ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Master Thesis
Chair of Bioinformatics
Department of Computer Sceince
Albert-Ludwigs-University of Freiburg

Local sequence and structure features in long RNAs

Hoor K. Al-Hasani

November 8, 2011

Supervisors
Prof. Dr. Rolf Backofen[1]
Prof. Dr. Christian Schindelhauer[2]

[1] Chair of Bioinformatics, University of Freiburg
[2] Chair of Computer Networks and Telematics, University of Freiburg

Abstract

This work proposes an automatic method to analyze different data sets based on their features. The features of human lincRNA were compared to those of mRNA.
The features include sequences compositions, structures, and base-pairs probabilities.

Afterward, a generic Background Model (BGM) was computed, which fitted the values of these features to a selection of distributions. The fitted distribution is used as a basis to determine cutoff values for significant regions within the sequences for different significant levels. With the help of the BGM, significant regions of lincRNA sequences could be highlighted and compared between different features.

Moreover, it was used to compare the similarity of the feature distributions in lincRNA with those of the different parts of the mRNA, namely the 5'UTR, the coding sequence, and the 3'UTR.

Finally, each step in our approach was visualized in automatic graphs to enable an easier identification of regions of interest or to manually check the computed results.

Zusammenfassung

In dieser Arbeit wird eine Methode vorgestellt um Datensätze mit vielen RNA-Sequenzen zu analysieren. Insbesondere soll das Verfahren dazu dienen, unbekannte RNA Sequenzen zu analysieren und deren mögliche Funktionen besser zu verstehen. Grundlage bilden dabei lokale Sequenz- und Sequenz-Struktureigenschaften der einzelnen Sequenzen. Diese lokalen Eigenschaften können dann für verschiedene RNA Klassen mit Hilfe statistischer Methoden verglichen werden.

In der vorliegenden Arbeit werden exemplarisch humane lincRNAs und humane mRNAs miteinander verglichen. Über die Klasse der lincRNAs ist bis auf einzelne Beispiele wie die XIST-RNA wenig bekannt. Die Klasse der mRNAs ist weitaus bessser verstanden.

Als Eigenschaften werden Mono- und Dinukleotid Gehalt der Sequenzen sowie Eigenschaften der lokalen Sekundärstruktur heranzgezogen. Um Signifikante Eigenschaften herauszufinden, werden für die Analyse vier Hintergrundmodelle betrachtet: jeweils eins für die drei unterschiedlichen Regionen einer mRNA sowie eins für randomisierte lincRNAs. Anschließend wird das Modell der lincRNAs mit den Hintergrundmodellen verglichen um Ähnlichkeiten und Unterschiede zwischen lincRNAs und mRNAs herauszufinden.

Jeder der einzelnen Analyseschritte lässt sich visualisieren um das der Ergbnis nachvollziehbar zu machen. Weiterhin lassen sich auch einzelne Sequenzen und deren Signifikante Bereiche Graphisch darstellen.

Acknowledgments

It is a pleasure to thank those who made this work possible, and to show my gratefulness to my supervisors, family, colleges and friends.

I would like to thank dearly Prof.Dr. Rolf Backofen for supervising and supporting me; and wish to extend my heartfelt thank to Prof.Dr. Christian Schindelhauer for always being there, and the understanding he showed me during my entire study.
Being under their supervision was a great change and addition; to them both I'm truly indebted .

A special thank to my supportive supervisors Steffen Heyen and Sita Lange who had a direct influence on this work and never hesitated in guiding and encouraging me all the way. Being with them and other members of Bio-informatics group Freiburg was such a pleasure and eyes opening experience.

I wish I can properly thank Dr. Dominic Rose, Dr. Martin Mann, Dr.Stefan Rührup and DAAD, for all the help, support and kindness they generously showed me during my master study.
Moreover, I would like to thank Fraunhofer institut IZI for allowing me finishing my thesis, and especially Dr. Kristin Reiche.

My deepest gratitude to my wonderful dear family, for being there in a number of ways. I dedicate this work to them all and particularly to my mother Dr. Bushra Bakir, and to my father Dr. Kadham Al-Hasani.

Finally, to my colleges and to every one made my stay in Freiburg such a rich and delightful experience: ganz herzlichen Dank!

Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Nowadays non-coding RNA is drawing more and more attention to itself. 98% of the human genome is not translated, thus it is believed that non coding parts are essential for the complexity of cells. This percentage decreases once the cell is less complicated [1]. In the last decade, a number of ncRNA classes were discovered that are involved in many different regulatory functions. For example, long ncRNA (lncRNA), piwi-interacting RNA (piRNA), and small nucleolar RNA (snoRNA) [2, 3, 4].

Many ncRNA classes have been identified by means of computational prediction and conservation, however their functions are still unknown or vaguely understood. Furthermore, some classes of ncRNA are highly conserved indicating the fact that they have been maintained by various species for a particular purpose.

Remarkably, an abnormality in the expression of various ncRNAs was observed in number of diseases. For example, cancer, alzheimer, and Prader-Willi syndrome [5].

A major incident in the ncRNA world namely the discovery of microRNA, was in 2001, which in turn led to a series of subsequent discoveries. The length of microRNA can vary from 19-25 nucleotides (nt); microRNA became the key in humans diseases, for its involvement in different human diseases and activities of the cell starting from various cancer metastasis to binding ncRNA with its targets in mRNA. A recent paper [6] proposed that mRNAs and ncRNAs have a hidden language, in which microRNA response elements (MREs) are the broker.

Lately a new ncRNA subtype was discovered called **Large intervening non-coding RNAs** (lincRNA). LincRNA were first identified in mouse cells [7], later in human and

another 21 mammalians [8]. Accordingly, about 3,289 lincRNAs were identified, which are highly conserved in mammalian and are often found to be structured[1]. Only very few of them have been functionally classified, however. known examples include XIST (X-inactive specific transcript) and HOTAIR (HOX antisense intergenic RNA); XIST is conserved among several species and play a major role in X inactivation [9]. HOTAIR is involved directly in repressing the gene basically with PRC2 (Polycomb Repeessive Complex 2) [8].
lincRNA is strongly assumed to play a role in RNA binding proteins, cell-cycle regulation, and innate immune system [7, 10].

Similarly to other ncRNA types, abnormality in lincRNA expression was seen to be a contributing factor in diseases for example cancer metastasis [8, 11].
A breast-cancer therapy approach has been proposed in [11], by targeting lincRNA with small interfering RNA (siRNA) to limit its role in cancer progression. The same approach was shown in an earlier study [12].

Biological aims

lincRNAs have been identified only recently, therefore many of its characteristics are not determined yet, this work investigates general properties of lincRNA sequences on the basis of sequence and structure features.

The most similar class of known RNA molecules are mRNAs due to their length distributions and both are processed by the splicing machinery. As the evidences indicate, mRNA and lncRNAs have a thick relationship [13].
In addition mRNAs are generally unstructured due to its main messenger function for protein production where the ribosomes frequently travel down the sequence, however, many regulatory elements are embedded within sequence that regulate translation and stabilize or destabilize the molecule. These regulatory elements are usually local, i.e. span across only a small region of the mRNA.

For this reason, the task of this thesis is to compare local sequence and structure features of lincRNAs to mRNAs. A mRNA molecule is divided into three main functional sections 5'UTR, CDS, and 3'UTR (see Sec. 2.3)

In this work, this comparison will be made by generating the features of both classes. As a result, we would like to know whether it is possible at all to make such a comparison, and what is the outcome of that. Moreover being able to address the differences between these classes.

[1] Yet to be investigated.

Scientific aims

The scientific aim of this work is to build a statistical analysis to RNA molecule. With the help of the proposed automatic analysis (Background model-BGM), additional properties in lincRNAs should be possible to recognize, such as identifying potentially functional local regions on single lincRNA sequences. Since there is only very little information available on such functional regions, we make the assumption that functional regions should show significant properties.

A property that is ubiquitous is unlikely to be functional due to a lack of specificity. Therefore, we approximate functional regions by identifying regions with significant feature values. A feature is a computational model of a local property that can be mapped to a single nucleotide position.

These regions of significant feature values should be visualized in a single graph so that it is easy to identify their location and correlations between different features. The identification of these significant and potentially functional regions can be a basis for future biological experiments and homology analyses.

Prerequisites

After obtaining dataset for both lincRNAs and mRNAs:

- feature selection, generation and visualization. It must be possible to assign a value for each feature to a single nucleotide position without gaps in the RNA sequence.
- compare feature distributions (raw values) of lincRNA and mRNA data using non-parametric test.
- determine significant values by fitting each feature to a statistical distribution and calculate significance thresholds (automatic process $\leadsto$ BGM).
- use the previously determined significance value as a cutoff to divide significant values from insignificant values.
- generate a tool to automatically visualize significant feature regions of single lincRNA sequences.
- overcome problems due to extremely large dataset size (visualization, fitting)

The determination of the significance of the sequences should be simple for the mRNA data, because of the large dataset size; lincRNA, however are far fewer than mRNA in a single organism, therefore it is more difficult to determine the background distribution of their features and therefore to obtain a correct significance cutoff.
Instead of using a cutoff determined from only the lincRNA data, we want to investigate whether it is possible to use the cutoff determined from the mRNA data.
In the process, we gain an additional test of the similarities between lincRNA and mRNA feature values.

Thesis structure

This chapter introduces the topics of the thesis and gives a clear idea of the aim and prerequisites that motivated this work.

In chapter 2, a review of the necessary biological concepts is given, additionally the biological-related computational methods that are required to understand this thesis.

Chapter 3 explains the programs, tools and the arithmetic concepts in probability theory. The chapter highlights the underlying computations and what influence the results mostly regardless of the biological use.

The pipeline of this work is explicitly described in chapter 4, and its associated implemented programs are briefly described in chapter 5.

Last but not least, the results are given in chapter 6 and are discussed in chapter 7. Finally this work is concluded in chapter 8.

CHAPTER 2

Biological and computational review

Protein expression starts with **transcribing** the DNA to messenger RNA (mRNA). The result of transcription is the RNA that shares the same sequence of nucleotides as the DNA with a some chemical modifications. DNA and RNA structures, however, differ greatly from each other, albeit having similar molecular properties. DNA is double stranded which limits the possible structures DNA can form. While RNA is single stranded which gives it more flexibility to fold inside the cell producing an enormous number of possible structures.

After transcription pre-mRNA is spliced to its mature form mRNA and subsequently the protein is read from the mRNA in the **translation** process [1, 14].

Non-coding gene expression is similar to that of protein coding genes, but its missing the translation step.

First the non-coding gene is transcribed and then processed into its mature function form. lncRNAs even use the same splicing machinery as mRNA for their processing and regulation.

The processes transcription, translation, and splicing are highly regulated and errors at any stage may have negative or even deadly consequences [15].

2.1. RNA molecule

DNA sequences are made of the four nucleotides components namely adenine (A), guanine (G), cytosine (C) and thymine (T). Likewise, RNA contains the same first three components A,G,C, but uracil (U) instead of thymine. Moreover, RNA nucleotides differ from DNA because they contain sugar ribose. Hence, DNA is less reactive than RNA, as the nucleotides of DNA are deoxyribose. Additionally, due to the fact that DNA is double stranded while RNA is single stranded, the smaller size of grooves in DNA with respect to RNA, makes RNA easier to break down and less stable than DNA and therefore more exposed to damaging enzymes [1].

2.2. Precursor messenger RNA (pre-mRNA)

After transcription, the coding RNA consists of exons (coding regions), introns (non-coding regions). This initial transcript is called precursor messenger RNA (pre-mRNA) and exists for very short time before it is spliced. RNA splicing generally requires the spliceosome for processing the pre-mRNA into mature mRNA. Spliceosome contains ∼200 proteins and 5 special ncRNAs called small nuclear RNAs (U1, U2, U4, U5, U6) [5, 1, 16]. Splicing is performed by recognizing the exons and introns and then removing the introns later stitching exons back together to form messenger RNA (mRNA) for later protein coding[15].

In humans, the introns are usually far longer than the exons and their properties differ from those in the untranslated regions and the coding sequences [15, 17]. Some introns are even self splicing, i.e. they do not require spliceosome. Research has shown that about 95% of the human genome is alternatively spliced, which implies that the alternative splicing of introns and exons play an extensive role in gene regulation [17, 5, 18]. The relationship between exons and introns also vary between different spicies, in human specifically the length of the intron with respect to the exon choice can affect the alternative splicing [18].

2.3. Messenger RNA (mRNA)

Messenger RNA (mRNA) is a single strand of nucleotides, which has been transcribed from DNA and encoded the information for protein assembly. These nucleotides A,C,G and U are grouped into sets each of three nucleotides, called codons. There are 64 combinations (codons) in total and each corresponds to one amino acid in the translation process.
Out of the 64 codons, four have a special task: AUG initiates translation, while UAA, UAG or UGA act as stopping signals (Fig. 2.1).

tRNA and rRNA are essential for the **translation**, in which, tRNA binds the matching amino acids with the ribosomes. This binding process elongate the chain of the produced peptide chain.
Before tRNA can proceed with mRNA, rRNA customizes mRNA first; this customization

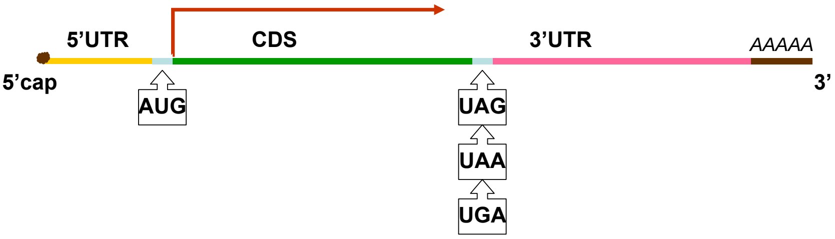


Figure 2.1.: Structure of mRNA. The 5' and 3' are the untranslated regions in yellow and pink respectively, and the coding sequences (CDS) is in green; blue areas are the codons which start (left) or stop (right) translation; the red arrow indicates where translation starts and its direction. Brown areas are the 5' cap on the left side and the poly(A) tail on the right side. RNA sequences are always read and written from the 5'(prime) end to the 3' end.

is literally sandwiching the mRNA, i.e. the large subunit is the upper part, and the small subunit is the lower part (Fig. 2.2).

Coding sequences (CDS)

Coding sequences (CDS) are defined as the regions between the translation start codon AUG and one of the stopping codons (UAA,UGA,UAG). This makes CDS an open reading frame (ORF), which can be decode by the ribosome machinery.
The main function of CDS is protein coding. With the help of ribosomes, the codons in the coding region make base pairing with complementary pairs in transfer RNA (tRNA), once ribosome reads the starting codon AUG. Every time a hit happens and the tRNA can pair with the mRNA a new amino acid is added to the polypeptide chain, which is a single chain of amino acids. This makes up a protein molecule either alone or a combination with other polypeptides [1]. Ribosome keeps reading the mRNA until a stopping codon is reached (UAG in Fig. 2.2), then the resulted chain is released.

Untranslated regions (UTR)

Untranslated regions (UTRs) appear in two parts of the mRNA, the starting (5' end) and ending (3' end). The untranslated region, which start from the first nucleotide of the transcript until the first AUG codon is called the 5'UTR, whereas the 3'UTR is located between the first stop codon (UGA, UAG or UAA) and the poly(A) tail.

Previously, untranslated regions were thought to have no effect on the translation process. More recently, however, research has detected a vast number of regulatory elements in these regions that affect translation.

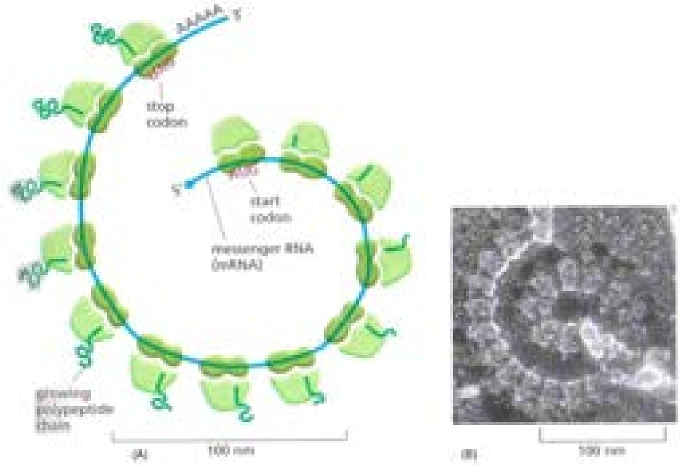


Figure 2.2.: Translation of the mRNA [1]. **(A)** Green parts are the ribosomes translating the mRNA molecule, each starts with the AUG and ends with the UGA, UAG or UAA. The pigtails are polypeptides. **(B)** An electron microscopic photograph taken from an eucaryotic cell.

UTRs 5' and 3' share some properties, such as being highly structured, richness in G and C nucleotides, controlling gene expression regulation via post-transcriptional regulation, plus other features [19, 20, 21]. This involvement drew the attention to their importance in protein expression, as mutations in these UTRs causes abnormality in the whole translation process and results in diseases instead of protein.

The length of 5'UTR varies between taxonomies; some references, however, give a range of 170 $\sim$ 210 nt on average [19, 20]. 5'UTR and 3'UTR differ in length, GC-richness in each, some properties of introns, and some other features [20, 17].

The cis-regulatory elements in 5' and 3' UTRs can influence the stability of mRNA and therefore the translation. As the cis-regulatory elements in 5'UTR can vary from those in 3'UTR, their influence vary as well. For instance mutations that affect these elements in 5'UTR can be associated to breast cancer, alzheimer's disease and other diseases. Similarly to 5'UTR, modification in the secondary structure of 3'UTRs can cause diseases (the relation is still hazy), and there are other drawbacks for affecting the termination codon or other elements in 3'UTR which influence negatively stability and localization of mRNA and the translation process [20, 19, 22].

2.4. Non-coding RNA (ncRNA)

Only two percent of the entire human genome encodes for protein. The last decade, however, has given rise to the characterization of many classes of regulatory RNA, which are transcribed from the remaining 98%.

NcRNAs are found to regulate gene genes expression and mRNA splicing, and there is evidence for their involvement in various diseases[23]. NcRNA classes vary in length and in their properties which sometime make them distinguishable. Hence there are many types of ncRNAs now recognized and well known, such as microRNA (19 - 25 nt), tRNA ($\sim$ 80 nt), ribosomal RNA (rRNA: large subunit [60S eukaryotes, 50S prokaryotic], small subunit [40S eukaryotes, 30S prokaryotic]) [23]. However, it is still tricky to identify a vast number of ncRNAs and in particular their functions.

In addition to gene expression regulations, some ncRNAs are so called housekeeping ncRNAs, e.g. tRNA, and rRNA [24], due to the fact that they are produced in all types of cells, and are involved in ubiquitous translation process.
Another ncRNA class involved in gene expression is microRNA. MicroRNAs bind to 3'UTR of the mRNA and hinder the translation either by degradation of the transcript or inhabitation of the ribosome machinery. This exact regulation found to have the affect on the cell function in a good or a malicious way.

2.4.1. Long ncRNA (lncRNA)

Long ncRNA (lncRNA) is a sub-class of ncRNA, which are longer than 200 nt in length and generally do not contain sequence homology. lncRNAs regulate gene expression via *chromatin modification, transcriptional regulation, and post transcriptional regulation*.

In [25], five sub categories of lncRNA were mentioned, namely: (1) antisense transcripts, (2) intron re-presenters, (3) UTR promoter transcripts, (4) independent transcripts, and (5) non protein-coding genes transcripts. This classification was based on the loci in the genome, considering that genomes vary among taxonomies.

Interestingly, some previously unclassified lncRNAs, which share nothing in common with the rest of lncRNA sequences, have been called as lincRNA, as lincRNAs are strongly assumed to have roles in many human diseases [11].

2.5. RNA structure

RNA is a single strand of nucleotides, these nucleotides chemically bond with their complementary pairs to form a base-pair. According to the Watson-Crick model [26], the complementary pairs are A with U, G and C, also G can pair with U in RNA structures.

Inside the cell, when RNAs fold into their native structure, complementary nucleotides chemically bond to each other. Such a bond might be a hydrogen bond in which a hydrogen atom interacts with an electronegative atom. The result of this bonding is

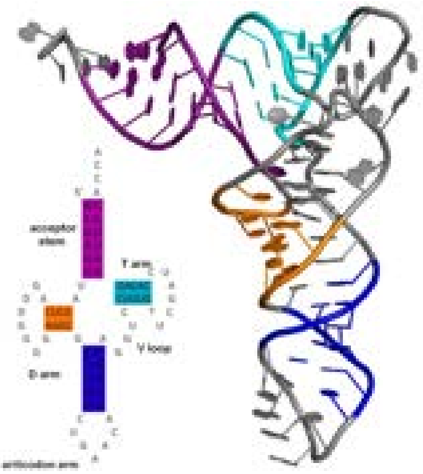


Figure 2.3.: RNA structure [http://en.wikipedia.org/wiki/Biomolecular_structure]. **Left** The secondary structure, in which the original sequence can be seen starting from 5' to 3'; loops are the unpaired bases. **Right** The tertiary structure of RNA molecule. The colorful regions are those in which bases are paired and gray areas represent the unpaired bases. The nucleotide sequence of tRNA is GCGAUU-UAGCUCAGDDGGGAGGCGCCAGACUAAACAUCUGGAGGUC-CUGUGTUCGAUCCACAGAAUUGCACCA.

structured RNA. Moreover, the total number of possible structures for RNA sequence grows exponentially with its length [27].

RNA folding might produce stable structures or unstable ones. Taking in account how many Cs, Gs, As and Us in the sequence and their positions, the resulted structure therefore is sort of predictable. Moreover, the produced structure is more unstable when more bases are left unpaired.

Three levels of RNA structure exist: (1) primary structure, (2) secondary structure, and (3) tertiary structure (Fig. 2.3).
RNA primary structure is the sequence of nucleotides that make up the RNA strand. This sequence of nucleotides holds information of homology and RNA specific features. Predicting RNA tertiary structure (3D structure) is very complex and not possible for long RNAs. However, RNA secondary structure is kind of well established approach and its requirements are within reach. Therefore, this work concentrates only on secondary structure as the primary structure is very difficult to predict and impossible to do so for long transcript such as mRNA and lncRNA.

Next, some RNA secondary structure terminology and its prediction are briefly introduced.

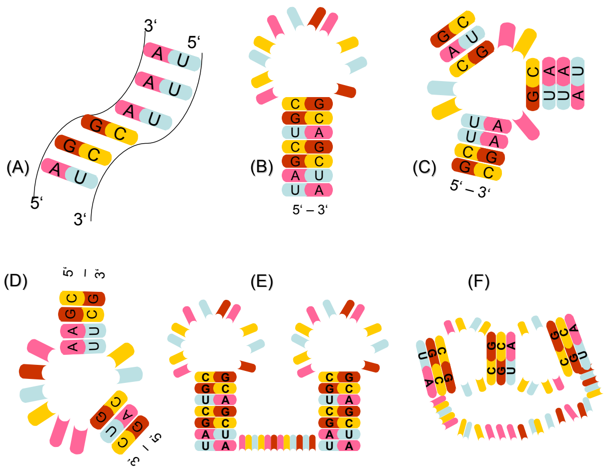


Figure 2.4.: Building blocks of RNA secondary structure. **(A)** Stem, or stacking where complementary bases from different parts of the RNA strand pair with each other. Stems are a stack of at least two base-pairs. **(B)** Hairpin, a loop closed by one base-pair (here CG). **(C)** Multiple loop, an interior loop closed by at least 3 closed base-pairs (UA,CG,GC), the number of stems branching off defines the size of the multiple loop, i.e. this is a 3- multiple loop. **(D)** Interior loop, where some nucleotides are unpaired and are flanked by two base-pairs (UA,CG). **(E)** External loop is the stretch of unpaired nucleotides between two closed structures, such as the two haipin structures in this illustration **(F)** Pseudo-knots the edges between the two hairpins .

RNA secondary structure (SS)

RNA secondary structure is built from two main building blocks : loops and stems. Loops (*hairpin*, *interior loop*, and *k-multiple loop*) are free unpaired bases that are grouped together and named according to their position in RNA SS. Another type of loop is the *external loops* which are the un-paired bases and the base-pairs with no closing base pairs. Stems are stacks of neighboring base-pairs. RNA structure can consist of base-pairs, which their edges cross with other base-pairs, these crossing base-pairs are the pseudo-knots (Fig. 2.4).

2.6. RNA structure prediction

Predicting RNA SS is finding the most likely structure among all possible structures for the given sequence. The early predicting methods focused on sequence conservation among several homologous sequences. However, choosing the sequences and considering the comparative mechanism influence the required time and make this approach a time

consuming one.
Later on, the interest in the possible structures of one single sequence became the main investigated area. For this purpose using **Gibbs Free Energy** definition of a system became nowadays more established and precise.
On the other hand, every single sequence can have an exponential number with its length of possible structures. To find the one with the minimum free energy (MFE), all possible structures of a sequence are required to be explored.

2.6.1. RNA structure and minimum free energy (MFE)

Gibbs Free Energy system is given by $\Delta G = \Delta H - T\Delta S$, where ΔH is the enthalpy, T is the absolute temperature, and ΔS is the entropy. Hence, the lower the energy of a system, the more stable the structure [28].
Predicting RNA SS is now finding the structure from all possible structures comprising non-crossing base-pairs that have the minimum free energy.
The overall energy of the system is computed through the energy of its base-pairs and loops. Loops usually have a high energy contribution due to the fact that bases are unpaired, while stems are the stable parts of the structure (2.1).

$$E_{loop} = E_{mismatch} + E_{size} + E_{special} \tag{2.1}$$

Where $E_{mismatch}$ is the energy of the neighboring base-pairs to the loop closing base-pairs, E_{size} the energy of the size of the loop, and the last term $E_{special}$ considers special cases such as tetiary loops [29].

2.6.2. RNA structure and dynamic programming

It is time consuming to try generating all possible structures, therefore dynamic programming (DP) approaches became useful, as it is based on exploring all possible structures without actually generating them. DP has two phases: scores computation, and optimal solution found from trace backing [27].
The standards algorithms based on this concept are:

1. **Nussinov's algorithm** $\rightsquigarrow$ **1970**: Nussinov's algorithm is one of the first algorithms, that predict RNA SS. The algorithm maximizes the number of base-pairs which can be found in an RNA sequence S of length L; the sequence is compared against it self in a matrix of size $L * L$, which makes the required space of this algorithm $O(L^2)$, and the running time $O(L^3)$[28]. With Nussinov all optimal and suboptimal structures are produced, however, the predicted structure could be biologically irrelevant (no loops penalty).

2. **Zuker's algorithm $\rightsquigarrow$ 1981**: Zuker's algorithm predicts one structure, which is found to have the minimum free energy ΔG [28]. Making the advantage of DP, Zucker's algorithm predicts the optimal structure using two matrices, one stores the MFE of subsequence $s[i \dots j]$, and the other stores the MFE of the structure representing that subsequence $s[i \dots j]$ (inclusive) [5, 30]. The complexity of Zuker algorithm is therefore $O(L^2)$ for space, and $O(L^3)$ for running time. The drawback of Zuker is predicting unique optimal structure and no information about other suboptimal structures, that might occur in the cell [30].

3. **McCaskill algorithm $\rightsquigarrow$ 1990**: McCaskill predicts the optimal structure using a partition function for base-pairing probability. McCaskill considers the probability of the structures, that RNA sequence might fold to, based on the distribution of the system energy, which is assumed to be a Boltzmann distribution [31].
   In the first phase "scores computation" the partition functions are computed for all sequence components, in the "trace backing" phase the probabilities of base-pairs are determined [27, 31, 32]. The complexity of McCaskill is similar to Zuker.

CHAPTER 3

Scientific background

In this chapter, tools and concepts used in this work for the analysis of different features are introduced. First, the underlaying methods for the analysis of RNA SS are defined. After that, the statistical concepts, tests, tools and methods are explained.

3.1. RNA structure prediction programs

There are several approaches for the prediction of RNA SS [`http://en.wikipedia.org/wiki/List_of_RNA_structure_prediction_software`]. In this work two programs are used, namely RNALfold and RNAMotid.

3.1.1. Local RNA secondary structures (RNALfold)

The Vienna RNA package provides an algorithm for predicting local stable structures in RNA sequences called *RNALfold*. The algorithm uses a simpler alternative to the full loop based energy model called maximum circular matching problem (MCMP) which is a model based on base-pair strength based model (3.1). Using McCaskill partition function, in a sequence S of length n, the probability of base-pairs is computed, and the list of the local stable structures is produced using a forward recursion within a span $L \mid L < n$ [33].

$$E_{ij} = min \left\{ E_{i,j-1}, \min_{\substack{k=i...j-m \\ \Pi_{kj=1}}} E_{i,k-1} + E_{k+1,j-1} + \epsilon(k,j) \right\} \tag{3.1}$$

where E_{ij} is the energy of the most stable structure within the span i,j (i,j are included).

RNALfold returns the predicted secondary structures, the minimum free energy (MFE) of each predicted structure and its starting position in the original sequence.

The predicted secondary structures might overlap, i.e. in sequence S $\{p_1...p_i...p_n\}$, the position $\{p_i\}$ can be involved in more than one structure within a span of size L, often with different scores $p_i = \{sc_1...sc_j...sc_m\}$ where m is the total number of the overlapping scores.

The ability of controlling the size of the sliding span L gives this work the flexibility of comparing different features produced by different tools by setting a constant span size for all of them.

The time complexity of the program is $O(n \times L^2)$, and the space complexity is $O(L+n)$.

3.1.2. Local RNA elements accuracy based identification (RNAMotid)

RNAMotid is a tool for the prediction of local RNA structure provided from University Freiburg [34]. Two concepts the program maintains namely the maximum expected accuracy (MEA) for RNA subsequences, and the probability of base-pairing and un-pairing.

MEA model predicts the structure with the highest probability. The drawback of this model is that short local structures are predicted more often than long ones. Therefor RNAMotid introduced an additional locality function (3.2), so that the MEA is alternated for RNA subsequences and hence, the program is capable of predicting small and large RNA elements (e.g. cis-acting elements).

The locality function introduced (3.2) has the advantage of being flexible and having a changeable behavior based on the input parameters, which in this case can handle different classes of RNA considering their different properties.

$$Accuracy = \frac{Pr[base\_pair] + Pr[base\_unpair]}{start\_value + subsequence\_length^{degression}} \tag{3.2}$$

For the probability of the different base pairs and un-pairs, the program obtains these probability from `pfl_fold` and `RNAplfold`, the two functions are included in Vienna package. However, the restrictions upon windows length in `pfl_fold` was overcome by a successive calling to the function, paying attention at the same time to the overlapping sequence [34].

The time complexity of the tool is $O(nm^2)$, where n is the sequence length, and m is the length of the chunk to `pfl_fold`. Finally the space complexity is $O(n^2)$

3.2. Sequence shuffling (esl-shuffle)

Shuffling a sequence is a technique that creates random sequence with a restriction of maintaining the different nucleotide frequencies. There are different types of shuffling depending on the nature of the sequence such as mono-nucleotide shuffling and di-nucleotide shuffling [35].

Sequence shuffling is one step among other steps to examine the stability/conservation of the molecule structure [36, 37, 38]. In relation to RNA, once the sequence was shuffled, the MFE of its structure is compared to the MFE of the structure predicted from original sequence.

In this work Mono-nucleotide shuffling is used (single-base shuffling) using esl-shuffle program, which is provided from HMMER project from Howard Hughes Medical Institute. The shuffling tool, plus others in HMMER project, is based on profile hidden Markov models (profile HMMs) [39].

3.3. Distributions

3.3.1. Normal distribution

Normal distribution or gaussian distribution is one of the most important concepts in continuous probability distribution [40], which is given by the following equation:

$$Y = \frac{1}{\sigma\sqrt{2\pi}}\ e^{\frac{-1(x-\mu)^2}{2\sigma^2}} \tag{3.3}$$

where μ is the mean of the distribution, σ is the standard deviation, and $e = 2.71828$. Normal distribution $\aleph(\mu, \sigma^2)$ is famous with several properties such as the bell shape (Fig. 3.1) and being analytically very well established. Moreover, the mathematical background of normal distribution is the central limit theorem which states sampling distribution of independent random variables is approximately normal [40, 41, 5].

The mean of normal distribution is also its median and mode, while σ^2 is the variance of it, i.e. how dens the values of the distribution around its mean.

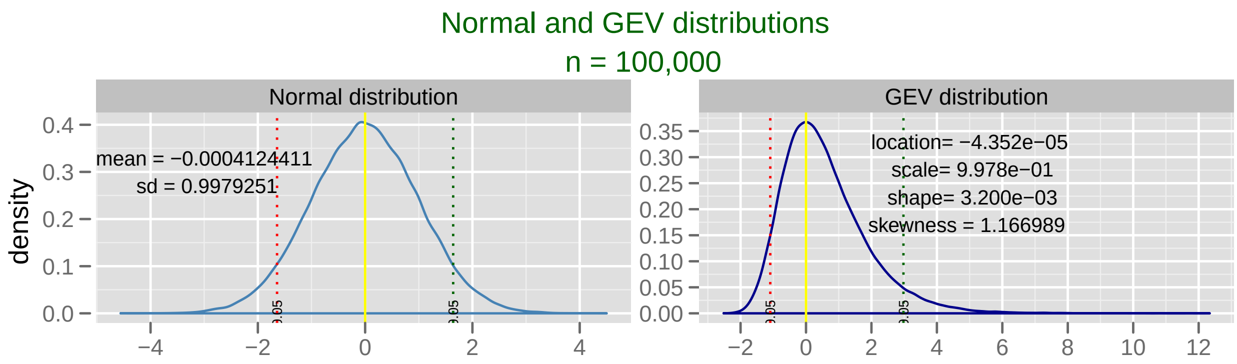


Figure 3.1.: Normal distribution and GEV distribution and their parameters. Green line represents the upper significant 0.05, red is the lower in both plots, and yellow line indicates the mean(normal distribution) or location (GEV). As GEV is right tailed the value of skewness is positive.

Z-score

Standard scores (z-scores) are the score of the deviation from the mean measured in standard deviation units (3.4). Z-scores are very important in a sense of comparing the distributions [40, 41]. Additionally, The distribution of z-score is always normal $\aleph(0, 1)$.

$$Z = \frac{X - \mu}{\sigma} \tag{3.4}$$

where X is the value of which the z-score will be computed.

3.3.2. Extreme value distribution (EVD)

Extreme value distribution or the extreme value theorem describes the cases of extreme events like an extreme earth quick or flood.
EVD has three types : Gumbel distribution, Fréchet distribution, and Weibull distribution; each describes a different case.
Generalized extreme value distribution (GEV) is the general form of the three EVD types which is given by (3.5)[42].

$$f(x) = \begin{cases} \frac{1}{\delta} \quad exp(-(1+\varsigma z)^{\frac{-1}{\varsigma}})(1+\varsigma z)^{-1-\frac{1}{\varsigma}} & \varsigma \neq 0 \\ \frac{1}{\delta} \quad exp(-z - exp(-z)) & \varsigma = 0 \end{cases} \tag{3.5}$$

where ς, δ are shape and scale of the distribution, and $z = \dfrac{x - \eta}{\delta}$, η is the location.
Likewise, GEV ($\wp(\eta, \delta, \varsigma)$) has several properties, which is famous with such as its recognizable tail (Fig. 3.1), and the widely usage in finance, economics, etc.

Gumbel distribution

Gumbel distribution is used to model the extreme of samples despite the type of the underlying distributions [5]. It is possible that the maximum of a normal distributed sample can be modeled as Gumbel distribution as well [43], as the maximum of normal distribution and other types of distributions converges to Gumbel distribution.

Gumbel is usually skewed to the left and its probability density function depends on scale and location only (3.6)[42, 43]:

$$f(x) = \frac{1}{\delta} \quad exp(-z - exp(-z)) \tag{3.6}$$

the mean μ of Gumbel distribution and its standard deviation σ can be computed via (3.7) [5]:

$$\begin{aligned} \mu &= \eta + \gamma\delta \\ \sigma &= \frac{\delta\pi}{\sqrt{6}} \end{aligned} \tag{3.7}$$

where γ is Euler-constant = 0.5772156649015328606.

3.3.3. Empirical p-value distribution (EMP)

The empirical distribution is the distribution of the empirical p-values, which is the ratio of a chosen value to the total number of the set [40, 5].

The empirical p.value is an established approach for describing the data without assumption. Hence, the distribution of empirical p.values can address any random variables and therefore through EMP one can make assumptions on the data even if the assumptions were statistically independent, e.g. assuming that a sequence is significantly low expressed, and it has regions with uncorrelated structures.

3.4. Definitions

Type I and Type II errors

Accepting the null hypothesis H_0 or rejecting it, is not a trivial decision and not error free. There are two types of error can be made namely the type I and type II.

If H_0 was rejected when it should be accepted, then the type I error has been made; if H_0 was accepted when it should be rejected, then the second type has been made.

Either accepting or rejecting the H_0 the probability of making one of these types should be minimized, however, there is no way to escape one of them without falling in the other [40].

Although the probability of making errors always exist, one can limit it by increasing the sample size. In this work, the total amount of sequences is over 21,000, which increases the chances of making a good decision.

Significance levels (α)

The definition of a significance level (α) is the maximum probability at which, type I error might be made, i.e. rejecting the H_0 at point x knowing that there is a probability α the H_0 should be accepted [40].

There are six standard levels of significance for one-tailed test [41] namely: *0.10,0.05,0.025,0.01,0.005,0.0005*. It makes a difference choosing the levels of significance for one or two tailed test, as for two-tailed test the levels of significance would be *0.20,0.10,0.05,0.02,0.01,0.001* .

In this work the decision was made for the upper/right one-tailed test R_α and lower/left L_α, hence any distribution has left and right tail despite the inequality between them.

To determine the significant of x at α the decision was made via x can be either significant (maximally (3.8), minimally (3.9)), or not significant; so the null hypothesis states the position of value x (p_x) in a sequence is significant with probability=1.

$$Pr[p_x] = \begin{cases} 1 & x > R_\alpha \\ 0 & x < R_\alpha \end{cases} \tag{3.8}$$

$$Pr[p_x] = \begin{cases} 1 & x < L_\alpha \\ 0 & x > L_\alpha \end{cases} \tag{3.9}$$

Law of large numbers

This law was first proved by the Swiss mathematician James Bernoulli in 1713, however, it was first propossed by the Italian mathematician Gerolamo Cardano in 15xx. According to this theorem the average of sampling the data for large number of trials is close to the expected value [44, 5].

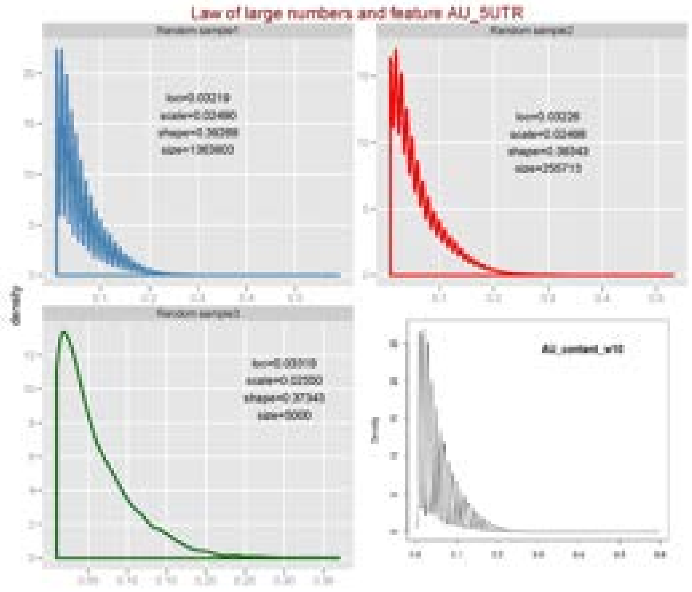


Figure 3.2.: Lower-right the original feature with n = 4091409, location = 0.03223, scale = 0.02492, and shape = 0.39164. A visual example of the concept of law of large number, where a random sample doesn't differ much from the actual/expected distribution.

The expected value of a variable X is defined as the sum over its value (x_i) time its probability ($pr[x_i]$) (3.10)[40]. Law of large numbers accordingly can be written as in (3.11), and $\bar{X} \longrightarrow E[X]$.

$$E[X] = \sum_{i=1}^{n} pr[x_i].x_i \tag{3.10}$$

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{3.11}$$

In Fig. 3.2 random samples each of different size, were taken from one of the features generated in this work (AU_content_w100_5UTR), and the overall distribution of the whole feature. The largest sample is about 30% of the feature total size, however, the parameters of the distribution do not differ much; the same can be said for the smallest sample (less than 1%).

Sum of independent variables

This method addresses the problem of summing up independent sets and the result of that sum. If X and Y are two independent **distributions** of the same type, then X+Y is also a distribution from the same type. This method been applied and proved for different data types either discrete or continuous [44, 5] and with different cases.

Proving this theory was done on many levels, such as geometrically, distributions functions comparison or distributions parameters comparison [44], while in [5] the theorem was given and proven only for the case of normal distribution.

3.5. Statistical tests

3.5.1. Kolmogorov-Smirnov test (KS)

Kolmogorov-Smirnov test is a nonparametric test, that can be applied to almost all known distributions, such as normal distribution, extreme value distribution, gamma distribution, etc. Nonparametric tests don't require pre knowledge of the data, or any additional parameters, and this is one of the main reasons for using KS-test here.

There are two types of KS-test, the two-samples KS test, and the one sample KS test; both are applied in this work. In general, KS test reports the statistical maximum difference (D) between the given parameters, and a p-value [40, 45, 46][1].

The two-samples KS tests whether an independent unknown sample x is equal to an independent known sample y, in a sense they have the same type of distribution. This is computed by comparing the accumulative distribution functions of both samples and then measuring the maximal difference between them. Hence, the smaller D is, the better the chance is, both distributions are identical.

In case of one-sample test the result of the test D is compared to the alpha level α (level of significance), if $(D > \alpha)$ then the null hypothesis is rejected.

3.5.2. Wilcoxon-Rank-Sum test

Wilcoxon-Rank-Sum test or Mann-Whitney is a nonparametric test for two independent samples n_1, n_2; the null hypothesis of the test H_0 is the expected value of n_1, n_2 equal to each other $\eta_1 = \eta_2$ despite their statistical origin [47, 40].

Wilcoxon tests the two samples (n_1, n_2) by ranking their values (R_1, R_2) which might influence the final result of the test, as some of the values can be identical (ties) or equal to zero. These two cases make computing p-values for wilcoxon more complicated. Solving the zero values problem is usually done by excluding them. The final decision

[1] Internet is full with resources!

is computed as $U = min(U_1, U_2)$ (3.12), and (3.13) [47, 40]; the smaller U is, the more likely to reject H_0 and accept the alternative hypothesis instead.

To overcome the ties, some software use the standard method (summing the ranks up), while some others used the approximate method.
R uses the approximation method, which mean these sum of ranks will be converted to z-score and the p-value will be seek from normal distribution.

$$U_1 = n_1.n_2 + \frac{n_1(n_1+1)}{2} - R_1 \tag{3.12}$$

$$U_2 = n_1.n_2 + \frac{n_2(n_2+1)}{2} - R_2 \tag{3.13}$$

3.6. R - tools and methods

This work is based on statistical analyzes, therefore R is used especially for its ability in handling large data set in remarkable time.
R is a statistical computing and graphics environment and was developed to handle the data and visually producing it. Plus being statistical tool R can be easily extended to serve other purposes like geometric problems or serving supplementary programs (Matlab, Mathematica), this extension is done by adding new packages [48, 49].

In this work, the following packages/libraries and the functions were used:

1. library `evd`: this library has the required functions for the generalized extreme value distribution (GEV)

   a) `fgev`: fitting the generalized extreme value distribution using maximum-likelihood estimation (MLE). MLE is very robust, efficient, and is modeled basically for non-linear regression. Such accuracy, however, relays on maximizing the likelihood function which has in return some conditions such as the shape of log-likelihood function should be convex [50]. Practically to speak, once the shape of the estimated parameter is less than -0.5, R fails to fit the given sample to GEV.

   b) `rgev`: creates a random GEV according to the given parameters[1].

[1] The help of R can provide more details on the default parameters.

c) `qgev`: finds the quantile function [5] for the given value (x_p). If the parameter `lower.tail` was given the value `False`, then the function should find the closest actual value on the distribution (x_d) that is above the given value $x_d > x_p$, otherwise, the returned value is $x_d \leq x_p$, i.e. upper threshold or lower threshold.

2. library `stats`:

   a) `rnorm`: creates a random normal distribution according to the given parameters. Default parameters are $\mu = 0, \quad sd = 1$.

   b) `qnorm`: same as `qgev` but for normal distribution.

   c) `quantile`: a generic function implemented in R, and computes the quantile function using nine different methods categorized into two catogeries: discontinuous sample quantile (type 1-3), and continuous sample quantile (type 4-9). Although R'default type is 4, and the recommended type from Hyndman and Fan [51] is 8, the type chosen in this work is 1.

   d) `ks.test`: Kolmogorov-Smirnov test.

   e) `wilcox.test`: Wilcoxon-Rank-Sum test.

3. library `timeDate`:

   a) `skewness`: to computes the skewness of the distribution, and accordingly if the result was positive, then the distribution is right-tailed one or the other way around. This information is important for non normal distributions for the fact that every distribution has two tails, but which one is bigger.

4. library `ggplot2`: this library combines the strengths of base and lattice graphics and tries to avoid the weaknesses of them [52]. In this work, ggplot2 failed to create multiple-layers plots for large input (each file has number of entries which consist of 7 digits on average), but this problem was solved (see Chap. 4).

5. other R functions:

   a) `sample`: an implemented function in R which takes random samples from a given sample, however, the only mathematical detail given about randomness mechanism, is the probability of choosing the elements in case of no replication is proportional to the weight of the rest items.

   b) `mean,sd,summary,sum,max,min`

CHAPTER 4

Methods

In this chapter, the methods are described of how the local features of RNA sequences were generated, how the statistical background model (BGM) was computed based on features probability distributions. The BGM provides information on the different significance cutoffs for each data set. Thus the significant feature values of individual sequences were determined.
Afterwards, two types of comparison were done: 1) a comparison of the original feature distributions, and 2) the same comparison based on the ratio of significance values per sequences.
The first comparison aims to compare the whole space, and the second comparison determines whether two data sets have the same number of the significant regions.

The following steps were followed to make this investigation:

1. **data sets**: human lincRNA were compared to mRNA and shuffled-lincRNA sequences.

2. **features**: RNA nucleotide content and RNA secondary structure were the generated features. Via these features the data sets were compared. All features were window-based, i.e. this work focused only in RNA local features.

3. **background model (BGM)**: the statistical model automatically determines the significance thresholds of each feature for different RNA sets. This model is capable of determining the suitable distribution of each feature, which make the two comparisons reliable. The BGM consists of the following parts:

   a) distribution determination

   b) distribution fitting

   c) goodness of the fitting

   d) significance levels

4. **significance ratios**: the significance ratios represent the overall significant positions per sequence, this is computed based on the BGM for the whole set.

5. **visualization**: visualizing the raw data of single features and different combinations of features. In this phase the huge amount of data is reduced to focus into regions of interest. Additionally, binary plots are produced which highlight the significance of each position in a sequence.

4.1. Data sets

The sequences used for Background Model (BGM) are mRNA which were downloaded from `http://genome.ucsc.edu/`. Later, the criteria of excluding and keeping which of them are explained in two different sections (Sec. 4.1.3, 4.2.3).

4.1.1. mRNA

All human mRNA were taken from `Refseq Genes -hg19`, and each sequence must have the three regions: coding region and untranslated regions (Chap. 2).

All together, the total number of mRNA sequences used in the following calculations are 21844 (Sec. 4.1.3).

4.1.2. lincRNA

The lincRNA sequences were downloaded from `http://ftp.ensembl.org/`. Those sequences were filtered (see Sec. 4.1.3) so the overall number was reduced to 3021 sequences.
Considering that lincRNA set is a small one, applying the background model (BGM) will not be much of gain in a sense of precision. Moreover, for making a reliable significance comparison, the different cutoffs were obtained from the other BGMs computed in this work namely: $BGM_{5'UTR}, BGM_{3'UTR}, BGM_{CDS}, BGM_{shuffled-lincRNA}$.

Table 4.1.: RNA features

	RNA mono-nucleotide	RNA di-nucleotide	RNA secondary structure
Feature	A, C, G, U	AC, AG, AU, GC, GU, UC	MFE, MEA

4.1.3. Shuffled lincRNA

All lincRNAs were shuffled ($\sim$ 7x) using `esl-shuffle`. By this amount of shuffling, the total sequence number for the shuffled background model $BGM_{shuffled-lincRNA}$ is almost as much as the total sequence number of other BGMs $\{BGM_{5'UTR}, BGM_{3'UTR}, BGM_{CDS}\}$. This approach will provide the knowledge of the null model for lincRNA structure.

Data set filtering

The two sets mRNA and lincRNA were filtered as following:

1. repeated sequences are excluded from the main data set
2. sequences with no UTR regions or CDS are excluded as well. I.e. only sequences with the form [5'UTR - CDS -3'UTR] are considered.
3. sequences with high noise are excluded (4.1).

4.2. Feature generation

Because the generated features are local features, all the featured were categorized according to their origin: the 5'UTR features, 3'UTR features, and CDS features.
In Table 4.1 the features generated in this work are given. By applying the sliding-window mechanism a value will be assigned to each position in each sequence. This value represents the score of that window as will be seen next.

Despite the nature of the feature, the output of this phase is $f(S, P, V) \mid f \in F\{f_1, \ldots, f_m\}$, where S is the sequence ID, P is the position, V is the value and m is the total feature number.

The di-nucleotide contents CA,GA,UA,CG,CU are computed within the previous contents, i.e. if CAs were found in a sequence, then their score will be included within AC and so on.

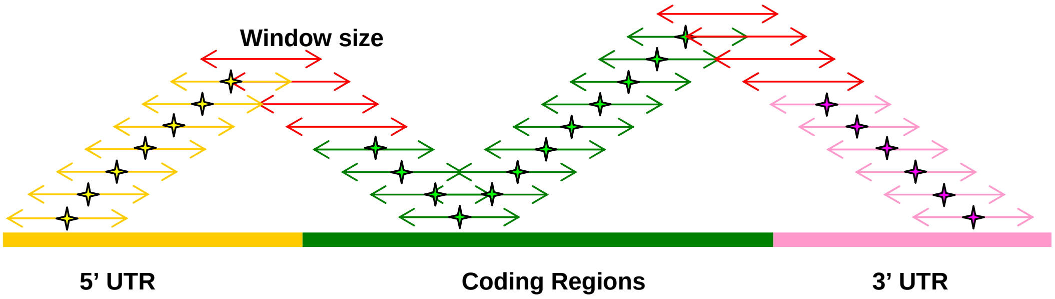


Figure 4.1.: An illustration represents how the compositions of an mRNA sequence been computed, in which: red lines are the overlapping windows between different regions, and the Stars are the window’s center.

4.2.1. RNA sequence compositions

This work analyses the following sequence contents: mono-nucleotide frequencies A, C, G, U, di-nucleotide frequencies AC,AG,AU,GC,GU,UC, and N, Where N indicates any content $x \notin \{A, C, G, U, T\}$ (4.1), i.e. N indicates the noise in the sequence so that it is possible to measure the quality of the sequences[1] (Sec. 4.2.3).

$$\frac{\sum x \notin \{A, C, G, T, U\}}{\mid s_i \mid} \geq 25\% \mid s_i \in S\{s_1 \dots s_n\} \tag{4.1}$$

Where n is the total number of sequences.

In Figure 4.1 a **sliding-window** technic is applied on every sequence by shifting the window one position at a time starting from first position p_1 till the end of the sequence ($S = \{p_1 \dots p_n\}$).

After choosing an arbitrary window length[2] the window slides over the sequence $W = \{p_i \dots p_j\} \mid i, j < n \ \& \ p_i, p_j \in P\{p_1 \dots p_n\}$, then the density of each content (e.g GC) within the range of the window will be computed and assigned to the center of the window (4.2)

$$p_{\frac{w}{2}} = \frac{\sum_i^j (GC_l \dots GC_k) + (CG_l \dots CG_q)}{\mid w \mid} \tag{4.2}$$

[1] In this step, the sequences were lacked from any noise in all positions, i.e. no sequences were deleted for being lower than the allowed limit

[2] In this work $\mid w \mid = 100$.

where $(GC_1 \dots GC_k), (CG_1 \dots CG_q)$ are all the appearance of GC, CG respectivley in that window; the next position will be $p_{\frac{w}{2}+shft}$, where $shft$ is the shifting size[1] (p_{i+shft}, p_{j+shft}).

The first and last $\frac{w}{2}$ positions in each sequence are masked, due to scoring the positions of window's center.

4.2.2. RNA local secondary structure

Minimum free energy (MFE)

The fact that RNALfold reports all overlapping structures, here only dominant ones are kept that are located entirely in one region. Hence the structure with the minimum energy is taking: $P_i = min\{e_1...e_n\}$, where $\{e_1..e_n\}$ are predicted structures energies, and the position P_i is involved in all these structures.
Additionally positions which are not involved in any structure are neglected ($P_i = 0$).

Maximum expected accuracy (MEA)

RNAMotid spots different areas in the sequence, comparing to RNALfold results. I.e. the overlapped predicted local secondary structures are similar in some regions and differ in others to those predicted by RNALfold. Eventually, RNAMotid result is treated the same way, so only high scores are taking among the overlapping structures. $P_i = max\{sc_1...sc_n\}$, where $\{sc_1..sc_n\}$ are predicted local structures scores, and the position P_i is involved in all these structures.
Likewise positions with no scores are neglected, and structure in only "one-region" is kept.

4.2.3. Filtering of features

Putting it all together a filtering has been applied on many levels ; as a matter of course it is a decision of the user which features to generate and how to handle its results.
Filtering the features was conquered by some/all of next filters:

1. when computing any feature, the length of the sequence should be no smaller than the parameters specified for that feature (e.g. L in RNALfold).
2. computing RNA sequence compositions, the overlapping windows among regions are neglected.

[1] In this work $shft = 1$.

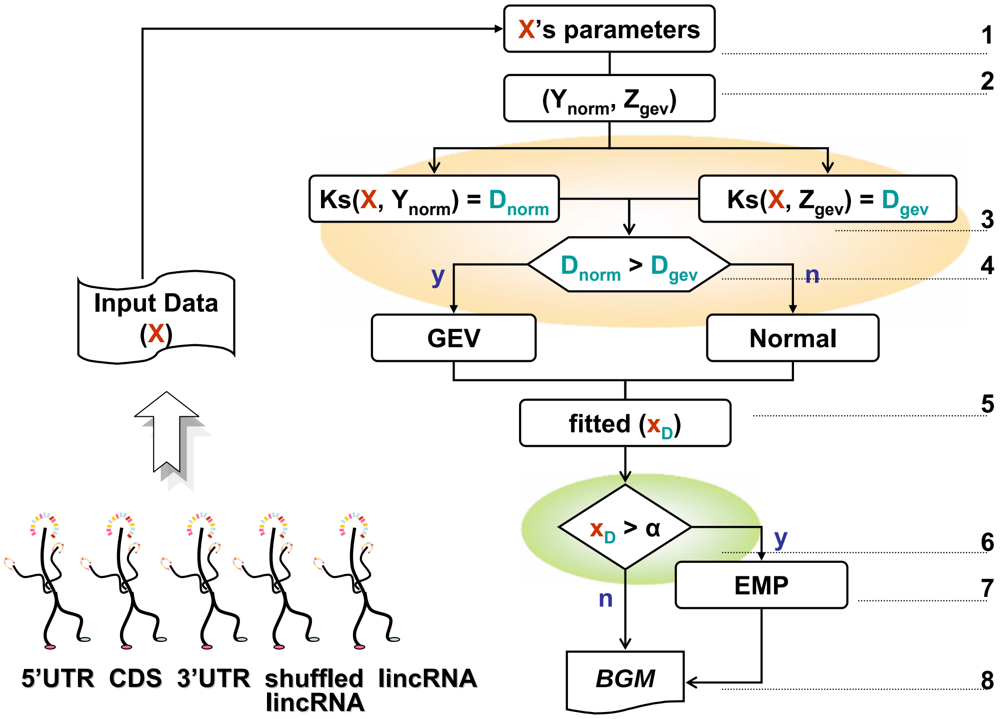


Figure 4.2.: A diagram of BGM where the different phases are shown.

3. computing RNA(Lfold-Motid), the positions with no values are neglected.

4. only human - lincRNA sequences are investigated!

The sequences used for BGMs (3',5' UTRS and CDS), were filtered by all mentioned filters except the last one; lincRNA features were filtered by (1,3,4).

4.3. Background model (BGM)

The background model is a statistical computation model for features analysis. BGM determines the threshold of different features based on their probability distributions.

Such ability of determination makes it possible to identify the regions in a sequence, which have significant scores. Moreover, it allows to analyze different features from different origins searching for a significant pattern they might share.

It can also be seen the other way around, by being able to decide if the sequence statistically have certain information, based on the threshold the background model computed.

4.3.1. Distribution determination

The main idea of BGM is distinguishing between significant sequences and non significant ones. Therefore, BGM should be able to determine the most suitable distribution to each feature, but this determination requires some level of knowledge, which distribution is more likely the data tend to form. Holding such knowledge is not that trivial as it sounds.

An additional property of BGM its flexibility, i.e. the model was designed in a way, such that more distributions can easily included, as will be shown later.
Finally, feature F will be described as normally distributed, if normal distribution fitting was better than GEV or EMP to the distribution it forms. With the help of *Kolmogorov-Smirnov test* for two-samples, such decision can be made based on the result of the test D, the maximum statistical difference. In Figures 4.2 the main steps to compute BGM of data sets are highlighted.

Parameters of the distribution

To address the behavior of every feature $F(S, P, V)$, the values of sequences positions will be examen. First the values will be assumed to be normal distributed, then the mean and the standard deviation of these values will be computed. Second the values will be assumed to be Generalized extreme value distributed (Sec. 3.3), then location, scale and shape will be computed (Fig. 4.2.1).

Kolmogorov- Smirnov two samples test

Based on the parameters computed from the previous step, two independent samples are generated (Y_{norm}, Z_{gev} (Fig. 4.2.2)). In Fig. 4.2.3 these two samples are tested against the original values X using KS test.

The two statistical differences (D_{normal}, D_{gev}) coming from testing original data set X against a surely normal sample Y, and testing original data set X against a surely gev sample Z will be compared (Fig. 4.2.4), the smaller one indicates that X is more similar to this sample than it is to the other one.

Due to this simple mechanism, it is easy to extend the BGM and introduce new distributions to it, as long as the same concept is followed.

4.3.2. Distribution fitting

Fitting each features distribution means finding the corresponding parameters of the distribution, which found to be the best representing this feature. Additionally computing the threshold of the features at different significance levels.

The parameters of normal distribution are : *mean, and standard deviation*; in this model the corresponding **z-score** for each significance level is computed as well (Fig. 4.2.5). GEV has different set of parameters: *location, scale, and shape*. Knowing that the tail of GEV is its fingerprint, the tail is indicated with letter *r,l* for right tailed GEV, left tailed GEV respectively.

Once the decision was made, which distribution the values tend to form, computing its threshold is quite straight forward.
Since every distribution has two tails despite their equivalence, the model computes both tails for the three mentioned distributions.

fgev and GEV

Previously, fgev fails to fit the sample to GEV if the estimated shape was less than -0.5 (Sec. 3.6). This was handled in the BGM in a yes-no manner, i.e. if the function fails then the BGM assumes this sample is definitely not a GEV, and fits it directly to normal distribution.
The results (Sec. 6), however, showed that this is not always the case, and fgev fails to fit GEV although that the sample is a definite GEV, as the case of MFE distributions for different data sets for instance (Fig. 6.3).

4.3.3. Significance levels

The background model computes one tail at a time, since both tails are important, i.e. there will be twelve entries representing lower and upper features thresholds.

4.3.4. Goodness of the fitting

Due to the fact that features are fitted to pre-chosen distributions, an additional test must be applied, to check the goodness of the fitting. For this purpose the one-sample *Kolmogorov-Smirnov test* is used (Fig. 4.2.6). If ($D > \alpha$) then the choice of the distribution was not good enough and the feature should be recomputed and fitted as an EMP (Fig. 4.2.7), otherwise, the fitting was successful.

4.3.5. Empirical p-value distribution fitting

This should be the failing state, therefore, it is highly recommended to lower the significance level ($D > \alpha$) in (Fig. 4.2.6) so that a normal or a GEV decision is made rather than entering this state!
The reason is, the original values are not well presented with the EMP. It is more like returning back to the first step, where only probabilities of the values are in hand.

EMP is computed by ranking the values and the tides are treated insensitively, i.e. if a certain value occurs in several positions, the first position to occur is the one with higher rank, as the values are sorted ascending. The empirical p.value is given as following:

$$emp\_pr[v_t] = \frac{R_{v_t}}{\sum \mid s_1 \mid \cdots \mid s_n \mid}$$

where v_t is any value within feature $f_i \mid f_i \in F\{f_1 \dots f_m\}$, and R_{v_t} its associated rank.

4.3.6. Features configuration

After computing all features or after applying BGM, the result of each feature is stored in one file. From these computed scores, it is now possible to spot significant sequences, and regions.
Additionally the fitted distribution and goodness of that fitting is stored (Fig. 4.2.8). At this point, three different BGMs for each region $BGM_{5'UTR}, BGM_{3'UTR}, BGM_{CDS}$ are generated.

4.4. Significance ratio

After computing the threshold of each feature $F(S, P, V)$, every sequence s_i will be tested for significance with respect to each of the twelve significance levels. This is done with every single feature and check each sequence with its configured background model.
Ratios of sequences at each significance level for every feature $R_*^{S_*^{f_*}}$ are computed as following:

let $T_{l_{0.1}}^{f_k}$ be the threshold of feature f_k at 0.1, l is the left tail i.e. the lower threshold.

Given the set of features $\boldsymbol{F} = \{f_1 \dots f_m\}$

Given the set of sequences $\mathbf{S} = \{s_1 \dots s_n\}$, and the set of values of each sequence $\mathbf{V}^{f_k} = \{V^{s_1} \dots V^{s_n}\}$ for feature f_k.

$L = \mid s_i^{f_k} \mid$

$$if\left(v_j^{s_i} < T_{l_{0.1}}^{f_k}\right) \Rightarrow k++ \mid v_j^{s_i} \in \{v_1^{s_i} \dots v_L^{s_i}\}$$

$$R_{l_{0.1}}^{S_i^{f_k}} = \frac{0.1 * k}{x}$$

For the right tail, the same rule is used except the values $v_j^{s_i}$ should be larger than $T_{r_*}^{f_k}$.

This step is repeated for every single feature, BGM was computed for, and is repeated as well to all different data sets. At this point for any feature of lincRNA $F_{lincRNA}$, the significance of its sequences will be tested according to the threshold of its match from other data set F_{5UTR}, $F_{3UTR}, \dots$.
The results of this phase will be explained in section 4.6

4.5. Feature visualization

Another aim of this work is designing multi-layers plots, in which one can spot easily the regions of interest.
Previously mentioned (see Sec. 3.6), the library ggplot2 within R can provide multi-layers plots, however, with extreme large data sets it failed to initiate the layers of the plots.

Taking into account some of the desired plots are intended to represent the data and their BGM despite their size, and this must be visualized as such, or at least the plot must be able to give an intuition how the data might look like. Thus a compromise is needed and the data are going to be visualized, less precise however.

This was achieved by taking random subsets of the original data and estimating the resulted distribution of each, so that the plot is the estimated distribution and not the actual original data.
The drawback of this estimation is some of the cutoffs computed by the BGM might not visualized.

The mathematical ground of this estimation is: law of large numbers and sum of independent variables (see Sec. 3.4).

In case of samples with large size, in general the following steps are done:

1. find the total size of V $\rightsquigarrow \mid V \mid \in F(S, P, V)$.
2. based on $\mid V \mid$ compute a subset size *ss_size*, which is not very small, and not large. In general any number equal or more than 7 digits should be avoided.
3. generate random subsets $(SS_1 \dots SS_k)$ out of the original set $V \in F(S, P, V)$ each of size *ss_size* and unique $S \in F(S, P, V)$.

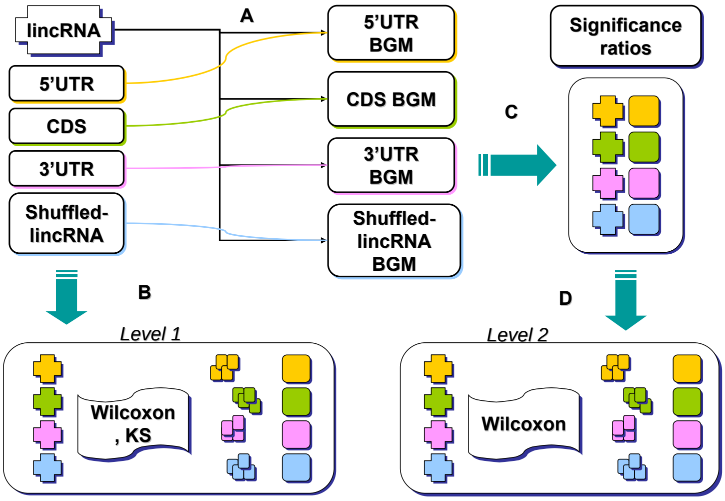


Figure 4.3.: The mechanism of comparing lincRNAs with other data sets. (A) the features of lincRNA will be computed based on the four BGMs. (B) raw-data comparison is done using KS two-samples test, and Wilcoxon test. (C) Significance ratios of all sequences of all features from BGMs are determined, and vary accordingly. (D) the significance of the sequences comparison.

4. According to BGM, find the fitting parameters of all subsets $(prm_i(SS_1)\dots prm_j(SS_k))$, i.e. if F was GEV distributed according to BGM, then $(prm_i(SS_1)\dots prm_j(SS_k))$ are the location, scale and shape.

5. Find the average of these parameters:

$$Estimated\_prm_F = \begin{cases} prm_i(SS_{1\dots k}) = \frac{1}{k}\sum_i^k prm_i(SS_1)\dots prm_i(SS_k) \\ \vdots \quad\quad \vdots \quad\quad \vdots \\ prm_j(SS_{1\dots k}) = \frac{1}{k}\sum_i^k prm_j(SS_1)\dots prm_j(SS_k) \end{cases} \tag{4.3}$$

6. Generate the estimated distribution, which should not greatly differ from the original distribution: $Estimated_F = distrb(ss\_size, prm_i(SS_1)\dots prm_i(SS_k))$

4.6. Evaluation

In this part the two types of comparison is applied (Fig. 4.3). For comparing the raw data, both tests KS and Wilcoxon are applied. We are interested in the minimum statistical difference KS computes, and the maximum of U wilcoxon computes (see Sec. 3.5). For both tests, however, the maximum p-value is taken.

The raw-data comparison was applied for 100 round, and after this comparison, it should be possible to tell, how similar the features of lincRNAs to other data set are.

The significance test is done by comparing the ratios of significance from lincRNAs against their counterparts in other data set, which the significance of lincRNA were first computed upon the BGM of that data set.

First, lincRNAs significance ratios are computed repeatedly one BGM a time for all features (Fig. 4.3.A,4.3.C). Later, a random sample is taken from the BGM to compare against lincRNAs, both samples have the same size (Fig. 4.3.D). Afterward, Wilcoxon-test is applied (Fig. 4.3.D). Repeating this mechanism several times will benefit the results by being able to see the average similarity or difference, and checking how high/low the probability of lincRNAs and other sequences can reach (Fig. 4.3).

In this phase the input is the previously mentioned five different data sets; each of these sets has twelve features to be computed (contents compositions 10x, RNALfold predicted structures 1x, RNAMotid predicted structures 1x).

4.6.1. Verification of the results

In both comparisons the **law of large number** (see Sec. 3.5) was applied, while the combination of this law and **sum of independent variables** was only used to visualize the data.

Due to the limitations the fitting function suffer from, the data were considered as normal distributed or even EMP distributed, because `fgev` failed to compute the corresponding parameters to GEV. Another possibility to handle these features is to apply both laws while fitting them (Sec. 3.6) in the same manner they were visualized (Sec. 4.5).

Following this approach on one hand, will make the computed parameters and fits less sensitive, as they are estimated, on the other hand, the fitting can be balanced again, and the choice to made is back to consider GEV.

CHAPTER 5

Implementation

This automatic analyzation is done through the following programs. The analyzation is applicable for any data set, and the background model can be computed to any data set. One important thing to note is some of these programs are designed especially for the purpose of analyzing mRNA. In Fig. 5.1, the main steps are illustrated.

5.1. Programs

The language used to implement the following programs is **perl**, and the statistical tests within these programs are in R.

5.1.1. Features generation

The following programs generate the different features including the different regions of mRNA.

Sequences_UTRCDS_classifier

The task of this script is to find the borders of the different regions (UTRs,CDS) of each sequence. The format of the output file in Tab. 5.1

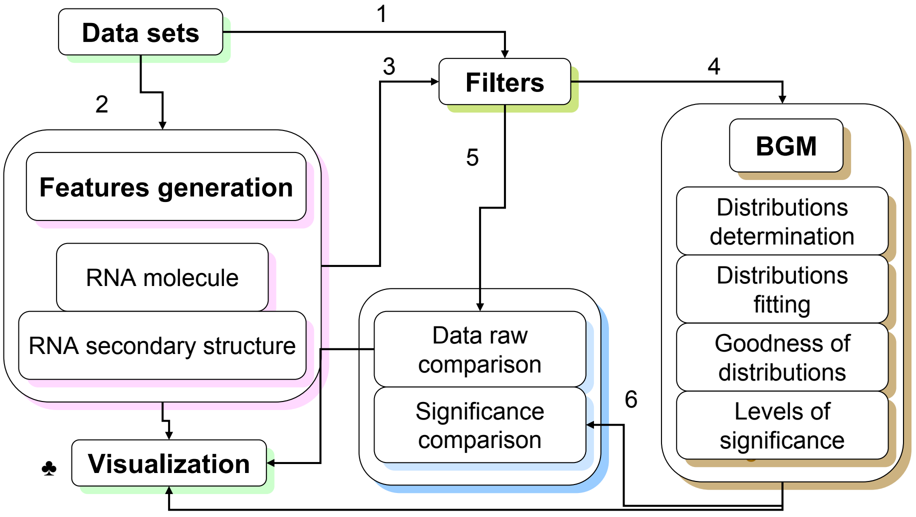


Figure 5.1.: 1- Filtering the sequences, 2- Generating the features 3- Filtering the generated features, 4- Computing the BGM for every feature, 5-6 Comparing the features twice: First,the raw data with pre-assumption of their distribution identity(KS) and without pre-assumption (Wilcoxon). Second, comparing the significant ratios of the sequences. ♣ Visualization can be done in any of these steps.

Table 5.1.: Sequences_UTRCDS_classifier Output files

File name	Columns		
my_fasta_UTRCDS.tab	sequenceID	Type	SPosition
	ID_1	CDS	113
	ID_1	3'UTR	4538
my_fasta_uncomputed_sequences.tab	Number of neglected sequences and list of their IDs		

sequence_content_raw_feature_generation

This script computes the compositions for different sequences despite their nature; i.e. the overlapping between different regions is not considered and must later taken care of in case of such need.

Features_UTRCDS_classifier

This script trims the scores of the windows,so that the overlapping contents over the CDS (5.2) and UTRs (5.1) are deleted. This is done by keeping the scores of the windows which are entirely in one region.

$$\begin{aligned} 5'UTR\_Position &= (Position + Window\_size/2) < CDS\_starting\_border \\ 3'UTR\_Position &= (Position - Window\_size/2) \geq 3'UTR\_starting\_border \end{aligned} \tag{5.1}$$

$$\begin{aligned} CDS\_Position = CDS\_starting\_border \leq (Position - Window\_size/2) \\ and (Position + Window\_size/2) < 3'UTR\_starting\_border \end{aligned} \tag{5.2}$$

RNA secondary structure

sequence_RNALfold_generation (MFE) and sequence_RNAMotid_generation (MEA) can detect mRNA sequences [1], therefore there will be an output for each.

RNAshuffle

The output will be the number of shuffled fasta files (the shuffled sequences are saved separately). The naming scheme in each shuffled file will pass the filters.

5.1.2. Features visualization

As the name indicates, the program is intended to visualize the features, and to provide a variety of plots. There are four categories: (A) sequences and their features, (B) a combination of sequences and their features and the corresponding distribution, (C) the fitted distribution computed by the background model for the features, and finally (D) binary plots of the significance of the sequence. Further details on using it can be found in its help.

[1] *my_fasta_UTRCDS.tab* exists

5.1.3. Background model (BGM)

BGM configures features!

Fig. 4.2 gives an idea how BGM is structured. The program is well commented and the task of each function is declared. There are as well parts of the programs are commented, where might be important to extend it or to read.
It is very important to mention that BGM always stores its results in a file called **Features_configuration_file.tab**.

In every run BGM checks the existence of its **Features_configuration_file.tab**, if there is an existing one, then it checks for other features than those stored in this file. BGM announces every time which features are going to be configured or no features to be configured at all.

BGM does not distinguish between files, as long as they are in the format $\{< SequenceID >, < Position >, < Value >\}$, therefor, if it is important to classify the features before using BGM.

Input: input directory where the features are stored and BGM will be saved {-d}.
Output: always one single file **Features_configuration_file.tab** contains all features found to be configured in the given directory.

Introducing a new distribution to BGM

Adding a new distribution to the BGM is way too easy! Start with creating R file[1], in which you run KS test and follow the instructions in the BGM program (1,1.A,1.B,1.C).

For visualization, you might want to add your distribution there as well, hints are in there too.

5.1.4. Sequence significance analysis

Significance_Ratio_compute

Identity of the given feature and the one configured in BGM is required.

Significance_Ratio_compute_lincRNA

Unlike the previous program, here the features can belong to different data sets the BGM was computed from. For the fact that this program was used to compare the significance of lincRNAs with mRNAs, it has been given the same name but with lincRNA ”footer”.

[1] Checking normtest.R and gevtest.R is recommended!

Table 5.2.: Wilcoxon test output

Round	bgmFtr	rSig0.1	...	lSig0.0005
1	Sequences_...	$(W^1_{r0.1}, P.Value_{r0.1})$	...	$(W^1_{l0.005}, P.Value_{l0.005})$
			$\vdots$	
-itr	Sequences_...	$(W^{-itr}_{r0.1}, P.Value_{r0.1})$	...	$(W^{-itr}_{l0.005}, P.Value_{l0.005})$
Average	Sequences_...	$\frac{\sum W^*_{r0.1}}{-itr}$	...	$\frac{\sum W^*_{l0.0005}}{-itr}$

Wilcoxon_Ratio

Previously mentioned, in this work the four BGMs (*5'UTR BGM, CDS BGM, 3'UTR BGM, shuffled_lincRNA BGM*) are computed from sequences which are at least three times more than lincRNAs. Hence, the program applies the **law of large numbers** to compare the data sets, therefore it is important to keep the origin of the data in mind and chose the parameters of the program accordingly.

Moreover, knowing the procedure Wilcoxon follows to compute the p-value (see Sec. 3.5), the program will produce warning messages considering the ties.

Input: sequences ratios file from small sample, sequences ratios file from large sample, and number of times the significance ratios between two files should be tested {-ff,-bgm,-itr}.
Output: one file as in Tab. 5.2,
e.g. **Wilcoxon_test_Sequences_Ratio_Sig_fold_L100_*my_fasta*.tab**.

5.1.5. Supplementary programs

mypackage.pm

A package used in almost every program, contains following subs:

- *read_fasta_file* : returns a hash (key: SequenceID, value: Sequence)
- *seq_check* : replaces $x \notin [ACGTU]$ to n, and T to U
- *one_line_fasta* : whole sequence in one line
- *check_length* : checks the given sequence is at least equal to the given length parameter (L for RNALfold)
- *check_n* : returns a hash with sequences have noise maximally as much as the given percentage
- *dupl_del* : **overwrites** the given fasta and delete all duplicated sequences; deleted sequences'IDs are stored in **_uncomputed_sequences*

normtest.R

In this R script the Kolmogorov-Smirnov test for normality is done.:

Passed parameters: feature file. ***Returned parameters***: 1) D 2) p.value 3) mean 4) standard deviation

gevtest.R

In this R script, the Kolmogorov-Smirnov test for generalized extreme value distribution is done using the package ***evd*** (see Sec. 3.6).

Passed parameters: feature file. ***Returned parameters***: 1) D 2) p.value 3) location 4) scale 5) shape

Ftr_Vis_help.tab

This is the help, visualization program requires, both should be in the same directory.

5.2. Technical notations

This section draws the attention to some details while running the programs.

5.2.1. Features generation

This phase requires fasta file, which contains the sequences of interest. Within every program in this phase the method "*dupl_del*" is called from "*mypackage.pm*", which in return overwrites the the input file and keeps in it the unique sequences only.
Before overwriting the file, a backup copy is created, carrying the same original name but with "*_Backupcopy*" ending.
After overwriting a file with the same original name but with " **_uncomputed_sequences*" ending is also created. In this file, all IDs of replicated sequences are listed, beside some information about total amount of sequences before and after...
Later on when applying additional filters all sequences which fail the test will be added to " **_Backupcopy*". Once all the filtering is done, the final total number of sequences to be computed will be at the end of this file.

mRNA borders and features

If the fasta file contains mRNA sequences, and the different regions are needed then the first program to run is "*Sequences_UTRCDS_classifer.pl*". This step is very essential for the rest of programs (RNALfold,RNAMotid, content compositions), as the output file "*_UTRCDS.tab*" is basic of later computations for the different features.

Once this file was generated the order of generating the features is not critical. However, as "*_UTRCDS.tab*" checks for duplicated sequences, this filter should be skipped from all features generating programs."*one_line_fasta*" which is called from *mypackage.pm* is an additional filter to be skipped, as it is included as well.

If the sequences were lincRNAs as in our work, then "*_UTRCDS.tab*" is useless; in this case these two filters should me kept in one of the features generating programs which will be the first to run, and removed from the rest. On the other hand, keeping "*one_line_fasta*" will slow the program but has no side effects!

As our works includes lincRNAs, then there are two possibilities to handle these features:

- **mRNA sequences**: apply "*Features_UTRCDS_classifier.pl*" to compute contents over the different regions of mRNA. Later, generate more features (format of output files should be always the same), or go to the next phase.
- **lincRNA sequences**: generate more features (format of output files should be always the same), or go to the next phase.

RNAMotid and RNALfold

Both "*sequence_RNAMotid_generation.pl* and **sequence_RNALfold_generation.pl**" have filtering schemes that match the nature of the programs to be called (L,mx,mn). However, the results of these filters are not printed on the screen, but they are listed in the file "**_uncomputed_sequences*".
In this work we sat the value of L,W, and mx all to 100, therefore, the total number of sequences was always the same, which what intended.

RNA shuffle

There are two points to pay attention to when applying "*RNAshulle.pl*": its input file should be already filtered, i.e. this program doesn't apply any filtering scheme.
After shuffling the sequences, the output is several files, so an additional two line script is needed to combine these files!
The reason behind not combining these files immediately was to be able to check their randomness.

5.2.2. Background model (BGM)

Running "*Features_BGM_Configuration.pl*" requires the directory, in which the features are saved. Computing the BGM will cover all the features in that directory despite what they represent and will produce only one file with always the same name. The BGM requires time to go through all the tests and computations, therefore it is a good idea to split the features at the beginning.

5.3. How to get it work!

This work consists of three phases: generating ncRNAs features, computing BGM for each feature, and for using BGM with different ncRNAs. In Appendix B "Implementation-details" a brief example was given, with the expected output of the programs if no errors occur!

5.4. Typical errors

Listing the files with their full paths as an input (some programs accept list of files, e.g. *Features_UTRCDS_ classifier.pl*) should be "∼" free as those paths will not be interpreted.

Before starting, the following R libraries must exist: timeDate, evd, ggplot2, stats.

The programs should be in one directory, or at least the supplementary programs should always be in the same directory (mypackage.pm,normtest.R,gevtest.R) with their callers.

The required parameters should always be given as described with each program, as the program checks only for a valid complete input and not the quality of it, besides the bug detection is very general, so the input might suffer from several incomplete parameters but the error message does not mention any details.

Re-applying these programs to the filtered fasta file will remove all the sequences (in case of mRNA), and will list them in " **_uncomputed_sequences*" instead. When the programs overwrite the file first time, the overwrite is done in upper case, and the different regions are not recognizable any more.
The next time applying these programs to the already overwritten file, as the sequences are now all in upper case, they will fail the first mRNA-filter (see Sec. 4.2.3).

CHAPTER 6

Results

After applying all the methods and tests needed, the results of comparing lincRNAs to the different data sets will be interpreted. The table in raw-data comparison consists of the average of testing different features of lincRNA compared to the four computed BGMs and the statistical results (D,U) of Kolmogorov-Smirnov test (KS) and Wilcoxon-Rank-Sum test (Wilx) respectively .

In significance- ratio comparison all significance levels were tested, however, only the result of the right threshold 0.05 (upper) is shown.

6.1. Raw data comparison

Raw data have been compared with Ks and Wilx. Testing the data with KS was to impose that the data are statistically similar, i.e. they have the same type of distribution, which can give the permission to believe that they might share biological meaning as well, and potentially have the same behavior (Fig.6.1).
Due to the fact, that Wilx does not make any assumption regarding the type of the underlying distribution of both samples, whereas KS does, testing the samples with both tests can benefit and confirm the final conclusion.

Figure 6.1 illustrates that the distributions of arbitrary features are identical between original lincRNA sequences, shuffled-lincRNA and CDS (see Chap.7 for details). The fact that the three independent data sets have the same type of underlying distribution

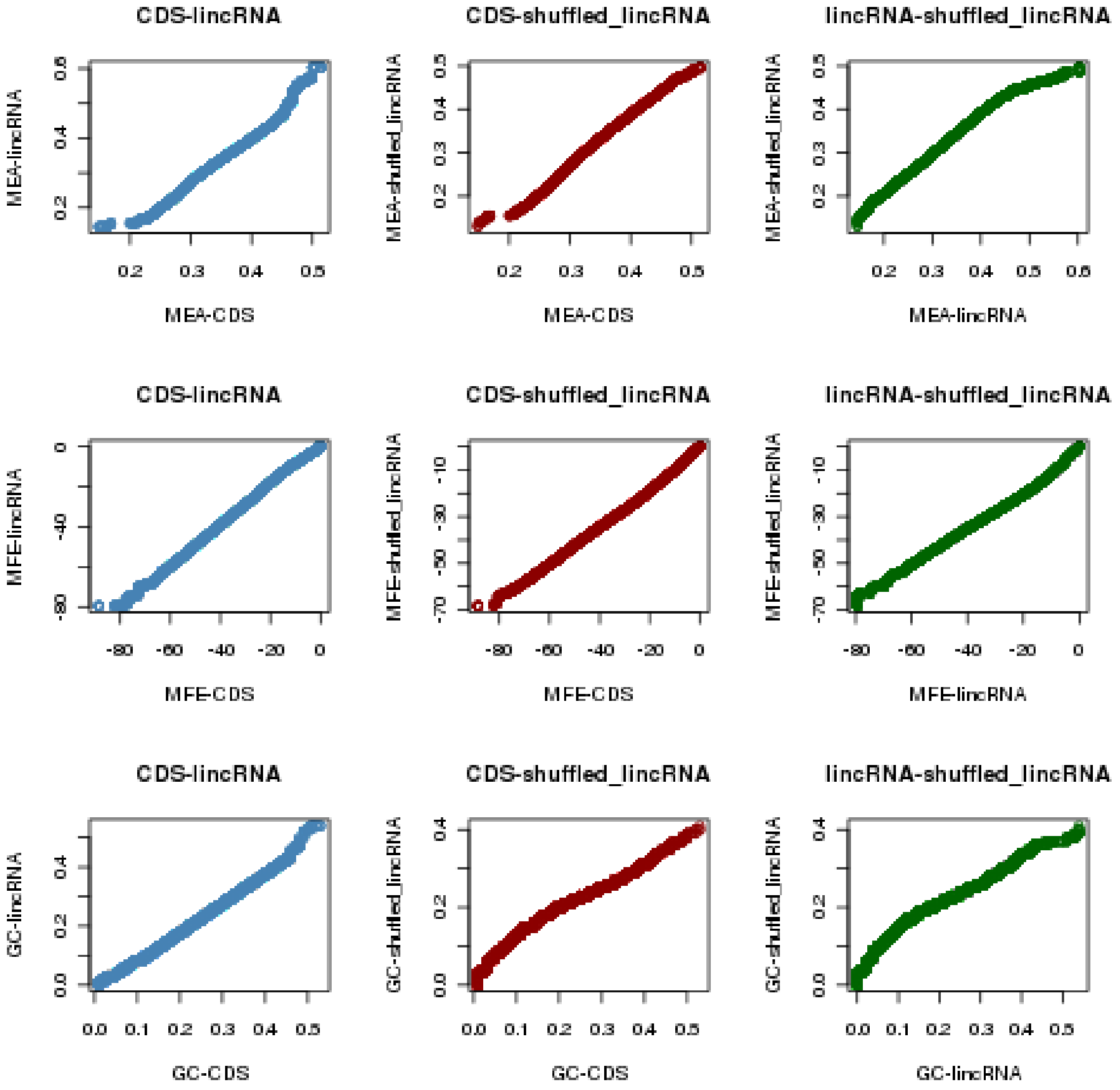


Figure 6.1.: Qqplot of an arbitrary features MFE-MEA-GC. Blue plots are the comparison between the MEA/MFE/GC of CDS and lincRNA, red plots are the comparison between the MEA/MFE/GC of shuffled-lincRNA and CDS, and green plots are the comparison between the MEA/MFE/GC of shuffled-lincRNA and lincRNA. The qqplot confirms the results of BGM, the local secondary structure and GC richness of three sets are statistically identical (same type of distribution).

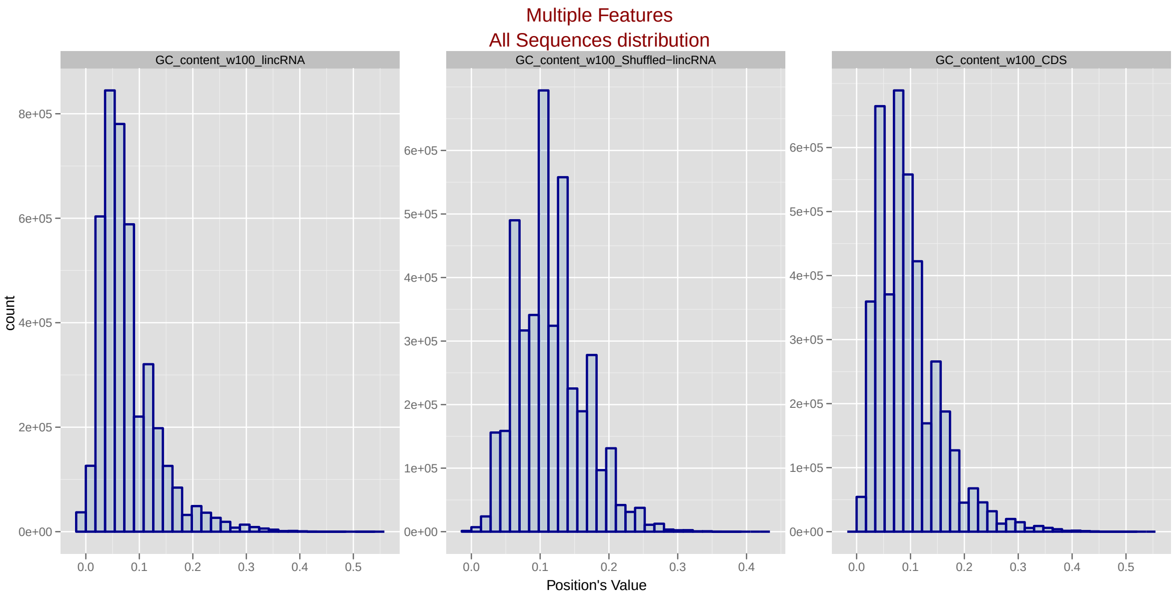


Figure 6.2.: The GC distribution of lincRNA, Shuffled_lincRNA and CDS generated by the visualization program. All distributions of the same type, however, the richness of GC of shuffled lincRNA is less than its counterparts in CDS and lincRNA and so is its distribution shape.

has been shown by our BGM as well (see Appendix A) despite the difference in their parameters. This difference was recognized by the KS and Wilx (Tab. 6.1).

The results of comparing features of lincRNA with features of both UTRs (Tab. 6.1) are not determent as one would expect in a sense the p-value of both tests is zero. Whereas comparing D to the significance levels, KS reports relatively large D for the three categories mono-nucleotide, di-nucleotide, and RNA secondary structure. Wilcoxon reports in return large U ($0 \leq U \leq n_1.n_2$), implying the difference between the medians (see Sec. 3.5).
Generally, testing lincRNAs against 3'UTR has better D,U, than testing it against 5'UTR except for the features AG,UC, and RNA SS-MEA.

The raw data of almost all lincRNAs show a reasonable similarity to the coding regions in mRNA in comparison to the untranslated regions. The relatively small D and the relatively large U indicate the possibility that lincRNA are more similar to CDS than the UTRs (Tab. 6.1). This indication will be further investigated by applying significance comparison, then comparing the significant regions in features with high probability (see Sec. 6.2).

The results of comparing shuffled lincRNAs with the original sequences (Tab. 6.1) led to completely different conclusion than what intended. For having better scores on mono-nucleotide level than other features was expected, for the fact that the shuffling was a mono-nucleotide, but having a better result for RNA SS was not expected. Shuffled

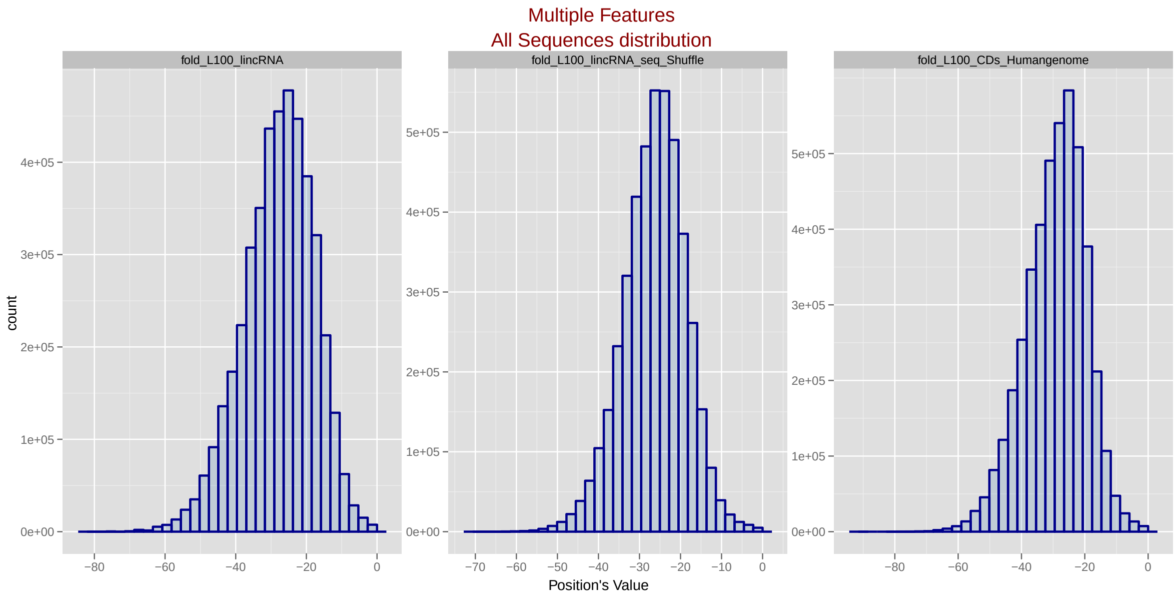


Figure 6.3.: The MFE distribution of lincRNA, Shuffled_lincRNA and CDS generated by the visualization program. An example of the failure case of $fgev$, however, the plot confirms the results of KS and Wilcoxon, as lincRNAs are particularly much like CDS .

sequences on di-nucleotide level have even better scores than mono-nucleotide (according to Wilx).

In Fig. 6.2, a visual comparison between the three data sets mRNA-CDS, lincRNA and shuffled-lincRNA, in which type of distribution and some of its properties can be easily detected. Moreover, those distributions have very asymptotic parameters (Tab. A.5, A.4, A.2).

The proposed analysis tool BGM was able to fit the three distributions to GEV (Tab. A.2, A.4,A.5). Although that Fig. 6.3 is an example of a failure case of BGM, the plots confirm the distributions identity of the three data sets.

The BGM configured MEA-CDS as a GEV, testing what kind of particular EVD[1] proved that it is a Gumbel distribution (Sec. 3.3).
This means, it is possible to assume that the underlying distribution is a normal one and compute its corresponding parameters (3.7). Hence, MEA-CDS could be $\aleph(0.35, 0.05)$ which does not contradict with neither the plots nor the tests.

[1] Testing MEA-CDS done via KS test against a random Gumbel sample that has the same fitted parameter MEA-CDS

Table 6.1.: Complete results of raw data comparison for KS and Wilx tests

Shuffled-lincRNA ⇔ lincRNA ⇔ mRNA								
	Shuffled-lincRNA		**5'UTR**		**CDS**		**3'UTR**	
	KS[D]	Wilx[U]	KS[D]	Wilx[U]	KS[D]	Wilx[U]	KS[D]	Wilx[U]
A	0.077109	8649337545104	0.34425	4708439351218	0.01871	8784628128027	0.077201	9392137866924
C	0.081767	8736939613535	0.27553	11716247321555	0.06755	9428595716328.5	0.196363	6438029272693.5
G	0.076251	8749028044550.5	0.31227	12099079275669	0.122991	9843844674315.5	0.188609	6486475756360.5
U	0.079444	8755200133651.5	0.294661	5162614546865	0.182386	6371688789441.5	0.258036	11434287151785.5
mono-nucleotide :								
$\frac{\sum mono}{4}$	**0.07864275**	**8719604226456.5**	0.30667775	8421595123826.75	**0.09790925**	**8607189327028.125**	0.18005225	6816113572850.75
$\sim$								
AC+CA	0.036379	8409735923417	0.23744	5798060320905	0.10362	9768194479212.5	0.100437	7381212743243.5
AG+GA	0.160557	6986860169010.5	0.02992	8334380973883	0.12465	9966687905062	0.144144	6868949777485
AU+UA	0.294034	11701940941934	0.34430	4662572266334.5	0.08424	7630155149658.5	0.215365	10910167587149
GC+CG	0.376172	12631057133880.5	0.43101	13233357185921	0.14079	10254308505050	0.1874656	6391405215047
GU+UG	0.046320	8211449294237.5	0.1668	6601103462071.5	0.01166	8413343713076	0.063196	9339918170503.5
UC+CU	0.136695	7193132141990.5	0.03851	8237610209770.5	0.0776	7769835198672.5	0.044104	8061064363050
di-nucleotide :								
$\frac{\sum di}{6}$	**0.1750262**	**9189029267411.667**	0.2079967	7811180736480.917	**0.0904267**	**8967087491788.583**	0.1257853	8158786309413
$\sim$								
MFE	0.079444	10490583056619.5	0.266821	6587424227663.5	0.067036	8938338571187	0.178397	11824529430438
MEA	0.105587	620466852414	0.178196	814745264546.5	0.178163	797986621028	0.238769	861260799415
RNA SS :								
$\frac{\sum ss}{2}$	**0.185031**	**5555524954516.75**	0.222509	3701084746105	**0.1225995**	**4868162596107.5**	0.208583	6342895114926.5
$\sim$								
The average of BGM								
$\frac{\sum avrg}{3}$	**0.1462333**	**7821386149461.6**	0.2457278	6644620202137.6	**0.10364515**	**7480813138308.1**	0.1714735	7105931665730.1

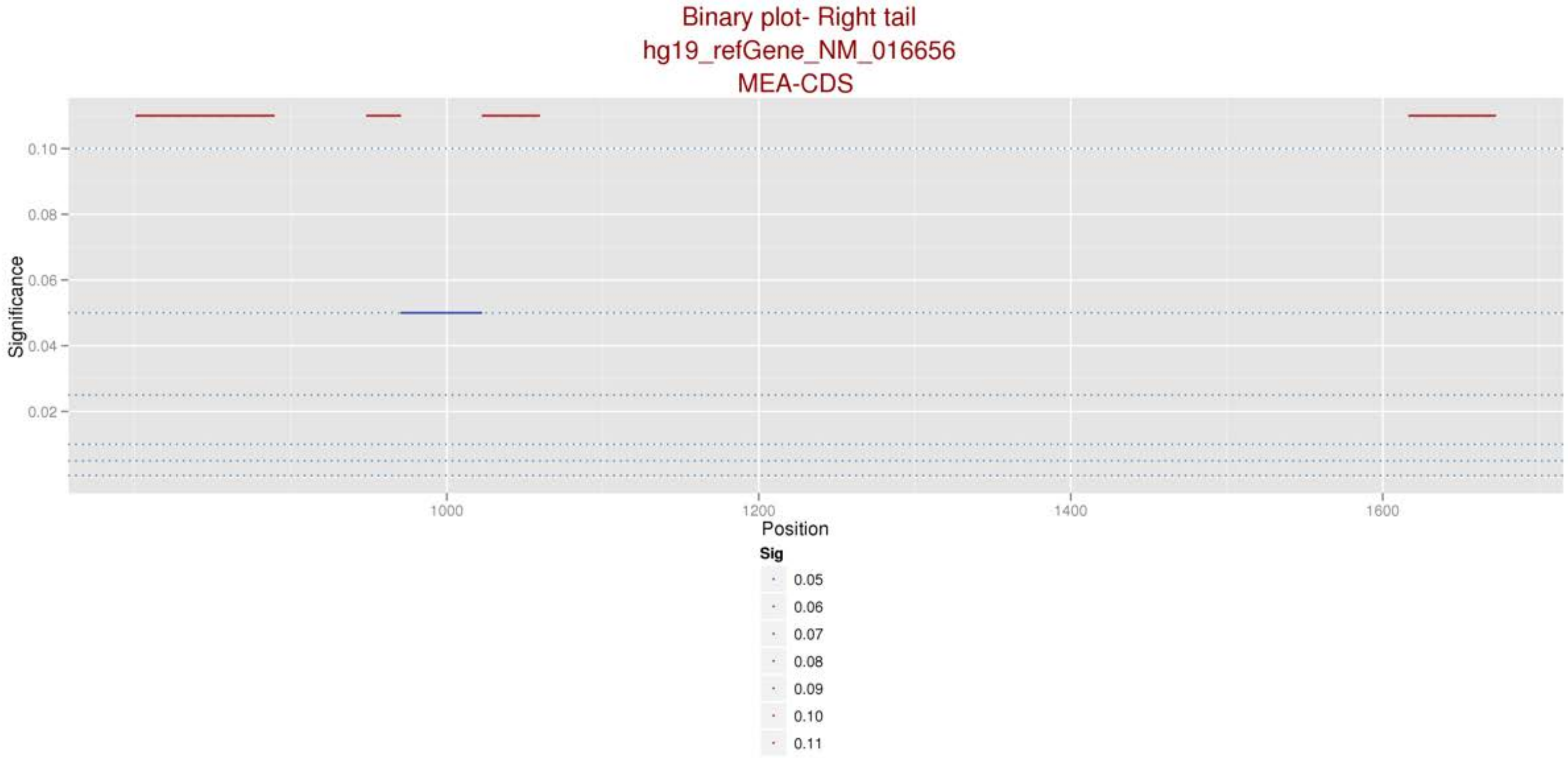


Figure 6.4.: The significant positions/regions of a random sequence hg19_refGene_NM_016656 generated by the visualization program. Blue positions (0.05) include the lower significance level as well (0.1); red positions are involved in non-significant structures(0.11). Blank areas are the areas RNAMotid predicted no structures in.

Table 6.2.: P-values of Wilcoxon test at 0.05

Feature	5'UTR	CDS	3'UTR	Shuffled lincRNAs
A	0	0.00013	0.00283	≈ 0
C	≈ 0	**0.68491**	≈ 0	≈ 0
G	≈ 0	**0.75542**	≈ 0	≈ 0
U	0	≈ 0	0	≈ 0
AC	0	≈ 0	≈ 0	≈ 0
AG	≈ 0	≈ 0	≈ 0	0
AU	0	≈ 0	≈ 0	0.00087
GU	≈ 0	**0.68491**	≈ 0	≈ 0
GC	≈ 0	≈ 0	0	≈ 0
UC	0	0	≈ 0	≈ 0
MFE	≈ 0	≈ 0	≈ 0	≈ 0
MEA	≈ 0	**0.20146**	0.00417	0.00012

6.2. Comparison of significance ratios

Previously, the significance ratios of lincRNAs were computed based on the four different BGMs (5'UTR, 3'UTR, CDs, Shuffled lincRNAs). Using Wilcoxon, the significance ratio of the sequences are tested.

In Tab.6.2 the p-values were approximated and most were abbreviated with ≈ 0 as their actual value is extremely small. On the other hand, there are p-values with exact zero value, which indicates, that these sequences under any condition are not presentable according to that BGMs (e.g. AU in 5'UTR).

6.3. Comparison of significant positions

The results imply that about 50% lincRNA features are significantly similar to their counterparts in mRNA-CDS. In order to test the sensitivity of the cutoffs offered by different BGMs, random lincRNA sequences were chosen, and visually their significant identification were compared among the four BGMs. In this work, one sequence is shown.

In Fig.6.4 a random sequence from mRNA-CDS was chosen to test its significance according to BGM_{CDS}. The total length of the sequence is 2207, the CDS starts at position 651 and ends at 1776; out of these 1125 positions only 256 found to be involved in secondary structure (according to RNAMotid), however, only the positions in $\approx]950-1050[$ are significant at levels $0.1, 0.05$[1].

[1] If the position found to be significant at position 0.025 for instance, it implies its significance at $0.1, 0.05$ as well.

In Fig. 6.5, the four different BGMS could identify significant positions/regions for the feature MEA (generated by RNAMotid). The $BGM_{5'UTR}$ detects no significant positions in the whole sequence (see Appendix A for more details about GC-5'UTR), while $BGM_{3'UTR}$ was the most sensitive one to this sequence and could identify significant positions/regions up to 0.005[1].
BGM_{CDS} and $BGM_{Shuffled-lincRNA}$ agreed on their significance identification for almost the whole sequence, however, $BGM_{Shuffled-lincRNA}$ found positions in the region 400 are more significant (up to 0.01).

The results of BGMs become even more interesting for other features, for instance GC and MFE (Fig. A.9, A.10). Those plots show how different BGMs reflect the determination of the significance of lincRNA.

[1] Positions around 400

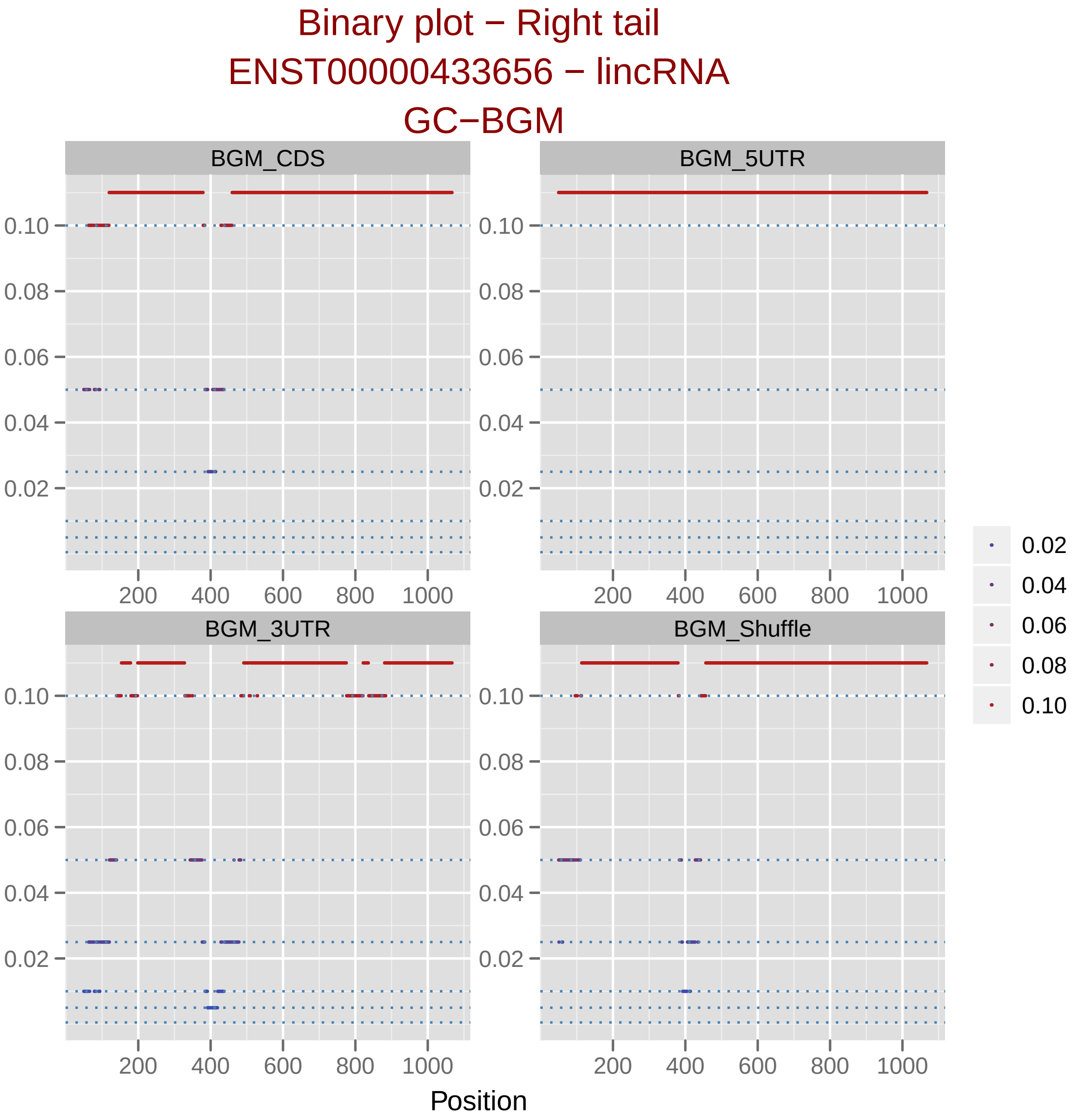


Figure 6.5.: The significant positions of a random lincRNA sequence ENST00000433656 vary according to the cutoffs of different BGM. Upper left, BGM_{5UTR} found no significant regions in the sequence, BGM_{3UTR} found many positions and regions significant up to 0.005. Both BGM_{CDS} and $BGM_{shuffled-lincRNA}$ identified almost the same positions to be significant.

CHAPTER 7

Discussion

In this chapter the results will be discussed, and features with distinguishable results will be interpreted in details.

At first glance, the results imply that the features of lincRNA are comparable to those in mRNA coding regions and to their shuffled version, noting that those sequences were mono-shuffled. Although that mono-nucleotide shuffling supposed to destroy the di-nucleotide frequency, the results of the non-parametric tests showed that some different di-nucleotide frequencies are still akin to the original sequences (e.g. GU). This kind of shuffling plus other methods was also used to investigate the structure of microRNA [53] , however, it was used with global structure and not local as the case in our work.

Moreover, comparing the significant positions/regions among different data sets, showed that about 50% of lincRNA features tend to have higher probabilities in the coding regions columns (Tab.6.2), i.e. lincRNAs, which have been configured by the BGM of the mRNA-CDS, have higher probability than those configured by other BGMs.

In other words, UTRs are known for their conserved structures, but according to the results, lincRNA are statistically very much CDS alike. We mean by statistically alike is, despite that those are completely independent different data sets, they still share not only the same type of distribution (Fig.6.1), but even the parameters of their distribution are very similar (Tab.A.5, A.4, A.2). Although that this is not enough information and does not enforce any biological meaning in particular [54], it could be a good start and might imply further meaning.

We assumed that significance regions indicate biological meaning and strongly assumed to have biological function. The other interpretation could be that these regions are on the contrast, and those extremely low probabilities of significance ratios between shuffling lincRNA and original lincRNA might imply that original lincRNAs are not structurally conserved after all. This implication can be proposed as well from comparing shuffled lincRNAs against lincRNA and CDS. This comparison was also visually done (Fig.6.2,6.3,6.1); for UTRs see (Fig.A.1)

In Tab.6.2 the probabilities varied from very low ones to very high ones. For instance it is known that UTR are rich of G+C. Hence they are supposed to be more sensitive to the Gs+Cs and the cutoffs taken from their distribution should be the most accurate one among other regions or sets. Clearly, however, lincRNAs have high probability with CDS again. This might support the previous assumption lincRNAs might not be rich of G+C and that has a direct influence on their structure.

In this work, richness of a feature is defined as the set of raw values, each value represents a window in the sequence, i.e. when richness of a feature is dropped, the maximum scores are massively reduced, but the average is still maintained if the sample was large enough (**Law of large number**); this is actually the main explenation for the behavior of shuffled-lincRNA (Fig. A.5,A.7).

Interestingly, according to the **fitting** made by **BGM** (Tab.A.5, A.4, A.2, A.1, A.3) the associated distribution of 3'UTR for feature GC is among all sets, the most similar to lincRNA. This might explain the plot (Fig.6.5). The BGM_{3UTR} could detect various significant positions in individual sequences more than other BGMs, however, testing the significance of lincRNA as a set ended up with very low probability (Wilx.p-value=0). These results was confirmed by the raw data comparison, as 5'UTR scores the worse statistical values for feature GC, shuffled-lincRNA was the second worse which was not a surprise for the type of shuffling; CDS and 3'UTR were the best of the worse.

Zooming in feature GC, showed that shuffling lincRNAs made the GC richness reduces $\sim 20\%$ from the total GC content[1], although that all the sequences still contain GC, the difference between the richness of lincRNA and shuffled-lincRNA was $\sim 20\%$. Because of that, the parameters of the distributions of shuffled-lincRNA differe from CDS and original lincRNA.

The feature AG was the only feature of lincRNA that has the worst results with shuffled-lincRNA according to both comparisons (Tab.6.2, 6.1). AG lost $\sim 50\%$ of its richness after shuffling[2], this is the main reason, the p-value of Wilx was exact 0 between all other features of lincRNA (Fig. A.4).

This sequence of procedures (Fig.5.1) is not error free; one major problem the proposed background model suffers from is its fitting function for the extreme value distribution (Sec.3.6), this problem was solved using two approaches: 1) take an alternative distribution instead 2) using *Law of large numbers* and *sum of independent variables*. The

[1] The raw values of the features for each set was tested, its density and the arthimetic range using R functions.

[2] The same zooming was applied to AG.

first approach was used with the BGM and hence some features were fitted to normal distribution (Fig.6.3). While the second approach used with visualization the features, for the library ggplot2 can not handle large amount of data (Sec.3.6).

Because of this problem (fitting problem), the BGM was forced to fit MFE to normal distribution, this can be the reason why we could not see much difference in identification of significant regions for those sets (Fig. A.10).
One can of course benefit from the two laws within the BGM while fitting the features (Fig.5.1.4, 4.2.1), i.e. computing the estimated parameters instead of the real ones.

Considering that the puzzle of ncRNA is neither solved nor well understood ”yet”, the proposed approach in this thesis was intended to model different data sets besides mRNA and lincRNA. The BGM was designed without any biological influence. Therefore, BGM can be considered as an independent tool and valid to different sequences particularly ncRNAs such as miRNA.

CHAPTER 8

Conclusion

In this work we investigated lincRNAs and their features and collected evidences about their possible structural nature with respect to known class of RNA namely mRNA.

We used statistics and nonparametric tests to categories the different regions and create a statistical analyzing tool. Our tool (BGM) could successfully fit the data to the most appropriate distribution and estimate its associated parameters.
Moreover, we highlighted the failure cases and proposed two different solutions for them. Nevertheless, the results were validated using two independent different non-parametric tests.

Afterward, the significant regions of the two RNA classes mRNA and lincRNA were identified based on that proposed tool. The ability of identifying these regions might ease the search for functioning regions in long sequences and short ones as well.

In this work, all the previous steps can be visualized, however, we showed only some results-related plots.

Finally, this work investigated only human sequences. BGM can be efficiently used among different species to find significant similarity or differences for its independence (no biological influence) and flexibility (other distributions can be easily added). Therefore BGM is capable of handling different data and biological sets, including the large ones even for different purposes, e.g. with biomarker.

APPENDIX A

Results-details

Features configuration - Background Model

After generating the features, the proposed analysis tool BGM has fitted the features to one of the three distributions mentioned in section 3.3. The critical value α was lowered to 0.1. The plots of some features are given as well.
Accordingly the features are best fitted to $\xi(\varrho_1 \dots \varrho_x)$, where ξ is the distribution and $\varrho_1 \dots \varrho_x$ its parameters:

Table A.1.: BGM- Features configuration

5' untranslated regions						
mono-nucleotide					RNA SS	
	A	C	G	U	MFE	MEA
Distribution	$\wp(0.2, 0.1, -0.1)$	$\wp(0.3, 0.1, -0.2)$	$\wp(0.3, 0.1, -0.2)$	$\wp(0.2, 0.1, -0.1)$	$\aleph(-34.8, 13.5)$	$\wp(0.3, 0, -0.2)$
di-nucleotide						
	AC	AG	AU	GC	GU	UC
Distribution	$\wp(0.1, 0, -0.1)$	$\wp(0.1, 0.1, -0.1)$	$\wp(0, 0, 0.4)$	$\wp(0.1, 0.1, -0.01)$	$\wp(0.1, 0, -0.1)$	$\wp(0.1, 0.1, -0.1)$

Table A.2.: BGM- Features configuration

Coding regions						
mono-nucleotide					RNA SS	
	A	C	G	U	MFE	MEA
Distribution	$\aleph(0.3, 0.1)$	$\wp(0.2, 0.1, -0.1)$	$\wp(0.2, 0.1, -0.1)$	$\aleph(0.2, 0.1)$	$\aleph(-29.2, 9.6)$	$\wp(0.3, 0, -0.2)$
di-nucleotide						
	AC	AG	AU	GC	GU	UC
Distribution	$\wp(0.1, 0, -0.1)$	$\aleph(0.2, 0.1)$	$\wp(0.1, 0, -0.1)$	$\wp(0.12, 0, 0.1)$	$\wp(0.1, 0, -0.1)$	$\wp(0.1, 0, -0.1)$

Table A.3.: BGM- Features configuration

3' untranslated regions						
mono-nucleotide					RNA SS	
	A	C	G	U	MFE	MEA
Distribution	$\aleph(0.3, 0.1)$	$\wp(0.2, 0.1, -0.1)$	$\wp(0.2, 0.1, -0.1)$	$\aleph(0.3, 0.1)$	$\aleph(-24.4, 9.6)$	$\wp(0.3, 0, -0.1)$
di-nucleotide						
	AC	AG	AU	GC	GU	UC
Distribution	$\aleph(0.1, 0)$	$\wp(0.1, 0, -0.1)$	$\wp(0.1, 0.1, -0.1)$	$\wp(0, 0, 0.2)$	$\wp(0.1, 0, -0.1)$	$\aleph(0.1, 0.1)$

Table A.4.: BGM- Features configuration

lincRNA shuffled sequences						
mono-nucleotide					RNA SS	
	A	C	G	U	MFE	MEA
Distribution	$\aleph(0.3, 0.1)$	$\wp(0.2, 0.1, -0.1)$	$\aleph(0.2, 0.1)$	$\aleph(0.3, 0.1)$	$\aleph(-26.3, 7.6)$	$\aleph(0.3, 0.04)$
di-nucleotide						
	AC	AG	AU	GC	GU	UC
Distribution	$\aleph(0.1, 0)$	$\wp(0.1, 0, -0.1)$	$\wp(0.1, 0.1, -0.2)$	$\wp(0.1, 0, -0.1)$	$\wp(0.1, 0, -0.1)$	$\wp(0.1, 0, -0.1)$

Table A.5.: BGM- Features configuration

lincRNA						
mono-nucleotide					RNA SS	
	A	C	G	U	MFE	MEA
Distribution	$\aleph(0.3, 0.04)$	$\aleph(0.2, 0.08)$	$\wp(0.3, 0.1, -0.2)$	$\aleph(0.3, 0.1)$	$\aleph(-27.9, 9.98)$	$\aleph(0.3, 0.05)$
di-nucleotide						
	AC	AG	AU	GC	GU	UC
Distribution	$\wp(0.1, 0, -0.1)$	$\wp(0.1, 0.1, -0.1)$	$\wp(0.1, 0.1, -0.001)$	$\wp(0.1, 0, 0.1)$	$\wp(0.1, 0, -0.1)$	$\aleph(0.1, 0.1)$

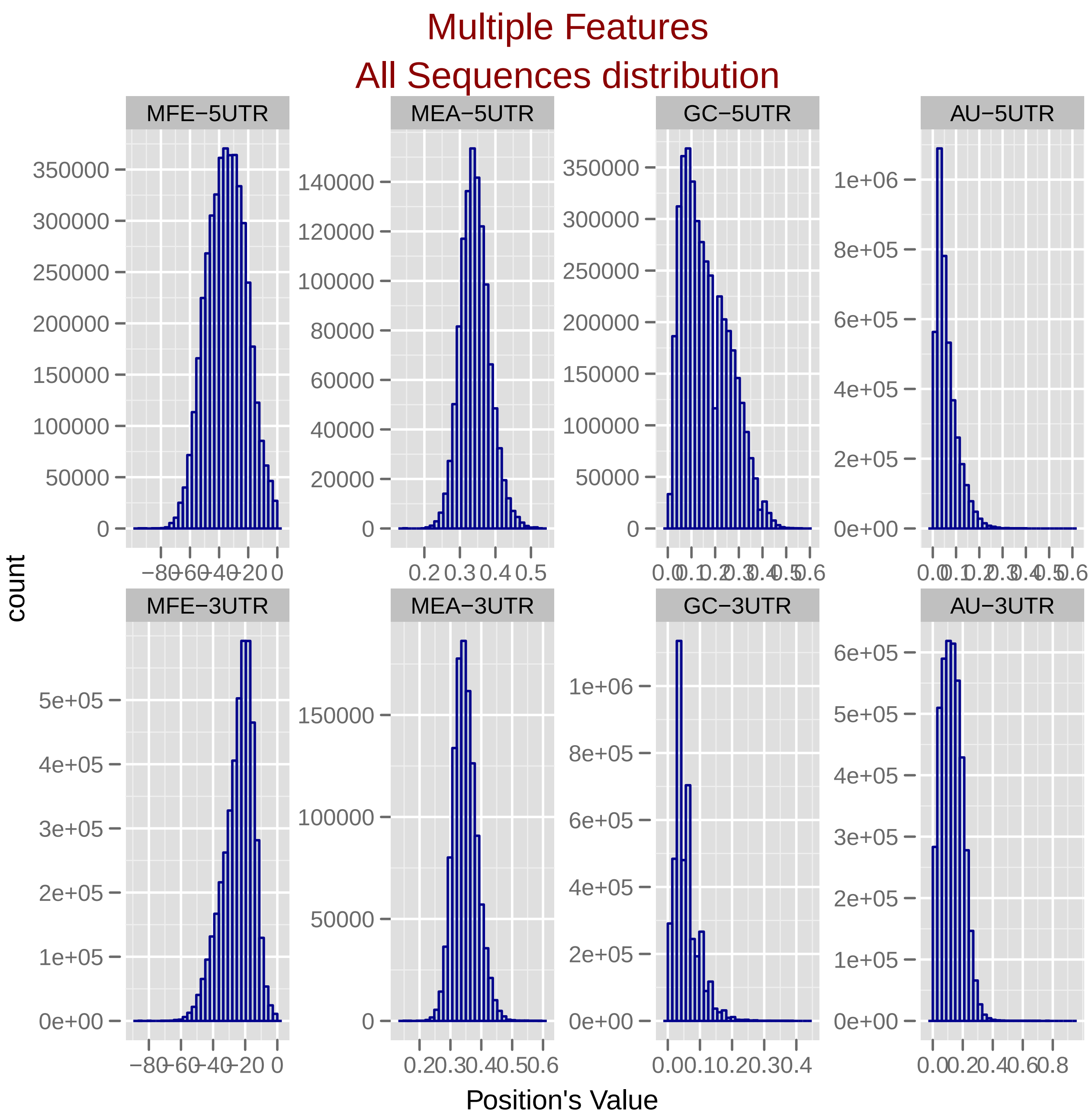


Figure A.1.: Three features of 3-5UTR namely MEA,MFE, and GC.

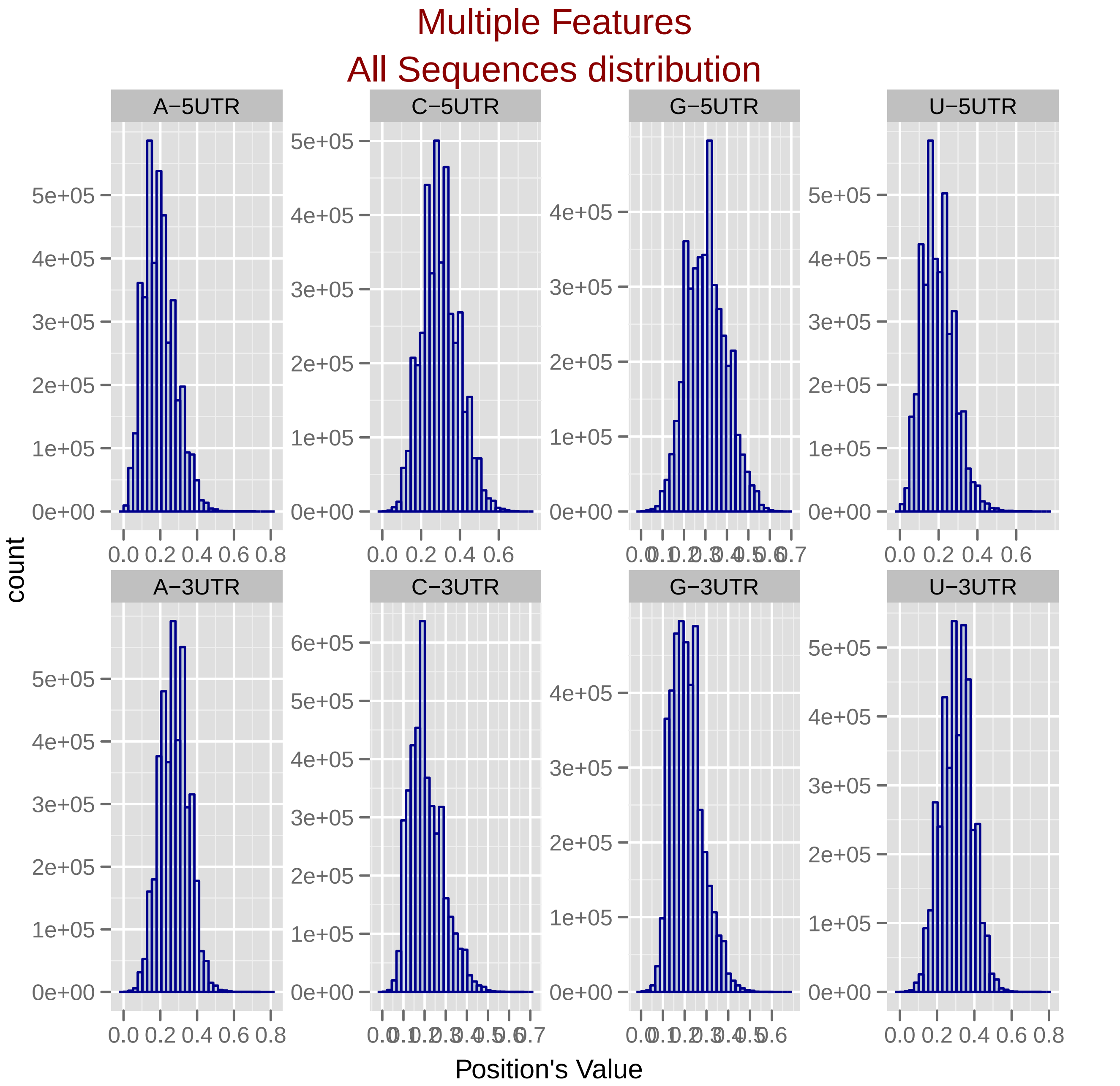


Figure A.2.: First row is the mono-nucleotide frequency of 5UTR. Second row is the 3UTR.

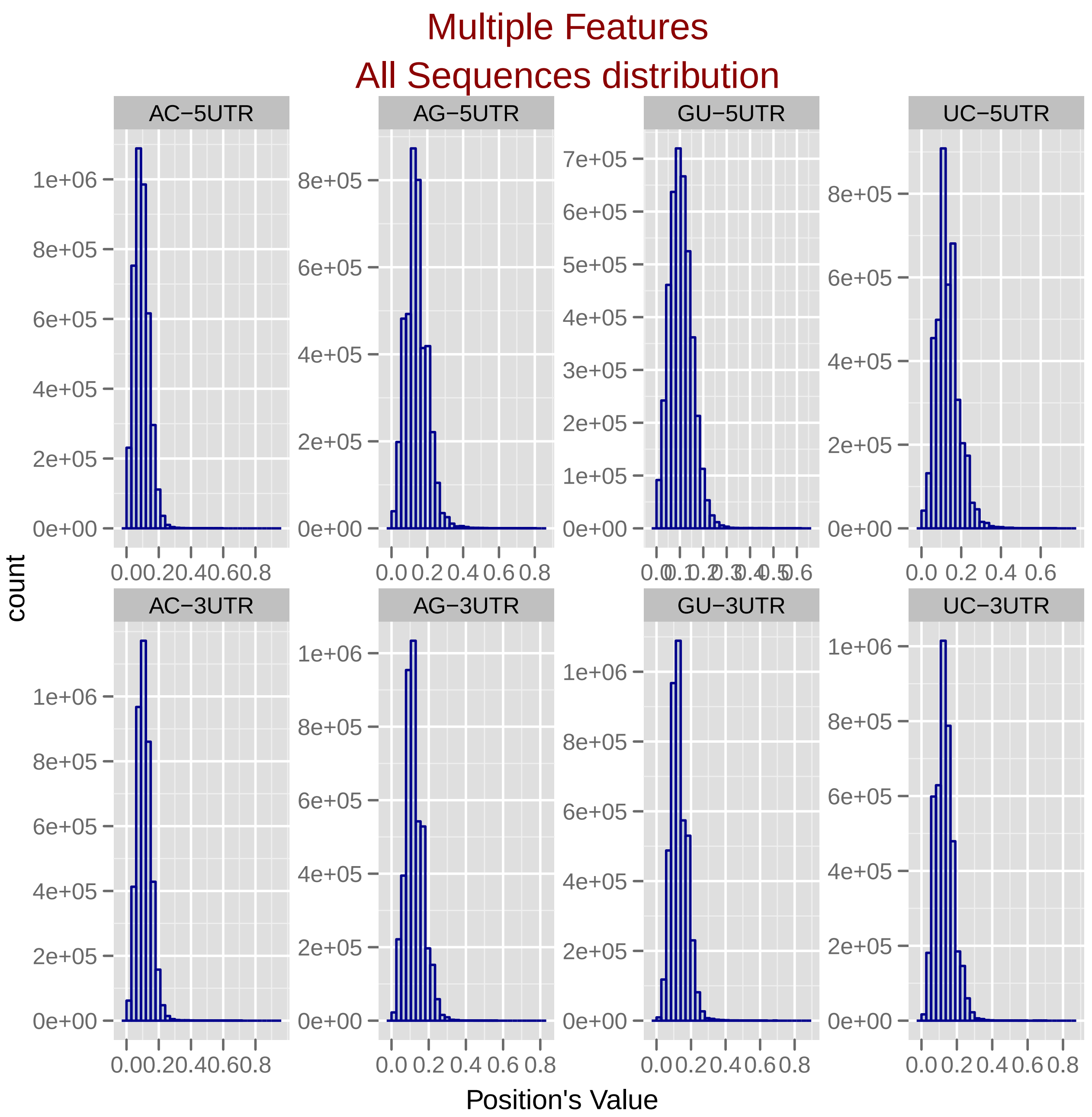


Figure A.3.: First row is the di-nucleotide frequency of 5UTR. Second row is the di-nucleotide frequency of 3UTR.

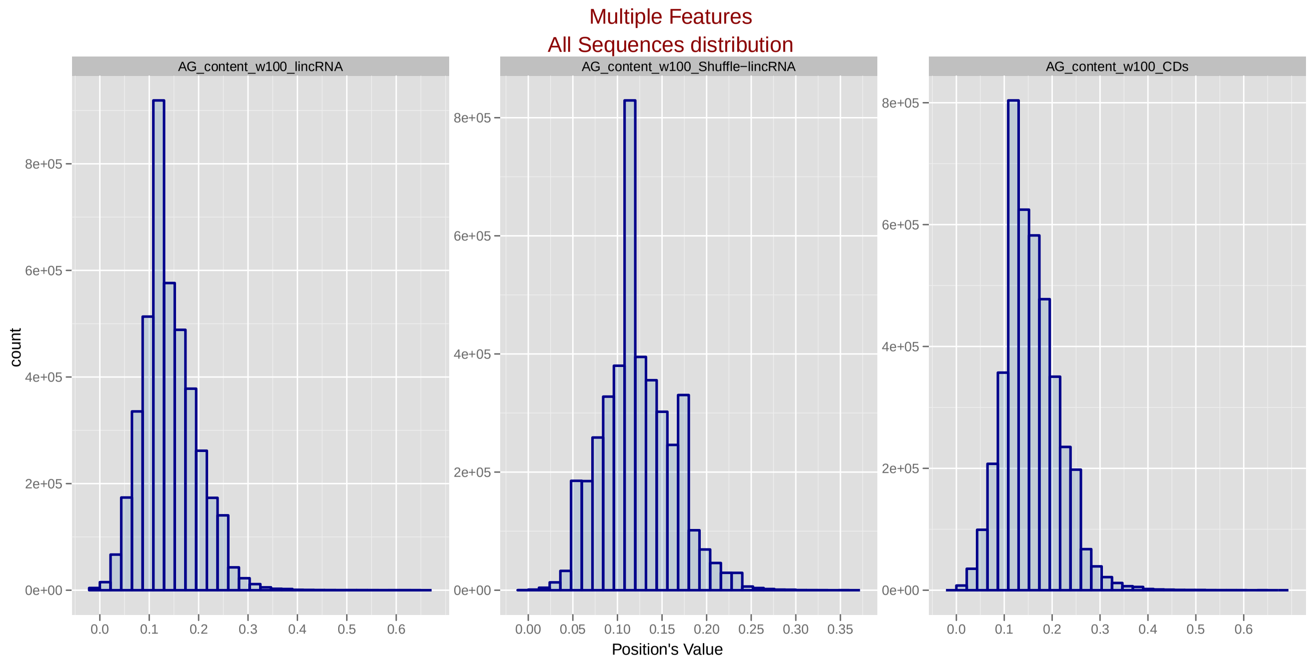


Figure A.4.: The distribution of feature AG in the three sets, the parameters of th distribution differ in shuffled-lincRNA from the other two sets.

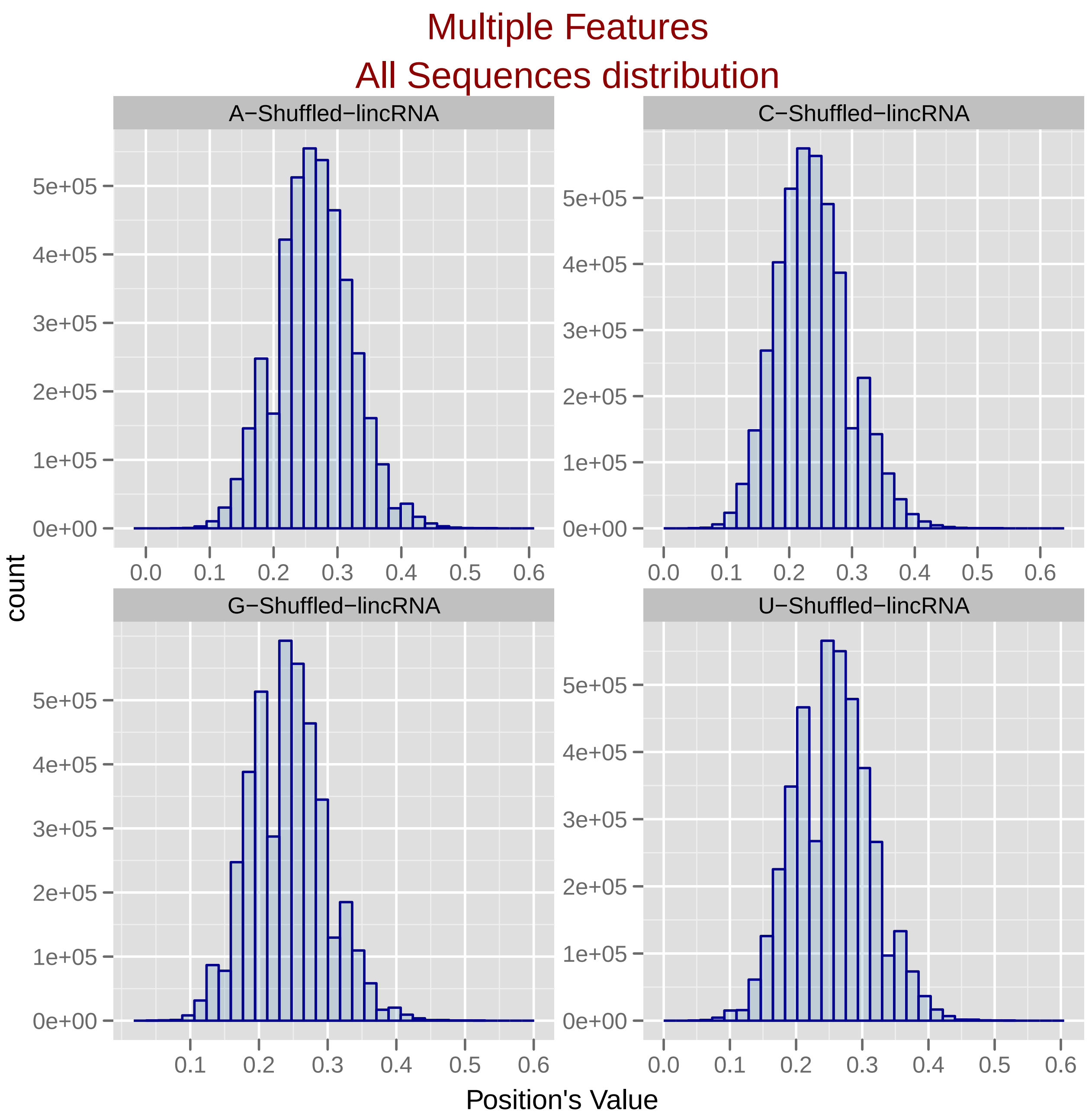


Figure A.5.: The distributions of mono-nucleotides of shuffled-lincRNA.

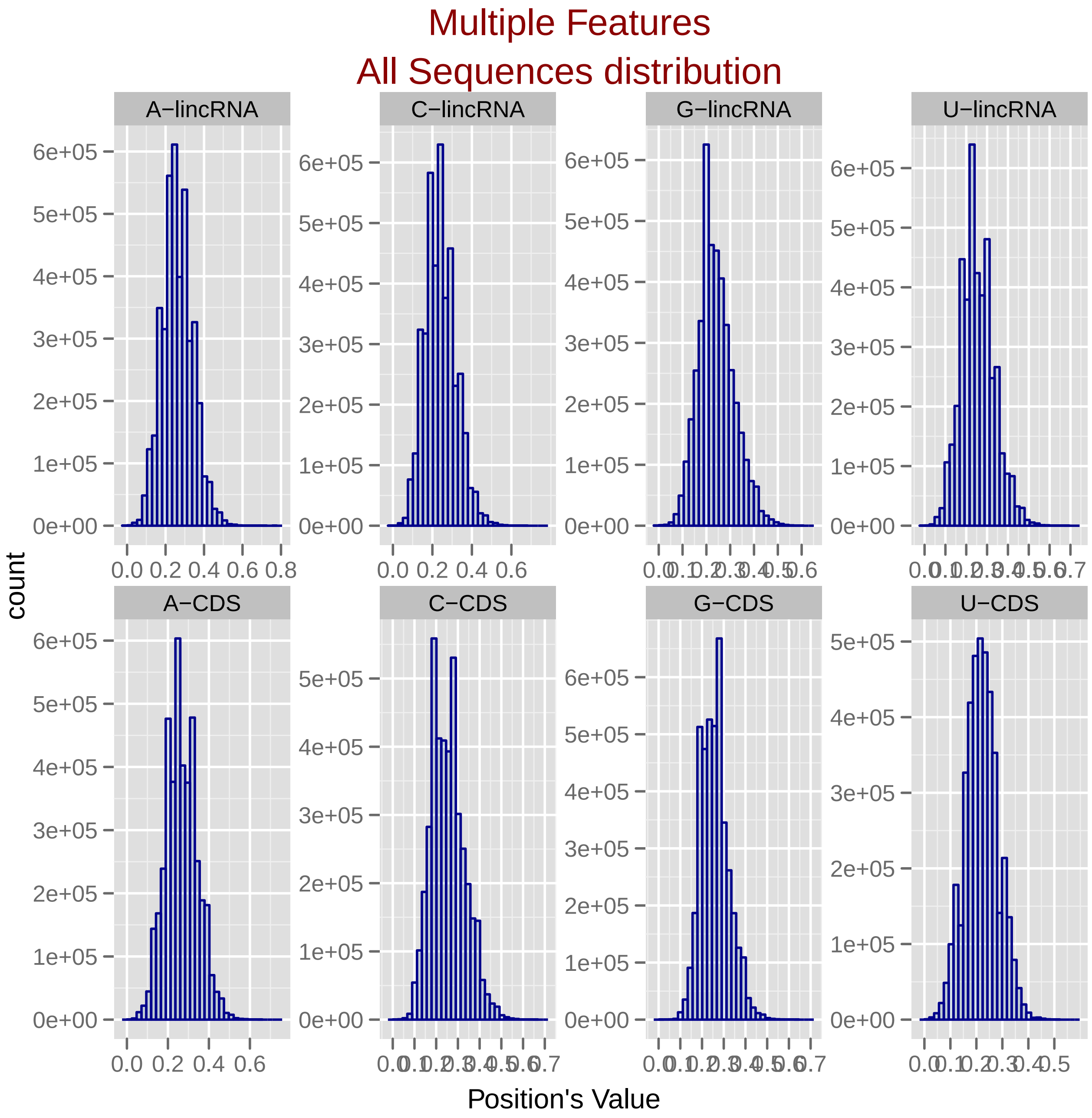


Figure A.6.: The distributions of mono-nucleotides of both CDS and lincRNA.

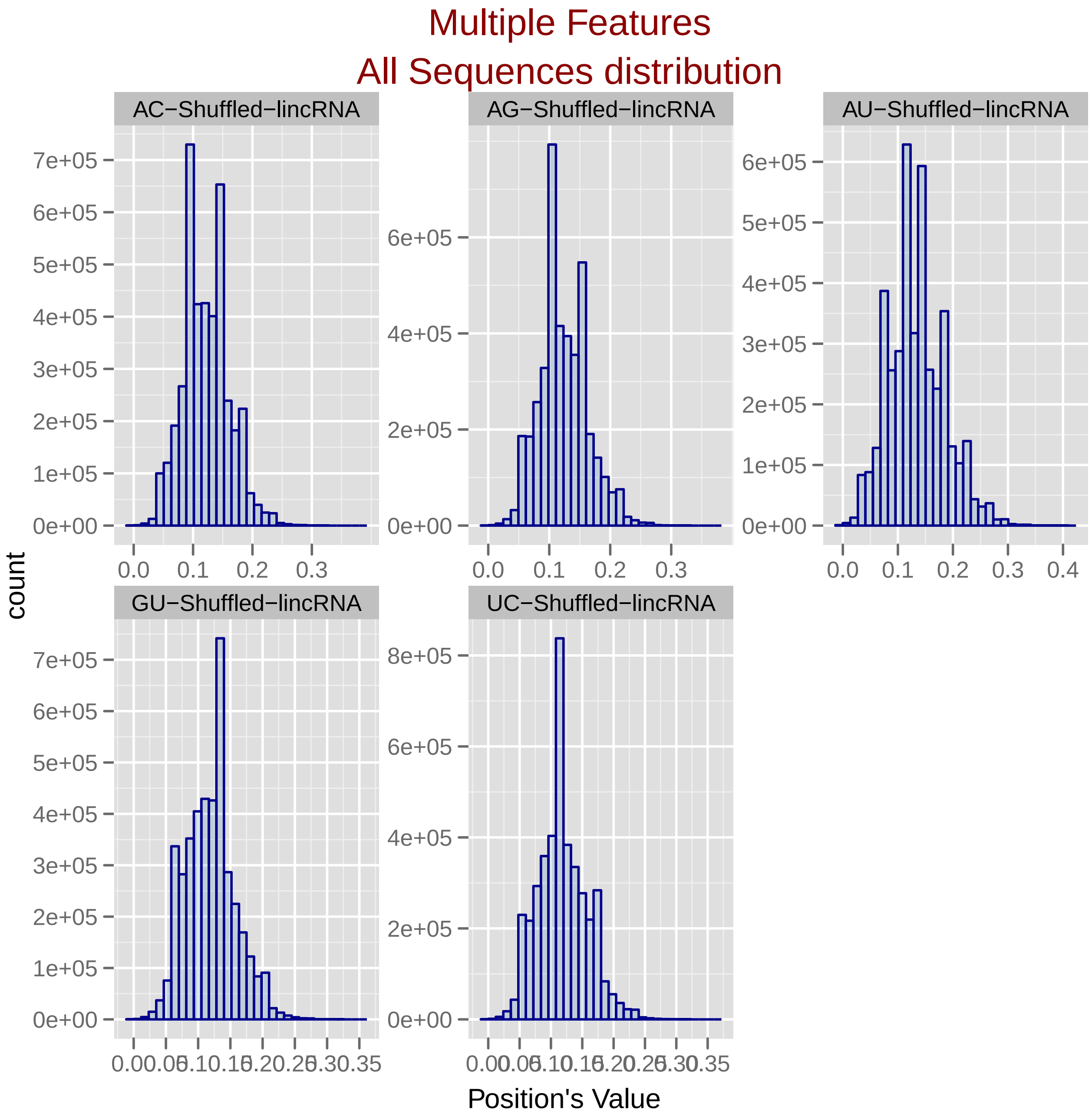


Figure A.7.: The distributions of di-nucleotides of shuffled_lincRNA.

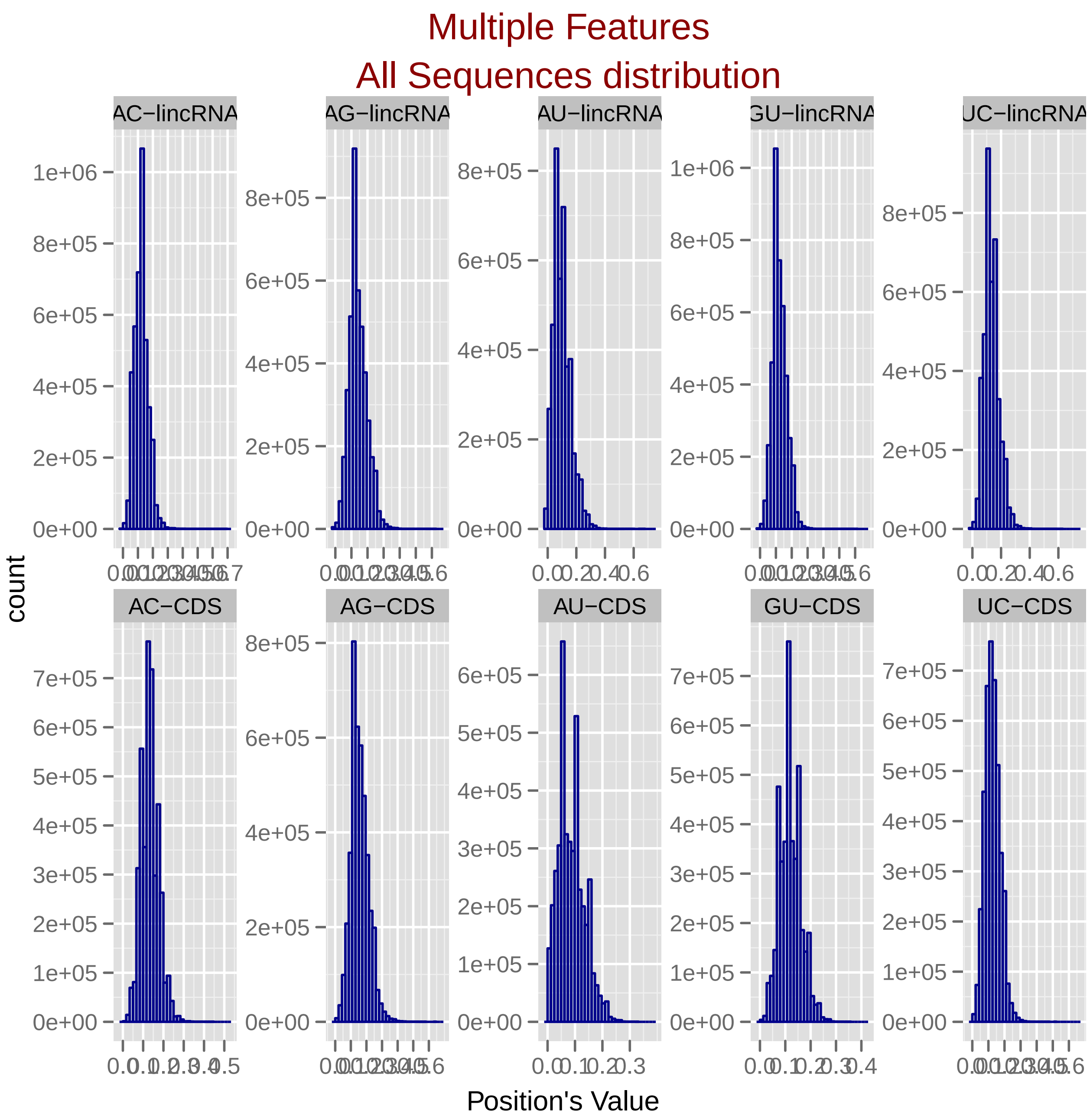


Figure A.8.: The distributions of mono-nucleotides of both CDS and lincRNA.

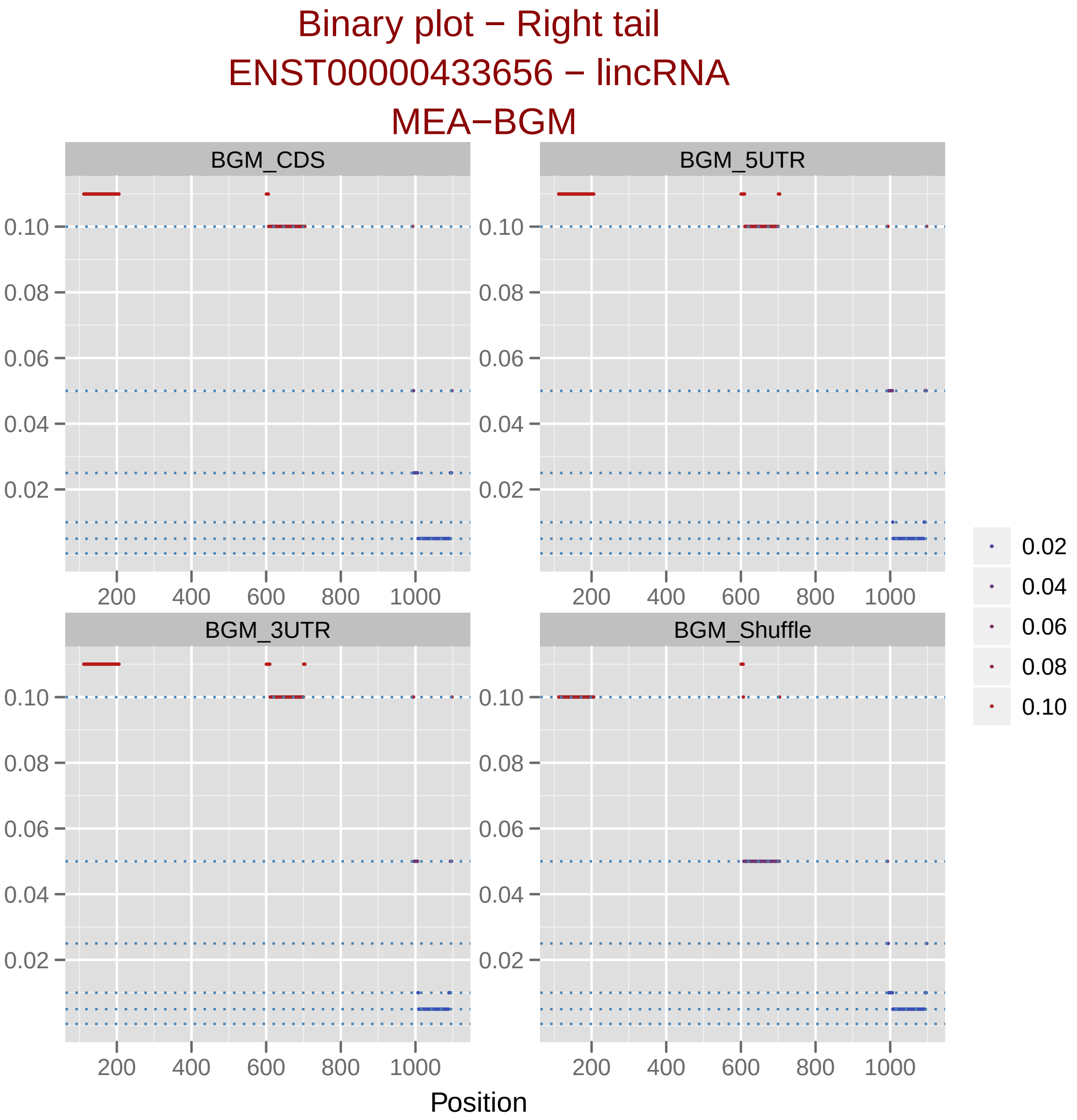


Figure A.9.: The significant positions of a random lincRNA sequence ENST00000433656.

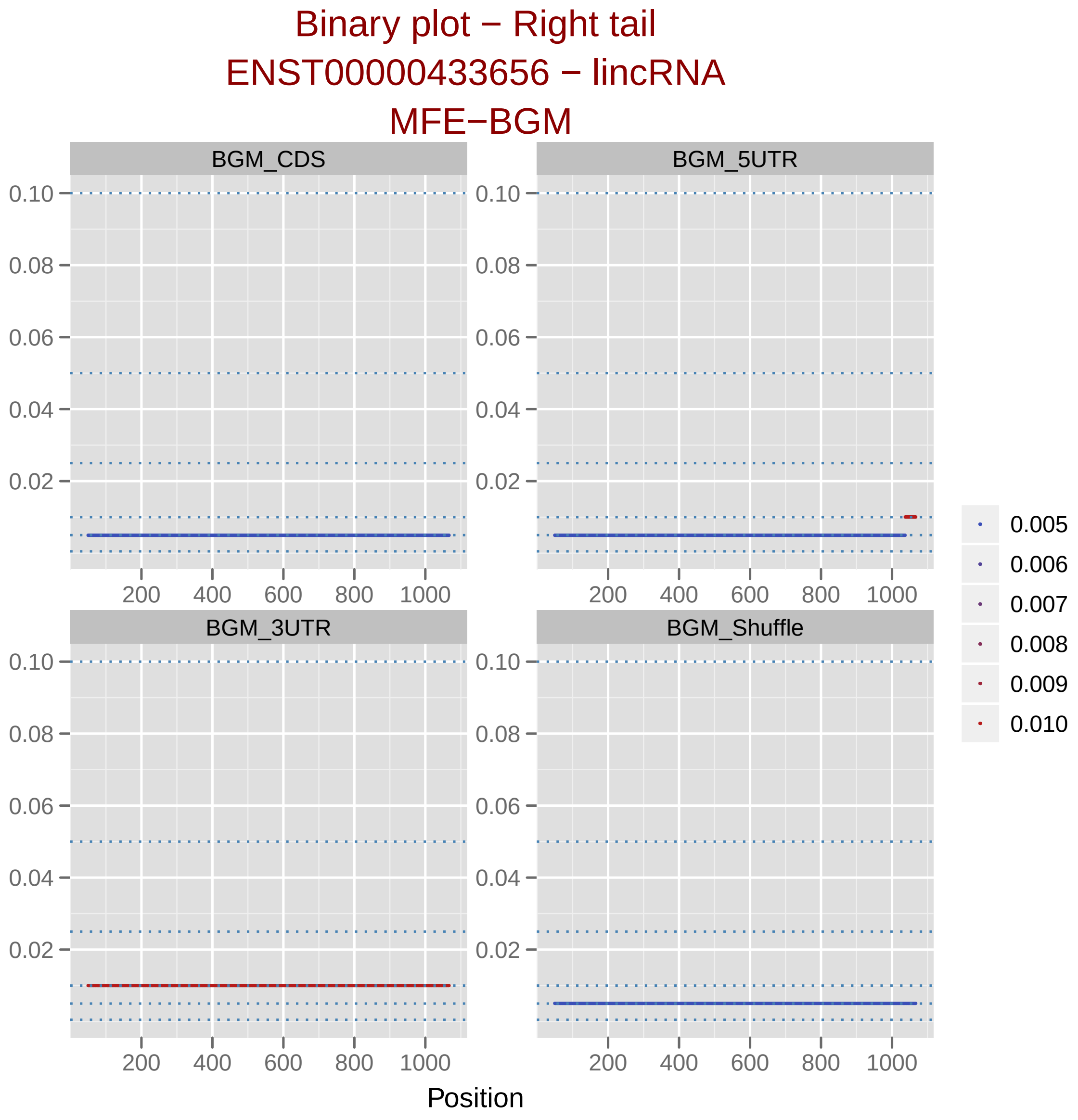


Figure A.10.: The significant positions of a random lincRNA sequence ENST00000433656.

APPENDIX B

Implementation-details

This detailed example was done from small mRNA sample size (∼ 2000 sequences), hence, the results in this example are not the final results of the whole mRNA set.

Features generation

1. **Input**: `Sequences_UTRCDs_classifer.pl ∼/mRNA/mRNAsequences`

2. **Input**:`sequence_content_raw_feature_generation.pl -w 100 -o ∼/mRNA/`
   `-f ∼/mRNA/mRNAsequences`

3. **Input**: `Features_UTRCDs_ classifier.pl -w 100 -o ∼/mRNA/contents/`
   `-f ∼/mRNA/contents/A_content_w100.tab`
   `-utrcd ∼/mRNA/mRNAsequences_UTRCDs.tab`

4. **Input**:
   `sequence_RNAMotid_generation.pl -mx 100 -mn 50 -o ∼/mRNA/`
   `-f ∼/mRNA/mRNAsequences`

5. **lincRNA**:`sequence_content_raw_feature_generation.pl -w 100 -o ∼/lincRNA/`
   `-f ∼/lincRNA/lincRNAsequences`

6. **lincRNA**:
   `sequence_RNAMotid_generation.pl -mx 100 -mn 50 -o ∼/lincRNA/`
   `-f ∼/lincRNA/lincRNAsequences`

Background model BGM

1. **Features_BGM_Configuration.pl**:

   - **Input**: `Features_BGM_Configuration.pl -d ~/mRNA/contents/`
   - **Output**: See figure B.1.
     If the one-sample KS test showed that some features are bad fitted, another couple of lines will be written right after step (5).

     ```
     -----------------------------------
     The following feature(s) has(ve) high D value(s) [D cutoff = 0.05]:
     1 - A_content_w100_5UTR 0.0575091575

     Compute the Empirical P-Value for A_content_w100_5UTR.tab
     Prepare data to fitting

     Saving empirical values of A_content_w100_5UTR.tab
     -----------------------------------------
     Creating and writing features into Features_configuration_file.tab
     ------------------------------------------
     Features_configuration_file.tab is saved in ~/mRNA/contents/
     -----------------------------------------
     ```

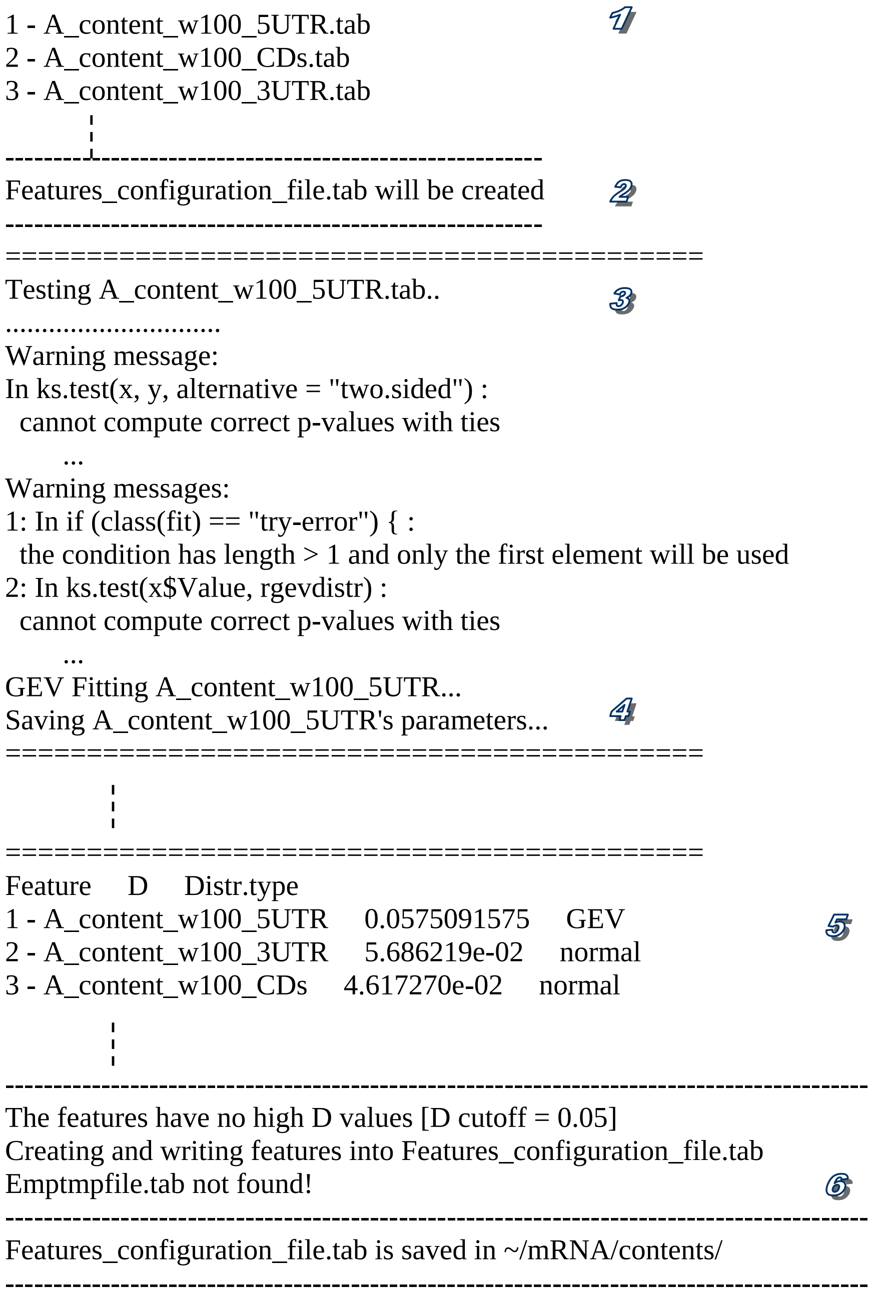
2. **lincRNA**:`Features_BGM_Configuration.pl -d ~/lincRNA/contents/`

Sequence significance analysis

1. **Input**: `Significance_Ratio_compute.pl -d ~/mRNA/contents/`
   `-o ~/mRNA/contents/`

2. **lincRNA**:`Significance_Ratio_compute.pl -d ~/lincRNA/contents/ -o ~/lincRNA/conte`

3. **Input**: `Wilcoxon_Ratio.pl -itr 5`
   `-bgm ~/mRNA/contents/Sequence_Ratio_A_content_100_5UTR.tab`
   `-ff ~/lincRNA/contents/Sequence_Ratio_A_content_100.tab`



```
1 - A_content_w100_5UTR.tab
2 - A_content_w100_CDs.tab
3 - A_content_w100_3UTR.tab

-----------------------------------------------------
Features_configuration_file.tab will be created
-----------------------------------------------------
=============================================
Testing A_content_w100_5UTR.tab..
..............................
Warning message:
In ks.test(x, y, alternative = "two.sided") :
  cannot compute correct p-values with ties
       ...
Warning messages:
1: In if (class(fit) == "try-error") { :
  the condition has length > 1 and only the first element will be used
2: In ks.test(x$Value, rgevdistr) :
  cannot compute correct p-values with ties
       ...
GEV Fitting A_content_w100_5UTR...
Saving A_content_w100_5UTR's parameters...
=============================================

=============================================
Feature     D     Distr.type
1 - A_content_w100_5UTR     0.0575091575     GEV
2 - A_content_w100_3UTR     5.686219e-02     normal
3 - A_content_w100_CDs     4.617270e-02     normal

--------------------------------------------------------------------------------------
The features have no high D values [D cutoff = 0.05]
Creating and writing features into Features_configuration_file.tab
Emptmpfile.tab not found!
--------------------------------------------------------------------------------------
Features_configuration_file.tab is saved in ~/mRNA/contents/
--------------------------------------------------------------------------------------
```

Figure B.1.: A typical output of the BGM, in which all computations were done normally. (1) A list of the files BGM found in the given directory and will check them. (2) BGM searches for Features_configuration_file.tab in this case it did not find an existing one so it gives a hint, that this file will be now created. (3) Testing the first feature in the list. (4) The results of testing says that this feature is going to be fitted to the best distribution, here it is a GEV one. Between 3 and 4 are the normal R warnings! (5) The result of fitting all features (step 5 in figure 4.2). (6) The Message Emp... not found! occurs when no features were found to have bad fitting and will not be EMP refitted. I.e. the previous decisions made in (5) are final.

APPENDIX C

Abbreviation

-A-	
A	Adenine
-B-	
BGM	Background model
-C-	
C	Cytosine
CDS	Coding regions
-D-	
DNA	Deoxyribonucleic acid
DP	Dynamic programming
D	Maximum statistical difference of KS

-E-	
EVD	Extreme value distribution
EMP	Empirical p-value distribution
-G-	
G	guanine
GEV	Generalized extreme value distribution
-H-	
HOX	HOX gene- determines basic structure of an organism
HOTAIR	HOX antisense intergenic RNA
-K-	
KS	Kolmogorov-Smirnov test
-L-	
lncRNA	Long non-coding RNA
lincRNA	Large intervening non-coding RNAs
-M-	
miRNA	microRNA - short ribonucleic acid
MRE	miRNA response element
mRNA	Messenger RNA
MFE	Minimum free energy
MEA	Maximum expected accuracy
MLE	Maximum likelihood estimation
MCMP	Maximum circular matching problem
-N-	
ncRNA	non-coding RNA
nt	nucleitide

-O-	
ORF	Open reading frame
-P-	
piRNA	piwi-interacting RNA
PRC2	Polycomb Repeessive Complex 2
pre-mRNA	Precursor messenger RNA
-R-	
RNA	Ribonucleic acid
RNA SS	see SS
rRNA	Ribosome RNA
-S-	
snoRNA	small nucleolar RNA
siRNA	small interfering RNA
SS	Secondary structure
-T-	
T	Thymine
tRNA	Transfer RNA
-U-	
U	Uracil
UTR	Untranslated region
-W-	
Wilx	Wilcoxon test
-X-	
XIST	X-inactive specific transcript

APPENDIX D

Declaration

Statement of authenticity

I hereby declare, that I am the sole author and composer of my Thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work. I hereby also declare, that my Thesis has not been prepared for another examination or assignment, either wholly or excerpts thereof.

Erklärung

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

Leipzig, November 8, 2011

Bibliography

[1] Alberts Bray Hopkin Johnson Lewis Raff Roberts Walter. *Essential Celll Biology*. Garland Science, Taylor & Francis Group and others, 2010.

[2] Jaspreet S. Khurana, Program in Molecular Medicine William E. Theurkauf, and Worcester MA 01605 USA Program in Cell Dynamics, University of Massachusetts Medical School. pirna function in germline development. 2008.

[3] Florian Erhard and Ralf Zimmer. Classification of ncRNAs using position and size information in deep sequencing data. *Bioinformatics*, 26(18):i426–i432, September 2010.

[4] Y. P. Mei, J. P. Liao, J. P. Shen, L. Yu, B. L. Liu, L. Liu, R. Y. Li, L. Ji, S. G. Dorsey, Z. R. Jiang, R. L. Katz, J. Y. Wang, and F. Jiang. Small nucleolar RNA 42 acts as an oncogene in lung tumorigenesis. *Oncogene*, aop(current), October 2011.

[5] URL `http://www.wikipedia.org`.

[6] Leonardo Salmena, Laura Poliseno, Yvonne Tay, Lev Kats, and Pier P. Pandolfi. A ceRNA Hypothesis: The Rosetta Stone of a Hidden RNA Language? *Cell*, 146: 353–358, 2011.

[7] Mitchell Guttman, Ido Amit, Manuel Garber, Courtney French, Michael F. Lin, David Feldser, Maite Huarte, Or Zuk, Bryce W. Carey, John P. Cassady, Moran N. Cabili, Rudolf Jaenisch, Tarjei S. Mikkelsen, Tyler Jacks, Nir Hacohen, Bradley E. Bernstein, Manolis Kellis, Aviv Regev, John L. Rinn, and Eric S. Lander. Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. *Nature*, 458:223–227, 2009.

[8] Ahmad M. Khalil, Mitchell Guttman, Maite Huarte, Manuel Garber, Arjun Raj, Dianali Rivea Morales, Kelly Thomas, Aviva Presser, Bradley E. Bernstein, Alexander van Oudenaarden, Aviv Regev, Eric S. Lander, and John L. Rinn. Many human large intergenic noncoding RNAs associate with chromatin-modifying complexes and affect gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, 106:11667–11672, 2009.

[9] Corinne Chureau, Marine Prissette, Agnès Bourdet, Valérie Barbe, Laurence Cattolico, Louis Jones, André Eggen, Philip Avner, and Laurent Duret. Comparative sequence analysis of the X-inactivation center region in mouse, human, and bovine. *Genome research*, 12(6):894–908, June 2002. ISSN 1088-9051.

[10] Howard Y. Chang, Dimitry S. Nuyten, Julie B. Sneddon, Trevor Hastie, Robert Tibshirani, Therese Sørlie, Hongyue Dai, Yudong D. He, Laura J. van't Veer, Harry Bartelink, Matt van de Rijn, Patrick O. Brown, and Marc J. van de Vijver. Robustness, scalability, and integration of a wound-response gene expression signature in predicting breast cancer survival. 102:3738–3743, 2005.

[11] Miao-Chih C. Tsai, Robert C. Spitale, and Howard Y. Chang. Long intergenic noncoding RNAs: new links in cancer progression. *Cancer research*, 71:3–7, 2011.

[12] Rajnish A. Gupta, Nilay Shah, Kevin C. Wang, Jeewon Kim, Hugo M. Horlings, David J. Wong, Miao-Chih C. Tsai, Tiffany Hung, Pedram Argani, John L. Rinn, Yulei Wang, Pius Brzoska, Benjamin Kong, Rui Li, Robert B. West, Marc J. van de Vijver, Saraswati Sukumar, and Howard Y. Chang. Long non-coding RNA HOTAIR reprograms chromatin state to promote cancer metastasis. *Nature*, 464:1071–1076, 2010.

[13] Jeremy E. Wilusz, Hongjae Sunwoo, and David L. Spector. Long noncoding RNAs: functional surprises from the RNA world. *Genes & development*, 23:1494–1504, 2009.

[14] Kevin V.Morris, editor. *RNA and the Regulation of Gene Expression*. 2008.

[15] Nuno A. Faustino and Thomas A. Cooper. Pre-mRNA splicing and human disease. *Genes & Development*, 17(4):419–437, February 2003.

[16] Minna-Liisa and Karla M. Neugebauer. Long Noncoding RNAs Add Another Layer to Pre-mRNA Splicing Regulation. *Molecular Cell*, 39:833–834, 2010.

[17] Can Cenik, Adnan Derti, Joseph C. Mellor, Gabriel F. Berriz, and Frederick P. Roth. Genome-wide functional analysis of human 5' untranslated region introns. *Genome biology*, 11(3):R29+, March 2010.

[18] Hadas Keren, Galit Lev-Maor, and Gil Ast. Alternative splicing and evolution: diversification, exon definition and function. *Nature Reviews Genetics*, 11(5):345–355, April 2010.

[19] Sangeeta Chatterjee and Jayanta K. Pal. Role of 5'- and 3'-untranslated regions of mRNAs in human diseases. *Biology of the cell / under the auspices of the European Cell Biology Organization*, 101:251–262, 2009.

[20] G. Pesole, G. Grillo, A. Larizza, and S. Liuni. The untranslated regions of eukaryotic mRNAs: structure, function, evolution and bioinformatic tools for their analysis. *Briefings in bioinformatics*, 1(3):236–249, September 2000.

[21] Stellamarie Reamon-Buettner, Si-Hyen Cho, and Juergen Borlak. Mutations in the 3'-untranslated region of GATA4 as molecular hotspots for congenital heart disease (CHD). *BMC Medical Genetics*, 8, 2007.

[22] Kelsey C. Martin and Anne Ephrussi. mRNA localization: gene expression in the spatial dimension. *Cell*, 136(4):719–730, February 2009.

[23] Mathew W. Wright and Elspeth A. Bruford. Naming 'junk': Human non-protein coding rna (ncrna) gene nomenclature. pages 90–98, 2011 January.

[24] Chris P. Ponting, Peter L. Oliver, and Wolf Reik. Evolution and functions of long noncoding RNAs. 136:629–641, 2009.

[25] Ulf A. Ørom and Ramin Shiekhattar. Long non-coding RNAs and enhancers. *Current Opinion in Genetics & Development*, 21:194–198, 2011.

[26] J. D. WATSON and F. H. CRICK. Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, April 1953. ISSN 0028-0836.

[27] David H. Mathews. Revolutions in RNA secondary structure prediction. *Journal of molecular biology*, 359:526–532, 2006.

[28] A. Krpgh G. Mitchison R. Durbin, S. Eddy. *Biological sequence analysis Probabilistic models of proteins and nucleic*. 2006.

[29] Jan Gorodkin and Ivo L. Hofacker. From Structure Prediction to Genomic Screens for Novel Non-Coding RNAs. *PLoS Comput Biol*, 7:e1002100+, 2011.

[30] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic acids research*, 9(1):133–148, January 1981.

[31] J. S. Mccaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119, May 1990.

[32] Stephan Bernhart, Hakim Tafer, Ulrike Muckstein, Christoph Flamm, Peter Stadler, and Ivo Hofacker. Partition function and base pairing probabilities of RNA heterodimers. *Algorithms for Molecular Biology*, 1(1):3+, March 2006.

[33] I. L. Hofacker, B. Priwitzer, and P. F. Stadler. Prediction of locally stable RNA secondary structures for genome-wide surveys. *Bioinformatics*, 20:186–190, 2004.

[34] Rolf Backofen Essam A. Hady Steffen Heyne. Accuracy-based identification of local rna elements. Bachelor Thesis, September 2010. URL `http://www.bioinf.uni-freiburg.de//Lehre/Theses/BA_Essam_Abdel_Hady.pdf`.

[35] Donald R. Forsdyke. Calculation of folding energies of single-stranded nucleic acid sequence: Conceptual issues. 2007.

[36] Vivek Thakur, Samart Wanchana, Mercedes Xu, Richard Bruskiewich, William Quick, Axel Mosig, and Xin G. Zhu. Characterization of statistical features for plant microRNA prediction. *BMC Genomics*, 12(1):108+, 2011.

[37] Peter Clote, Fabrizio Ferré, Evangelos Kranakis, and Danny Krizanc. Structural RNA has lower folding energy than random RNA of the same dinucleotide frequency. *RNA (New York, N.Y.)*, 11(5):578–591, May 2005.

[38] Shiquan Wu and Xun Gu. Random shuffling permutations of nucleotides.

[39] URL `http://hmmer.janelia.org/`.

[40] Larry J. Stephens Murray R. Spiegel. *Schaum's ouTlines STATISTICS*.

[41] Hubert M. Blalock. *Social Statistics*.

[42] URL `http://www.mathwave.com/articles/extreme-value-distributions.html`.

[43] Jesper Ryden Klara Persson. Exponentiated gumbel distribution fror estimation of return levels of significant wave height. Journal 12, Uppsala University, Uppsala University, February 2010.

[44] Charles M. Grinstead J. Laurie Snell. *Introduction to PROBABILITY*. American Mathematical Society, 1997.

[45] . URL `http://www.physics.csbsju.edu/stats/KS-test.html`.

[46] . URL `http://www.math.nsysu.edu.tw/~lomn/homepage/class/92/kstest/kolmogorov.pdf`.

[47] Christel Weiß. *Basiswissen Medizinische Statistik*. 1999.

[48] Michael J. Crawley. *The R Book*. 2009.

[49] Peter Dalgaard. *Introductory Statistics with R*. 2008.

[50] In J. Myung. Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1):90–100, February 2003.

[51] Rob J. Hyndman and Yanan Fan. Sample Quantiles in Statistical Packages. *The American Statistician*, 50(4):361–365, 1996.

[52] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis (Use R).*

[53] Eric Bonnet, Jan Wuyts, Pierre RouzÃ©, and Yves Van de Peer. Evidence that microRNA precursors, unlike other non-coding RNAs, have lower folding free energies than random sequences. *Bioinformatics*, 20(17):2911–2917, November 2004.

[54] Elena Rivas and Sean R. Eddy. Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs. *Bioinformatics*, 16(7): 583–605, July 2000.

[55] Ulf A. Ørom, Thomas Derrien, Malte Beringer, Kiranmai Gumireddy, Alessandro Gardini, Giovanni Bussotti, Fan Lai, Matthias Zytnicki, Cedric Notredame, Qihong Huang, Roderic Guigo, and Ramin Shiekhattar. Long Noncoding RNAs with Enhancer-like Function in Human Cells. *Cell*, 143:46–58, 2010.

[56] Moran N. Cabili, Cole Trapnell, Loyal Goff, Magdalena Koziol, Barbara Tazon-Vega, Aviv Regev, and John L. Rinn. Integrative annotation of human large intergenic noncoding RNAs reveals global properties and specific subclasses. *Genes & Development*, 25:1915–1927, 2011.

[57] Walter Gilbert. Why genes in pieces? *Nature*, 271, 1978.

[58] Dan Gusfield. *Algorithms on Strings, Trees, and Sequences.* Cambridge University Press, Cambridge, 1997.

[59] Michiaki Hamada, Kengo Sato, and Kiyoshi Asai. Improving the accuracy of predicting secondary structure for aligned RNA sequences.

[60] Chunjiang He, Fang Zhou, Zhixiang Zuo, Hanhua Cheng, and Rongjia Zhou. A Global View of Cancer-Specific Transcript Variants by Subtractive Transcriptome-Wide Analysis. *PLoS ONE*, 4:e4732+, 2009.

[61] Ivo L. Hofacker. Vienna RNA secondary structure server. *Nucleic acids research*, 31:3429–3431, 2003.

[62] Ivo L. Hofacker, Martin Fekete, and Peter F. Stadler. Secondary structure prediction for aligned RNA sequences. *Journal of molecular biology*, 319:1059–1066, 2002.

[63] Ivo L. Hofacker, Walter Fontana, Peter F. Stadler, Sebastian L. Bonhoeffer, Manfred Tacker, and Peter Schuster. Fast Folding and Comparison of RNA Secondary Structures. *Monatsh. Chem.*, 125, 1994.

[64] Maite Huarte, Mitchell Guttman, David Feldser, Manuel Garber, Magdalena J. Koziol, Daniela Kenzelmann-Broz, Ahmad M. Khalil, Or Zuk, Ido Amit, Michal Rabani, Laura D. Attardi, Aviv Regev, Eric S. Lander, Tyler Jacks, and John L.

Rinn. A Large Intergenic Noncoding RNA Induced by p53 Mediates Global Gene Repression in the p53 Response. 142:409–419, 2010.

[65] Kathleen Marchal, Gert Thijs, Sigrid D. Keersmaecker, Pieter Monsieurs, Bart D. Moor, and Jos Vanderleyden. Genome-specific higher-order background models to improve motif detection. *Trends in Microbiology*, 11:61–66, 2003.

[66] Svetlana Ekisheva Mark Borodovsky. *Problems and solutions in biological sequence analysis.*

[67] John S. Mattick. Linc-ing Long Noncoding RNAs and Enhancer Function. *Developmental Cell*, 19:485–486, 2010.

[68] Tim R. Mercer, Marcel E. Dinger, and John S. Mattick. Long non-coding RNAs: insights into functions. *Nature Reviews Genetics*, 10:155–159, 2009.

[69] J. L. Oliver and A. Marín. A relationship between GC content and coding-sequence length. *Journal of molecular evolution*, 43(3):216–223, September 1996.

[70] Jonathan Pevsner. *Bioinformatics and functional genomics.*

[71] Joseph N. Hall with Randal L. Schwartz. *Effective Perl Programming Writing Better Programs With Perl.* 2005.