 1994, pp. 89–114. ISBN: 978-0-412-30040-0.

[27] Garrett Grolemund and Hadley Wickham. “Dates and Times Made Easy with lubridate”. In: *Journal of Statistical Software* 40.3 (2011), pp. 1–25. URL: https://www.jstatsoft.org/v40/i03/.

[28] Arief Gusnanto et al. “Estimating optimal window size for analysis of low-coverage next-generation sequence data”. In: *Bioinformatics* 30.13 (July 2014), pp. 1823–1829. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btu123. URL: https://doi.org/10.1093/bioinformatics/btu123 (visited on 12/28/2022).

[29] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*. en. 2nd ed. Springer series in statistics. New York, NY: Springer, Feb. 2009.

[30] Jesse Hemerik and Jelle Goeman. “Exact testing with random permutations”. In: *Test (Madrid, Spain)* 27.4 (2018), pp. 811–825. ISSN: 1133-0686. DOI: 10.1007/s11749-017-0571-1. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6405018/ (visited on 01/23/2023).

[31] Lionel Henry and Hadley Wickham. *purrr: Functional Programming Tools*. R package version 0.3.5. 2022. URL: https://CRAN.R-project.org/package=purrr.

[32] Tao Huang et al. “Detection of DNA copy number alterations using penalized least squares regression”. en. In: *Bioinformatics* 21.20 (Oct. 2005), pp. 3811–3817.

[33] W. Huber et al. “Orchestrating high-throughput genomic analysis with Bioconductor”. In: *Nature Methods* 12.2 (2015), pp. 115–121. URL: http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html.

[34] Illumina. *Sequencing Read Length | How to calculate NGS read length*. en. commercial. URL: https://www.illumina.com/science/technology/next-generation-sequencing/plan-experiments/read-length.html (visited on 02/24/2023).

[35] S original by Jim Ramsey. R port by Brian Ripley <ripley@stats.ox.ac.uk>. *pspline: Penalized Smoothing Splines*. R package version 1.0-19. 2022. URL: https://CRAN.R-project.org/package=pspline.

[36] Chakravarthi Kanduri et al. “Colocalization analyses of genomic elements: approaches, recommendations and challenges”. In: *Bioinformatics* 35.9 (May 2019), pp. 1615–1624. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty835. URL: https://doi.org/10.1093/bioinformatics/bty835 (visited on 01/07/2023).

[37] Andrea Keszthelyi et al. "Mapping ribonucleotides in genomic DNA and exploring replication dynamics by polymerase usage sequencing (Pu-seq)". en. In: *Nature Protocols* 10.11 (Nov. 2015). Number: 11 Publisher: Nature Publishing Group, pp. 1786–1801. ISSN: 1750-2799. DOI: 10.1038/nprot.2015.116. URL: https://www.nature.com/articles/nprot.2015.116 (visited on 03/02/2023).

[38] Kyung Duk Koh et al. "Ribose-seq: global mapping of ribonucleotides embedded in genomic DNA". en. In: *Nature Methods* 12.3 (Mar. 2015), pp. 251–257. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/nmeth.3259. URL: https://www.nature.com/articles/nmeth.3259 (visited on 05/23/2022).

[39] Diane Lambert. "Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing". In: *Technometrics* 34.1 (1992). Publisher: [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality], pp. 1–14. ISSN: 0040-1706. DOI: 10.2307/1269547. URL: https://www.jstor.org/stable/1269547 (visited on 01/25/2023).

[40] Eric S. Lander et al. "Initial sequencing and analysis of the human genome". en. In: *Nature* 409.6822 (Feb. 2001). Number: 6822 Publisher: Nature Publishing Group, pp. 860–921. ISSN: 1476-4687. DOI: 10.1038/35057062. URL: https://www.nature.com/articles/35057062 (visited on 02/24/2023).

[41] Ben Langmead and Steven L Salzberg. "Fast gapped-read alignment with Bowtie 2". In: *Nature methods* 9.4 (Mar. 2012), pp. 357–359. ISSN: 1548-7091. DOI: 10.1038/nmeth.1923. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3322381/ (visited on 02/24/2023).

[42] Michael Lawrence, Robert Gentleman, and Vincent Carey. "rtracklayer: an R package for interfacing with genome browsers". In: *Bioinformatics* 25 (2009), pp. 1841–1842. DOI: 10.1093/bioinformatics/btp328. URL: http://bioinformatics.oxfordjournals.org/content/25/14/1841.abstract.

[43] Michael Lawrence et al. "Software for Computing and Annotating Genomic Ranges". In: *PLoS Computational Biology* 9 (8 2013). DOI: 10.1371/journal.pcbi.1003118. URL: http://www.ploscompbiol.org/article/info%3Adoi%2F10.1371%2Fjournal.pcbi.1003118.

[44] Nguyen Quoc Khanh Le et al. "Classifying Promoters by Interpreting the Hidden Information of DNA Sequences via Deep Learning and Combination of Continuous FastText N-Grams". In: *Frontiers in Bioengineering and Biotechnology* 7 (2019). ISSN: 2296-4185. URL: https://www.frontiersin.org/articles/10.3389/fbioe.2019.00305 (visited on 02/20/2023).

[45] Lee et al. "plyranges: a grammar of genomic data transformation". In: *Genome Biol.* 20.1 (2019), p. 4. URL: http://dx.doi.org/10.1186/s13059-018-1597-8.

[46] Wancen Mu et al. *bootRanges: Flexible generation of null sets of genomic ranges for hypothesis testing*. en. Sept. 2022. DOI: 10.1101/2022.09.02.506382. URL: https://www.biorxiv.org/content/10.1101/2022.09.02.506382v1 (visited on 02/02/2023).

[47] Kirill Müller and Hadley Wickham. *tibble: Simple Data Frames*. R package version 3.1.8. 2022. URL: https://CRAN.R-project.org/package=tibble.

[48] Yue S. Niu, Ning Hao, and Heping Zhang. "Multiple Change-Point Detection: A Selective Overview". In: *Statistical Science* 31.4 (2016). Publisher: Institute of Mathematical Statistics, pp. 611–623. ISSN: 0883-4237. URL: https://www.jstor.org/stable/26408091 (visited on 01/02/2023).

[49] Adam B. Olshen et al. "Circular binary segmentation for the analysis of array-based DNA copy number data". In: *Biostatistics* 5.4 (Oct. 2004), pp. 557–572. ISSN: 1465-4644. DOI: 10.1093/biostatistics/kxh008. URL: https://doi.org/10.1093/biostatistics/kxh008 (visited on 02/01/2023).

[50] H. Pagès et al. *Biostrings: Efficient manipulation of biological strings*. R package version 2.66.0. 2022. URL: https://bioconductor.org/packages/Biostrings.

[51] Belinda Phipson and Gordon K. Smyth. "Permutation P-values Should Never Be Zero: Calculating Exact P-values When Permutations Are Randomly Drawn". en. In: *Statistical Applications in Genetics and Molecular Biology* 9.1 (Oct. 2010). Publisher: De Gruyter. ISSN: 1544-6115. DOI: 10.2202/1544-6115.1585. URL: https://www.degruyter.com/document/doi/10.2202/1544-6115.1585/html (visited on 01/13/2023).

[52] Allison Piovesan et al. "On the length, weight and GC content of the human genome". In: *BMC Research Notes* 12 (Feb. 2019), p. 106. ISSN: 1756-0500. DOI: 10.1186/s13104-019-4137-z. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6391780/ (visited on 02/04/2023).

[53] R. L. Plackett. "The Truncated Poisson Distribution". In: *Biometrics* 9.4 (1953). Publisher: [Wiley, International Biometric Society], pp. 485–488. ISSN: 0006-341X. DOI: 10.2307/3001439. URL: https://www.jstor.org/stable/3001439 (visited on 01/25/2023).

[54] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2022. URL: https://www.R-project.org/.

[55] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2022. URL: https://www.R-project.org/.

[56] L.R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (Feb. 1989). Conference Name: Proceedings of the IEEE, pp. 257–286. ISSN: 1558-2256. DOI: 10.1109/5.18626.

[57] Martin A. M. Reijns et al. “Lagging-strand replication shapes the mutational landscape of the genome”. en. In: *Nature* 518.7540 (Feb. 2015). Number: 7540 Publisher: Nature Publishing Group, pp. 502–506. ISSN: 1476-4687. DOI: 10.1038/nature14183. URL: https://www.nature.com/articles/nature14183 (visited on 02/25/2023).

[58] Jason A. Reuter, Damek V. Spacek, and Michael P. Snyder. “High-Throughput Sequencing Technologies”. en. In: *Molecular Cell* 58.4 (May 2015), pp. 586–597. ISSN: 1097-2765. DOI: 10.1016/j.molcel.2015.05.004. URL: https://www.sciencedirect.com/science/article/pii/S1097276515003408 (visited on 02/14/2023).

[59] R. A. Rigby and D. M. Stasinopoulos. “Generalized additive models for location, scale and shape,(with discussion)”. In: *Applied Statistics* 54 (2005), pp. 507–554.

[60] Michael G. Ross et al. “Characterizing and measuring bias in sequence data”. In: *Genome Biology* 14.5 (May 2013), R51. ISSN: 1474-760X. DOI: 10.1186/gb-2013-14-5-r51. URL: https://doi.org/10.1186/gb-2013-14-5-r51 (visited on 02/18/2023).

[61] Meena Kishore Sakharkar, Vincent T. K. Chow, and Pandjassarame Kangueane. “Distributions of exons and introns in the human genome”. eng. In: *In Silico Biology* 4.4 (2004), pp. 387–393. ISSN: 1386-6338.

[62] Akira Sassa, Manabu Yasui, and Masamitsu Honma. “Current perspectives on mechanisms of ribonucleotide incorporation and processing in mammalian DNA”. en. In: *Genes and Environment* 41.1 (Dec. 2019), p. 3. ISSN: 1880-7062. DOI: 10.1186/s41021-019-0118-7. URL: https://genesenvironment.biomedcentral.com/articles/10.1186/s41021-019-0118-7 (visited on 05/23/2022).

[63] Valerie A. Schneider et al. “Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly”. en. In: *Genome Research* 27.5 (May 2017). Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab, pp. 849–864. ISSN: 1088-9051, 1549-5469. DOI: 10.1101/gr.213611.116. URL: https://genome.cshlp.org/content/27/5/849 (visited on 02/24/2023).

[64] Ashish Sen and Muni S. Srivastava. “On Tests for Detecting Change in Mean”. In: *The Annals of Statistics* 3.1 (Jan. 1975). Publisher: Institute of Mathematical Statistics, pp. 98–108. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/aos/1176343001. URL: https://projecteuclid.org/journals/annals-of-statistics/volume-3/issue-1/On-Tests-for-Detecting-Change-in-Mean/10.1214/aos/1176343001.full (visited on 02/25/2023).

[65] Smit, AFA, Hubley, R & Green, P. *RepeatMasker Open-4.0. 2013-2015 <http://www.repeatmasker.org>*.

[66] Mikis Stasinopoulos and Bob Rigby. *gamlss.tr: Generating and Fitting Truncated 'gamlss.family' Distributions*. R package version 5.1-7. 2020. URL: https://CRAN.R-project.org/package=gamlss.tr.

[67] Mikis Stasinopoulos and Robert Rigby. *gamlss.dist: Distributions for Generalized Additive Models for Location Scale and Shape*. R package version 6.0-5. 2022. URL: https://CRAN.R-project.org/package=gamlss.dist.

[68] Bioconductor Core Team and Bioconductor Package Maintainer. *TxDb.Hsapiens.UCSC.hg38.knownGene: Annotation package for TxDb object(s)*. R package version 3.16.0. 2022.

[69] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996). Publisher: [Royal Statistical Society, Wiley], pp. 267–288. ISSN: 0035-9246. DOI: 10.2307/2346178. URL: https://www.jstor.org/stable/2346178 (visited on 12/27/2022).

[70] Ingmar Visser and Maarten Speekenbrink. "depmixS4: An R Package for Hidden Markov Models". In: *Journal of Statistical Software* 36.7 (2010), pp. 1–21. URL: https://www.jstatsoft.org/v36/i07/.

[71] G. Wahba and S. Wold. "A completely automatic french curve: fitting spline functions by cross validation". In: *Communications in Statistics* 4.1 (Jan. 1975). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/03610927508827223, pp. 1–17. ISSN: 0090-3272. DOI: 10.1080/03610927508827223. URL: https://doi.org/10.1080/03610927508827223 (visited on 01/03/2023).

[72] G. Wahba and S. Wold. "Periodic splines for spectral density estimation: the use of cross validation for determining the degree of smoothing". In: *Communications in Statistics* 4.2 (Jan. 1975). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/03610927508827233, pp. 125–141. ISSN: 0090-3272. DOI: 10.1080/03610927508827233. URL: https://doi.org/10.1080/03610927508827233 (visited on 02/27/2023).

[73] Hadley Wickham. *forcats: Tools for Working with Categorical Variables (Factors)*. R package version 0.5.2. 2022. URL: https://CRAN.R-project.org/package=forcats.

[74] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4. URL: https://ggplot2.tidyverse.org.

[75] Hadley Wickham. *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.5.0. 2022. URL: https://CRAN.R-project.org/package=stringr.

[76] Hadley Wickham and Maximilian Girlich. *tidyr: Tidy Messy Data*. R package version 1.2.1. 2022. URL: https://CRAN.R-project.org/package=tidyr.

[77] Hadley Wickham, Jim Hester, and Jennifer Bryan. *readr: Read Rectangular Text Data*. R package version 2.1.3. 2022. URL: https://CRAN.R-project.org/package=readr.

[78] Hadley Wickham et al. *dplyr: A Grammar of Data Manipulation*. R package version 1.0.10. 2022. URL: https://CRAN.R-project.org/package=dplyr.

[79] Hadley Wickham et al. "Welcome to the tidyverse". In: *Journal of Open Source Software* 4.43 (2019), p. 1686. DOI: 10.21105/joss.01686.

[80] Jessica S. Williams and Thomas A. Kunkel. "Ribonucleotides in DNA: Origins, repair and consequences". In: *DNA repair* 19 (July 2014), pp. 27–37. ISSN: 1568-7864. DOI: 10.1016/j.dnarep.2014.03.029. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4065383/ (visited on 02/25/2023).

[81] Penghao Xu and Francesca Storici. "RESCOT: Restriction enzyme set and combination optimization tools for rNMP capture techniques". en. In: *Theoretical Computer Science*. Building Bridges – Honoring Nataša Jonoska on the Occasion of Her 60th Birthday 894 (Nov. 2021), pp. 203–213. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2021.08.006. URL: https://www.sciencedirect.com/science/article/pii/S0304397521004655 (visited on 03/01/2023).

[82] Byung-Jun Yoon. "Hidden Markov Models and their Applications in Biological Sequence Analysis". In: *Current Genomics* 10.6 (Sept. 2009), pp. 402–415. ISSN: 1389-2029. DOI: 10.2174/138920209789177575. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2766791/ (visited on 02/27/2023).

[83] Zhi-Xiong Zhou et al. "Ribonucleotide incorporation into DNA during DNA replication and its consequences". In: *Critical Reviews in Biochemistry and Molecular Biology* 56.1 (Jan. 2021), pp. 109–124. ISSN: 1040-9238. DOI: 10.1080/10409238.2020.1869175. URL: https://doi.org/10.1080/10409238.2020.1869175 (visited on 01/08/2023).

[84] Zhi-Xiong Zhou et al. "Roles for DNA polymerase  in initiating and terminating leading strand DNA replication". In: *Nature Communications* 10 (Sept. 2019), p. 3992. ISSN: 2041-1723. DOI: 10.1038/s41467-019-11995-z. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6728351/ (visited on 03/02/2023).