Statistical Modeling and Estimation Strategies for Repetitive and Noncoding Nucleic Acid Sequences

A DISSERTATION PRESENTED
BY
WILSON MCKERROW
TO
THE DEPARTMENT OF APPLIED MATHEMATICS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF
APPLIED MATHEMATICS

BROWN UNIVERSITY
PROVIDENCE, RHODE ISLAND
MAY 2018

Statistical Modeling and Estimation Strategies for Repetitive and Noncoding Nucleic Acid Sequences

Abstract

For most eukaryotic organisms, protein coding sequences are only a small portion of the genome. Much of the genome is made up of transposable elements, regulatory sequence and functional RNA molecules. Modern sequencing technologies show that these noncoding sequences are not merely "junk" DNA, but have important biological function. However the exact role of many of these sequences remains poorly understood. Probabilistic modeling and estimation provides a fruitful method for us to gain insight into this genomic dark matter. In the first part of this thesis, I will describe a model that makes it possible to apply next-generation sequencing data to transposable elements and other repetitive sequence. This method provides accurate predictions of RNA hyper-editing, genomic variation, and TE expression. The final chapter discusses the most informative basepair method, which provides insight into the Boltzmann ensemble of RNA secondary structure.

This dissertation by Wilson McKerrow is accepted in its present form by
the Department of Applied Mathematics as satisfying the dissertation requirement
for the degree of Doctor of Philosophy.

Date _____ _____

Charles E. Lawrence

Recommended to the Graduate Council

Date _____ _____

Robert Reenan
Brown University

Date _____ _____

Elie Bienenstock
Brown University

Approved by the Graduate Council

Date _____ _____

Andrew G. Campbell
Dean of the Graduate School

Wilson McKerrow

Department of Applied Mathematics
Brown University
Providence, RI

Phone: +1 (415) 971 6379
Email: wilson_wmckerrow@brown.edu

EDUCATION

Brown University, Providence, RI Sep 2012 - May 2018 (expected)
Ph.D. Applied Mathematics
· Advisor: Charles Lawrence

Haverford College, Haverford, PA Sep 2007 - May 2011
B.S. Mathematics

PUBLICATIONS

- WH McKerrow, YA Savva, A Rezaei, RA Reenan, CE Lawrence. Predicting RNA hyper-editing with a novel tool when unambiguous alignment is impossible. BMC Genomics. 2017; 18:522
- J Dvoř k, ST Mashiyama, M Sajid, S Braschi, M Delcroix, EL Schneider, WH McKerrow, M Bahgat, E Hansell, PC Babbitt, CS Craik, JH McKerrow, CR Caffrey. SmCL3, a Gastrodermal Cysteine Protease of the Human Blood Fluke Schistosoma mansoni. PLoS Neglected Tropical Diseases. 2009; 3:449

FORTHCOMING MANUSCRIPTS

- L Lin, WH McKerrow (co-first), B Richards, C Phonsom, CE Lawrence. Characterization and visualization of the Boltzmann landscape of RNA secondary structure via information theory. *Accepted at BMC Bioinformatics.*

SOFTWARE

- RepProfile: https://github.com/wmckerrow/RepProfile
  - Predicts RNA hyper-editing in transposable elements by calculating read alignment probabilities and preforming expectation maximization.
- MIBP: https://github.com/wmckerrow/MIBP

- Calculates the mutual information between basepairs under the nearest neighbor model for RNA secondary structure.
- Identifies key basepairs and shows their effect on the ensemble distribution.

ONGOING PROJECTS

Novel TE Reconstruction

- Many tools predict novel TE insertions from NGS reads, but provide no information about how the inserted sequence compares to other instances. By improving our RepProfile algorithm (https://github.com/wmckerrow/RepProfile) with a custom aligner that performs seeded forward/backward alignment sums, we are able to reconstruct the sequence of a novel insertion using simulated data.
- As part of an IGERT traineeship we are analyzing dual-seq data to understand the relationship between an amphipod species and a behavior altering trematode parasite.

TEACHING AND MENTORING

Brown University Su2013,Su2014
Mentoring Undergraduate research.

- Mentored undergraduate applying information theory to RNA secondary structure.
- Mentored undergraduates using NGS sequencing data to infer transposition history.

Brown University Fa2013,Fa2015,Sp2016,Sp2018
Teaching Assistant: Statistical Inference (I)

- First semester probability and statistics for applied math concentrators.

Brown University Sp2014
Teaching Assistant: Modern Applications in Probability and Statistics

- Combined graduate/undergraduate course overviewing computational methods for probability and statistics.

Brown University Fa2017
Teaching Assistant: Computational Probability and Statistics

- First semester course overviewing computational methods for probability and statistics.

FELLOWSHIPS

- Manning Fellowship, Brown University, AY 2012-2013.
- NSF IGERT Fellowship, Brown University, AYs 2013-2014,2014-2015,2016-2017.

REFERENCES

Charles Lawrence
charles_lawrence@brown.edu
Brown University,
Applied Mathematics

Robert Reenan
Robert_Reenan@brown.edu
Brown University,
Molecular Biology, Cell
Biology and Biochemistry

David Rand
David_Rand@brown.edu
Brown University,
Ecology and Evolutionary
Biology

Acknowledgments

I would first like to thank my advisor, Charles (Chip) Lawrence. Certainly none of this thesis would exist without his aid, mentorship and guidance. Chip has taught me not only probabilistic modeling and bayesian statistical methods, but also how to pose good research questions and then break them into manageable pieces.

I would also like to thank Robert Reenan, who has been a second mentor to me. Rob has provided me with the knowledge of molecule biology and Drosophila genetics that allows me to focus on problems that are not only mathematically challenging, but also biologically insightful. Our meetings have consistently filled me with renewed excitement about my research and new questions to pursue.

Thank you as well to my third reader, Elie Bienenstock.

This thesis and the research therein has benefited from the support, suggestions and aid provided by member of Chip's lab group. Thanks in particular to August Guang, my 2012 cohortee, for helping me at many stages and to William (Bill) Thompson for helping me write Python code. Thanks as well to the undergraduate researchers I have worked closely with: Chukiat Phonsom, Taesoo Kim and Dilum Aluthge.

This thesis benefits from sequence data and insights provided by Yiannis Savva. Without Yiannis's help the applications to real biological data would be severely limited.

Finally thank you to all my friends and family who have supported me through my graduate studies – especially my partner, Kate, and my close friend and office mate, Clark Bowman.

Contents

Listing of figures

0

Introduction

0.1 Biological sequences.

0.1.1 The basics of biological sequences

The central dogma of biology describes the flow of genetic information from DNA (the blueprint) to RNA (the messenger) to protein (the workhorse.) However as you will see in this thesis, there are many exceptions to this rule, from retrotransposon RNAs that are converted back into DNA to proteins that edit the RNA message to noncoding RNAs that are themselves the workhorse.

Nevertheless we will start the story with DNA (DeoxyriboNucleic Acid). DNA is a highly stable molecule constructed from four building blocks called nucleotides. Each nucleotide consists of a phosphate, a deoxyribose sugar and one of four nitrogenous bases. They are: adenine (abbreviated A), cytosine (C), guanine (G) and thymine (T). Nucleotides are

strung together linearly with the phosphate of one nucleotide attached to the sugar of the next nucleotide, forming a DNA molecule. A DNA molecule can be described by a single string on the four letter alphabet $\{A, C, G, T\}$, detailing which nitrogenous bases are present and what order they are in. The DNA string is written from 5' (the phosphate end) to 3' (the sugar end), reflecting the direction of RNA transcription. DNA molecules are rarely found in isolation, but instead are found intwined with a second DNA molecule – called its reverse complement – in the classic double helix structure. Each nucleotide has a complementary base depending on the number of hydrogen bonds it forms. A and T both form two hydrogen bonds and thus readily pair. Similarly, G and C form three hydrogen bonds, readily pairing together. Thus the reverse complement has nucleotide T, G, C or A attached to A, C, G or T respectively. Furthermore the 5' to 3' orientation of the pair is reversed. For example if one strand is $AATGCTA$ then the reverse complement would be $TAGCATT$. Inside a cell, the paired DNA strands are called a chromosome. In eukaryotes, most of the cellular DNA is contained in the nucleus with a small amount contained in the mitochondria. Taken together, the strings corresponding to all of an organism's chromosomes form the genome. While the genome does include multiple sequences, in this thesis, I will simplify notation by using a single index i ranging from 1 to n for all genomic positions.

The next stage of the central dogma is RNA (RiboNucleic Acid.) The molecular structures of RNA and DNA are very similar but with two primary differences. RNA nucleotides include a ribose sugar instead of a deoxyribose sugar and the nitrogenous base thymine (T) is replaced with uracil (U). Thus, as with DNA, an RNA molecule can be described by a string on a four letter alphabet: $\{A, C, G, U\}$. The structural differences between RNA and DNA mean that RNA is much less stable, rapidly degrading when it is outside of the cellular environment. RNA is transcribed from one of the strands of DNA by an RNA polymerase enzyme. The string corresponding to the transcribed RNA molecule is identical to the DNA it was transcribed from, with T bases replaced by U. Much of the machinery in a cell, especially in multicellular organisms, is devoted to making sure that the the right amount of the right sequence is transcribed at the right

time and in the right place. This control is called regulation.

The final step in the central dogma is protein. Proteins are not a focus of this thesis, but I will discuss them briefly for completeness. Many of the RNAs transcribed in the nucleus are messenger RNAs or mRNAs that indicate the sequence of a particular protein. Like RNA and DNA, proteins are chains of building blocks. However in the case of proteins there are 20 amino acid building blocks. Thus it takes three nucleotides, called a codon, to encode one of the 20 amino acids. Some codons are also so called stop codons that indicate the end of a protein. Because there are more possible codons (64) than items to encode (21) multiple codons encode the same information. As a result, some mutations will not change the protein sequence. These mutations are called synonymous mutations and tend to have a neutral effect on fitness. Mutations that change an amino acid codon into a stop codon are particularly deleterious. A cellular structure called the ribosome is responsible for translating an mRNA sequence into protein. Proteins then go on to perform much of the functional work in the cell.

0.1.2 Alignment

It is often useful to know how similar two sequences are. Sequence similarity can indicate evolutionary relatedness and similarity of function. As we will see in chapter 1, finding sequence similarity can also help us to piece together fragments of DNA or RNA. In some cases the similarity between sequences is obvious. Consider for example $AATGCTA$ and $AATCCTA$. If we line them up:

```
AATGCTA

AATCCTA
```

It is clear that they differ at exactly one position. However what if we want to compare $AATGCTA$ and $AATCTAG$? If we line these sequences up as before:

```
AATGCTA
AATCTAG
```

The two sequences are equal at less than half of the positions. However if we instead compare the sequences as follows:

```
AATGCTA-
AAT-CTAG
```

the sequences differ only by one deleted base (the G in position 4) and one inserted base (the G in position 7). Thus when comparing two sequences, we need to first add insertions and deletions (indels) to align them, and then we can count mismatches.

There is no single best way to decide whether one alignment implies more sequence similarity than another. A variety of alignment scoring mechanisms exist to rank alignments. Most standard schemes apply a large penalty for opening a new insertion or deletion (δ^{open}) and a smaller penalty for each additional base in the insertion or deletion (δ^{ext}). Matches and mismatches are scored with a match matrix, $M = \{m_{xy}\}$, where m_{xy} is the score for aligning x to y. The alignment score is then the sum of the scores for all the parts of the alignment. For example the score for the following alignment:

```
AATGCTA
AGT--TA
```

is

$$m_{AA} + m_{AG} + m_{TT} + \delta^{open} + \delta^{ext} + m_{TT} + m_{AA}$$

The best scoring alignment can be found using a dynamic programming algorithm. The Needleman-Wunsch algorithm[73] gives the best alignment between two entire sequences – called the global alignment. The Smith-Waterman algorithm[96] finds the best matching subsequences and gives the relevant subsequence alignment. This partial alignment is called a local alignment. These exact alignment algorithms run in $\mathcal{O}(mn)$ time, where m and n are the lengths of the sequences being aligned.

While this is much faster than a brute force approach, it is not fast enough for the "big data" applications that currently dominate sequence analysis. This has generated the need for approximate or "heuristic" alignment algorithms whose run time scales with the length of only one of the aligned sequences. The most famous heuristic alignment algorithm is BLAST (Basic Local Alignment Search Tool)[1]. BLAST makes it possible efficiently search a giant indexed database of sequences for local alignments to a query sequence. The query sequence is broken into pieces called seed sequences. BLAST first searches for exact matches to the seed sequences in the database. Because the database is pre-indexed, this search can be done very quickly. Once matches are found, the Smith-Waterman algorithm is used to extend those partial matches into local alignments. A similar strategy is used when aligning short read sequences (section 1.1.1) to a reference genome (next section). In these applications, 100s of millions of reads must be aligned to a reference genome that can be 100 of millions or even billions of nucleotides long, depending on the organism being studied. Popular tools for read alignment, such as Bowtie[52] and bwa[59], use the Burrows-Wheeler transform to index the genome and find seed matches with a small number of mistmatches.

0.1.3 The reference genome and variation

For model organisms such as the fruit fly *Drosophila melanogaster* as well as for humans (*Homo sapiens*) – and increasingly for many nonmodel organisms as well – one can use the published reference genome as a starting point when studying genomic DNA or transcribed RNA sequences. The reference genome includes the genomic sequences of all chro-

mosomes for an archetypal individual. While genomes do vary from individual to individual, they are far more similar than they are different. Thus if you want to learn the sequence of a particular individual, you need not resequence the entire genome (hundreds of millions of letters in the case of *Drosophila melanogaster* and billions in the case of *Homo sapiens*.) One need only learn the 1% or so of the genome that is different from the reference.

Variations from the reference genome are generally divided into two main types. The first type are position mutations in which some individuals have the reference letter in a given position, but other individuals have a different nucleotide at the position. These are called single nucleotide polymorphisms (SNPs). The other type are insertions and deletions, collectively referred to as indels. Sequence that is in some individuals but missing from the reference genome is called an insertion. Sequence that is present in the reference genome but missing from some individuals is called a deletion.

0.2 Transposable elements

0.2.1 What are Transposable elements?

Originally uncovered by Barbara McClintocks in her work on maize[67], transposable elements (TEs), also called transposons, are DNA sequences that are capable of moving or duplicating within the genome. TEs fall into two broad categories depending on whether or not they move through an RNA intermediate. DNA transposons encode a transposase enzyme that will cut the element out of the genome and insert it into a new site. While this generally results in a so-called "cut and paste", it can also result in a duplication of the transposable element if transposition occurs during chromosomal replication. RNA transposable elements or "retrotransposons" encode for a reverse transcriptase enzyme and move via an RNA intermediate. After RNA is transcribed from the element, reverse transcriptase copies the transcript back into DNA, which is then inserted into the genome. This results in a "copy and paste" mechanism.

Because transposable elements are capable of copying themselves, if unchecked, they will accumulate over time. At least 40-50% of the human genome is made up of transposable element or transposable element derived sequence[16]. In some plants, over 80% of genomic DNA is composed of transposable element derived sequence[56]. Over time, most TE copies have degraded to the point of no longer being able to mobilize themselves. However such "nonautonomous" elements can still recruit transposase or reverse transcriptase from intact ("autonomous") elements, allowing them to move and replicate. In humans it is believed that only one transposable element, the retrotransposon LINE1, remains autonomous, with 80-100 copies still encoding functional proteins[7]. However the proteins encoded by intact LINE1 elements are capable of copying degraded, nonautonomous LINE elements, SINE elements and SVA elements.

The presence of so much TE sequence across the genomes of most eukaryotic organism raises the question of how TEs originally developed. The evidence indicates that different types of TEs evolved from different endogenous or exogenous sequences at different times. Many retrotransposons – the so called LTR retrotransposons – derived their sequence from retroviruses[34]. When a cell is infected by a retrovirus, there is a chance that the DNA reverse transcribed from the viral genome is integrated into the host's genomic DNA. This is the process by which an exogenous retrovirus becomes an endogenous retrovirus and, if that sequence is passed on to an organisms offspring, a new TE is formed. Other TEs are derived from host sequence. For example the nonautonomous human Alu elements are derived from the 7SL subunit of the signal recognition particle (SRP)[36].

0.2.2 Transposon silencing

The presence of so many TE sequences leads to a host/parasite interaction between organisms and their transposable elements. In order to survive and "reproduce", transposable elements must copy themselves – especially in germline cells. However a rapid proliferation of TE sequences would likely be fatal to the host. This has lead to the evolution

of sophisticated mechanisms to control TEs[94].

Many TEs are suppressed (or silenced) by RNA interference (RNAi)[94]. RNAi was first discovered when it was observed that inserting double stranded mRNAs into the round worm *Caenorhabditis elegans* actually led to an decrease in protein translation[25]. Dicer proteins recognize double stranded RNA (dsRNA) and cut 21-30 nucleotide pieces out of the sequence. These short RNAs are called short interfering RNAs (siRNA). The siRNA molecules join with argonaute proteins which use the siRNA sequence as guide to cleave and destroy any RNA sequence that matches the siRNA. As I will discuss further in section 2.1 many transposable elements form dsRNA when transcribed, triggering their suppression by RNA interference.

Organisms also suppress TEs by preventing RNAs from being transcribed from TE sequence in the first place. This can be done by packing segments of genomic DNA around histone proteins so tightly that transcription factors and RNA polymerases cannot access these portions of the genome[94]. These inaccessible regions are called heterochromatin and are often full of TE sequence. Transcription can also be suppressed by DNA methylation[94]. Methyl groups are attached to DNA backbone, preventing the binding of transcription factors. This transcription suppression can be inherited epigenetically and is triggered when RNAs are being silenced by RNAi.

Animals use an additional method to protect their germline cells from TE insertion: the piwi pathway. The piwi pathway is very similar to RNAi, but instead of using siRNAs cut from dsRNA by dicer as a guide, piRNAs are transcribed and directly from the genome and then cut into 26-30 nucleotide pieces that serve as a guide for a special argonaute protein called aubergine[94].

0.2.3 Transposable elements and disease.

While the cell does have a range of mechanisms to limit transposon insertion in the germline, some transposition still occurs. If a transposon inserts into the exon of a

gene, proteins translated from that gene will be severely disrupted and likely destroyed completely. In many cases this result in an inviable organism, but in other cases this transposition will lead to the formation of a new heritable genetic disorder. The first such discovery were six independent insertions of LINE1 into coagulation factor VII – one of the proteins involved in the formation of blood clots[42]. Individuals with any one of these insertions carry an allele for hemophilia A. Since this discovery, 124 genetic disease alleles have been shown to arise from TE insertion[32].

While no direct causal link has been shown, TE activation is associated with certain types of cancer[7]. Control of LINE1 elements is lost in some forms of cancer, likely due to the loss of TE methylation. This includes 90% of breast, ovarian and pancreatic cancers, and about half of prostate, lung, esophageal and colon cancers. The widespread activation of LINE1 could potentially lead to insertions that disrupt cancer suppressing genes, clearing the path to malignancy. Indeed some examples of cancer suppression genes being damaged by TEs have been shown[7]. However it is not known how large a factor such insertions are in cancer development.

TE activation is also associated with neurodegenerative disease including amyotrophic lateral sclerosis (ALS) and fronto-temporal lobar degeneration (FTLD). In *Drosophila melangaster* Gypsy retrotransposon expression induces neurodegeneration. In human ALS patients, HERV-K proteins are expressed[61]. Both Gypsy and HERV-K are endogenous retroviruses. The relationship between neurodegenerative disease and transposable elements is still not fully understood. One hypothesis is that the uncontrolled insertion of TEs leads to DNA damage and cell death[48]. Another hypothesis is that cell death is caused by the toxicity of viral proteins synthesized from endogenous retroviral mRNAs[61]. The role of TEs in the brain is an area of growing interest. TE expression is also elevated in some stress disorders and is associated with alcoholism[86]. The observed activation of TEs in mammal neuronal development has also led researchers to hypothesize that TEs actually benefit the developing brain by providing genomic plasticity[23].

Transposable element activation is also associated with the aging process. The degrada-

tion of heterochromatin structure with age has been shown in many organisms, including *Drosophila melanogaster*, *Caenorhabditis elegans*, mouse and rat[108]. Because many TEs are normally tucked away in heterochromatin and thus suppressed, TE transcription increases with age. At least in fruit fly, this increase in TE expression leads to an actual increase in transposition, potentially causing DNA damage[109]. Calorie restriction, an intervention that is known to increase lifespan, slows the loss of heterochromatin and limits TE activation[109].

0.2.4 Transposable element domestication.

While much of the machinery surrounding TEs is defensive, it is only natural that organisms have evolved some methods to turn TEs to their benefit. This coopting is called "domestication." Telomerase provides a simple example of domestication. Because eukaryotic organisms have linear DNA chromosomes (as opposed to the circular shape used by bacteria) a little bit of the chromosome end is lost each time the chromosome is replicated. Thus it is necessary to add some non-essential sequence onto the ends of the chromosome between replications. This sequence is called the telomere. In the fruit fly *Drosophila melanogaster* three retrotransposons – HeT-A, TART and TAHRE – insert into the telomere, preventing the loss of essential sequence during chromosomal replication[112]. In contrast to *Drosophila melanogaster*, most eukaryotic organisms transcribe special telomerase RNAs that are reverse transcribed and then inserted into the telomeres, lengthening them. Both the telomerase sequence and the reverse transcriptase enzyme used are TE derived[86].

Telomerase is only one example of domesticated TE sequence. Many proteins make use of the DNA binding domain in transposase[24]. Another example of TE domestication is the adaptive immune system in which B and T-cell DNA rapidly mutates and rearranges using sequence derived from TEs[47]. Perhaps even more consequential are the novel regulatory frameworks that TE domestication allows for. As DNA transposons spread through the genome, they create a network of genomic sites all targeted by the same transposase

enzyme. Coopting this TE network into network of genes that are all expressed under similar conditions is much simpler than evolving such a network de novo. As the DNA elements continue to transpose, they will make potentially beneficial changes to the network[24].

1

A profile model for reads

1.1 Background

1.1.1 Illumina sequencing

In this chapter and the following chapter, we are primarily concerned with the analysis of high throughput short read sequencing data. There are several related technologies that generate this type of data, but Illumina sequencing currently dominates as it provides the highest quality data at the lowest cost. The sequencing experiment begins with a sample of DNA. In most cases this is either genomic DNA (DNAseq) or cDNA reverse transcribed from RNA (RNAseq). Mathematically, we think of the DNA sample as a set, $\mathcal{S}$, of strings of varying length over the DNA alphabet $\{a, c, g, t\}$. These strings are not directly observed. Our goal will be to use the sequence data to infer some property of the sample. To extract genomic DNA in a DNAseq experiment, proteins, RNA, and lipids are

dissolved, digested or otherwise removed from a biological sample, leaving only DNA. In an RNAseq experiment, the DNA is digested, but the RNA is not, leaving a sample of RNA molecules. Reverse transcriptase is then used to copy the RNA molecules into DNA. DNA and RNA can be extracted from tissue samples, from cell culture, or even in some cases from single cells.

Once the sample has been extracted it is sheared into fragments, and the ends of the fragments are sequenced. Typical fragment lengths are 300-500 basepairs. If we saw the sample as a set of strings over $\{a, c, g, t\}$, then the fragments are a set of substrings of those original strings. The fragments are then washed over a flow cell where ℓ nucleotides (letters) are read from each end of many fragments in parallel. For the current generation of sequencing machines, ℓ is typically 100, but it is smaller for older machines and will likely continue to grow as the technology improves. For a typical run, the number of fragments read may exceed 100 million. Thus the result of the sequencing experiment is a set of about 100 million pairs of ℓ long strings over the alphabet $\{a, c, g, t\}$. I will use R to refer to this read set. The read strings in R are substrings of strings from the sample set $\mathcal{S}$. Reads strings are subject to occasional nucleotide substitution errors. After filtering out low quality reads, errors are typically less than 1 in 1000.

While there are algorithms that infer the sample set $\mathcal{S}$ directly from the read set R, this "de-novo assembly" problem is challenging in unique sequence and all but impossible for repeats. To simplify the problem we assume that the sample sequences are all drawn from a single distribution that I will call the genome profile, denoted G. Our goal is then to infer the genome profile. In the case of DNAseq, the sample sequences are simply the chromosome sequences. While we do expect genomic DNA to differ between individuals of the same species, we expect the genomes of individuals to be far more similar than different. Thus we need only infer how the sampled sequences vary from the reference genome. In an RNAseq experiment, the sequences in $\mathcal{S}$ are transcribed from genomic DNA. Thus we expect the sequences to be copied from the reference genome, again possibly with some variation. Note that in the case of RNAseq, some parts of the genome are transcribed

more than other parts, so we need to take into account relative expression levels, which we will denote with X. Therefore samples from our genome profile distribution G, will look like substrings of the reference genome, potentially with a few SNP or indel variations.

If we assume that the set of sample sequences is large then we can model this process by drawing independent reads according to the following process:

1. A sample sequence $S \in \mathcal{S}$ is drawn from the genome profile G or (G, X) in the case of RNAseq.
2. A random fragment (subsequence) of S is chosen.
3. The first and last ℓ letters of the fragment are read.

Because a new $S \in \mathcal{S}$ is sampled for each read and only 2ℓ positions of S are sequenced, we can further simplify the read sampling to the following:

1. 2 sets of ℓ genomic coordinates are sampled, denoted by A_j, where j is the index of the read being sampled.
2. At each of the 2ℓ positions, a letter, $R_j(k)$, is sampled from $\{a, c, g, t\}$, where k is the index of the position on read j that is being sampled.

The process by which read data is generated is shown in figure 1.1.

1.1.2 The ambiguous alignment challenge

While we are able to observe the sequences at the ends of a fragment, we do not know the genomic coordinates that those letters were sampled from. I will refer to these genomic coordinates as the read alignment. If the letters at each genomic position were from an independent uniform distribution over $\{a, c, g, t\}$, then the sequence of each read would easily provide enough information to uniquely identify its genomic origin. For $2\ell = 200$ even if read errors and genomic variations prevent us from distinguishing reads with 10 or fewer differences, there are still

$$\frac{4^{200}}{\binom{200}{10}4^{10}} \approx 1.1 \times 10^{98}$$

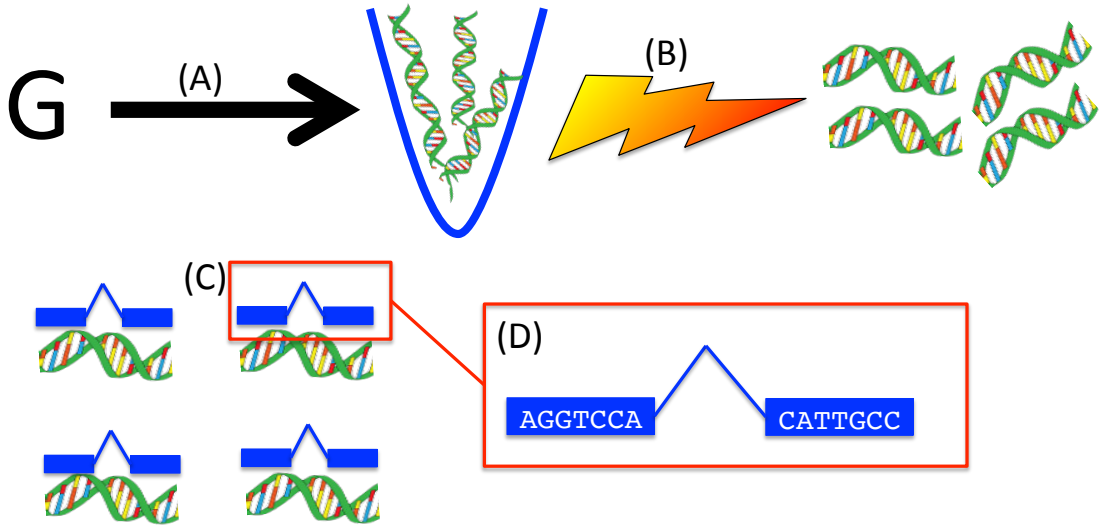

Figure 1.1: Read data generation. (A) We begin with DNA sequences sampled from the genome profile. (B) The sample sequences are sheared into fragments. (C) ℓ (usually 100) bases are sequenced from each end of each fragment. (D) The result is a set of pairs of ℓ longs sequences.

possible reads. This number is far larger than any genome, and indeed exceeds the estimated number of atoms in the universe. Thus it is extremely unlikely that two similar reads could be from non-overlapping sets of genomic coordinates. (There can still be some uncertainty about the exact alignment coordinates if there are indel variations near the edges of the read.) However the sequences of transposable element copies are anything but independent. Because the copies come from duplication events, they are likely to be exactly identical or at least nearly identical. Thus if we have a read that aligns well to a transposable element, we may not be able to tell which copy it was sampled from. If a set of reads indicates some variation in a transposable element, it will be difficult to tell which particular copy actually has the variation.

1.1.3 Strategy: Generative modeling and MAP estimation.

It is possible to infer which TE copy a read was sampled from when the read overlaps informative positions. Informative positions are positions where a particular repeat copy differs from other copies. Even if a read does not overlap informative positions in the ref-

erence genome, it may be possible to infer sample specific informative positions and then use those positions to align the read. For example suppose there is an unknown SNP near the edge of a repeat copy. If enough reads are sequenced we would expect to find some reads for which one read end aligns to the SNP and one read end aligns to unique sequence outside the repeat. These reads would indicate not only that there is a SNP, but also tell us which particular repeat copy that SNP is in. We can then use that SNP as a new informative position to help us align reads that overlap the SNP but fall completely within the repeat sequence.

In general it is the case that if an aligned read provides information about a SNP or other variation, then the variation will provide information back about the correct alignment of the read. Thus even if we cannot find the correct alignment for a read when we consider it in isolation, it may have a single most likely alignment when we consider an entire joint model that includes both read alignments and variations. In the following sections, I will define a probability distribution over reads, alignments and variations. Then I will show how we can use expectation maximization (EM) to find the set of variations that (at least locally) maximizes the posterior probability given the observed read sequences.

1.2 A model for read sequences

1.2.1 Glossary of variables and parameters

Here are the key random variables for this section:

1. $R = R_1, \ldots, R_j, \ldots, R_m$ are the read sequences. $R_j(k)$ is the k^{th} letter of read j.
2. $A = A_1, \ldots, A_j, \ldots, A_m$ are the alignments of each read. $A_j(k)$ is the aligned position of the k^{th} letter of read j.
3. $G = G_1, \ldots, G_i, \ldots, G_n$ is the genome profile. $G_i(x)$ is the probability of sequencing nucleotide x at position i.
4. $X = X_1, \ldots, X_q, \ldots, X_r$ is expression level. X_q is the probability that a given read came from repeat q.

5. $\delta^{open}, \delta^{ext}, \iota^{open}, \iota^{ext}$ are the indel parameters. δ^{open} is the probability of starting a new deletion. δ^{ext} is the probability of extending a deletion by an additional position. ι^{open} is the probability of starting a new insertion. ι^{ext} is the probability of extending an insertion for an additional position. Note that in section 1.4, the indel parameters will be allowed to depend on genomic position. An index if i will indicate the probability of starting or extending an indel directly after position i.

6. $H = H_1, \ldots, H_q, \ldots, H_r$ are the hyper parameters. They tell us which variations we expect to see in a given repeat and how frequently they will occur.

7. $T = T_1, \ldots, T_i, \ldots, T_n$ are the position types. Position types such as SNP tell us how the sequenced reads are likely to differ from the reference genome at that position.

1.2.2 Sampling aligned positions

In the sampling method described at the end of section 1.1.1, the first step is to sample 2ℓ aligned positions, $A_j = A_j(1 : 2\ell)$. First, a repeat is sampled from the expression distribution, $X = (X_1 \ldots, X_q, \ldots, X_r)$, where X_q is the normalized expression level for repeat q. For DNAseq, X will be a uniform distribution, but for RNAseq, X is allowed to vary. Then a starting position is chosen uniformly at random within the repeat and flanking sequencing. In most cases, the sequenced positions are in consecutive order, but when an indel occurs, either additional letters are inserted into a read or some genomic coordinates are not sequenced. For $1 < k \leq \ell$, the next position on the read is the next position in the genome (i.e. $A_j(k+1) = A_j(k)+1$) if there is not an insertion or a deletion after position $A_j(k)$. This occurs with probability $1 - \delta^{open} - \iota^{open}$. With probability δ^{open}, some genomic positions are deleted: $A_j(k+1) = A_j(k) + Y + 1$, where Y is a geometric random variable with mean $1/\delta^{ext}$. Finally with probability ι^{open}, some read letters are inserted: $A_j(k + Y + 1) = A_j(k + 1)$ where Y is a geometric random variable with mean $1/\iota^{ext}$. Letters at the inserted positions – $R_j(k + 1 : k + Y)$ – are sampled independently from the uniform distribution. The geometric distribution is chosen for computational efficiency, but in section 1.4 we will consider non-homogenous indel parameters. However

we must still retain a Markovian structure for computational feasibility. Once the first ℓ aligned positions have been sampled, Z genomic positions are skipped between the two read ends. Z is sampled from a normal distribution with mean μ_{inner} and variance σ^2_{inner}. Thus, $A_j(\ell+1) = A_j(\ell) + Z$. Finally for $\ell+1 < k \leq 2\ell$ we proceed as with $1 < k \leq \ell$. Figure 1.2 shows how insertions and deletions affect A_j. This yields the following conditional distribution for A:

$$P(A|X) = \prod_{j=1}^{m} (\delta^{open})^{(\#\delta_j)} (\iota^{open})^{(\#\iota_j)} \left(\delta^{ext}\right)^{(n\delta_j - \#\delta_j)} \left(\iota^{ext}\right)^{(n\iota_j - \#\iota_j)} \phi^*(A_j(\ell+1) - A_j(\ell)) X_{A_j} \tag{1.1}$$

$$\stackrel{def}{=} \prod_{j=1}^{m} h(A_j) X_{A_j} \tag{1.2}$$

where ϕ^* is the normal pdf with mean μ_{inner} and variance σ^2_{inner}, $\#\delta_j$ is the number of deletions in read j, $\#\iota_j$ is the number of insertions in read j, $n\delta_j$ is the total number of deleted positions, and $n\iota_j$ is the total number of inserted letters. If we group together the X_q terms, we get the following conditional probability:

$$P(A|X) = h(A) \prod_{q=1}^{r} X_q^{V_q} \tag{1.3}$$

where V_q is a function of $A = A_{1:m}$ that counts the number of reads aligned to repeat q.

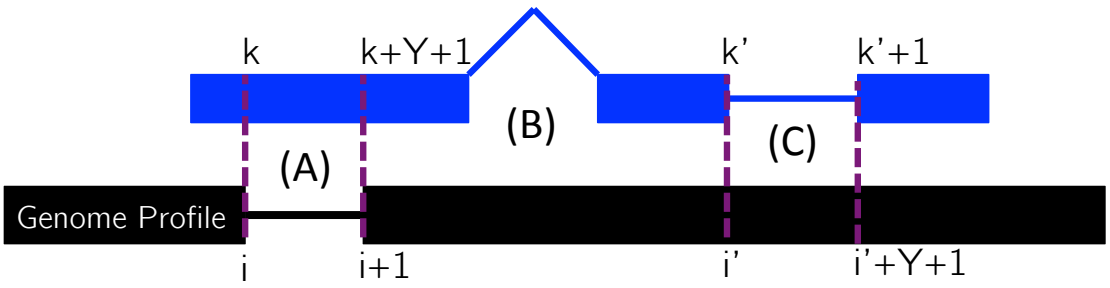

Figure 1.2: Alignment of a paired end read to a genome profile. (A) When there is an insertion in the read of length Y, the genome profile index increments by 1, but the read index increments by $Y+1$. (B) A section of the genome profile is skipped between the read ends. (C) When there is a deletion in the read of length Y, the read index increments by 1, but the genome profile increments by $Y+1$.

1.2.3 Sampling letters for aligned reads

Once the aligned positions have been sampled, the second step is to sample nucleotide letters at those aligned positions. We assume that read positions are independent. However this independence assumption still provides a good approximation and is necessary for efficient computation. Therefore each aligned position can be sampled from the distribution $G_{A_j(k)}$, which gives the probability of sequencing each letter at position $A_j(k)$. Thus the conditional probability of the read set is:

$$P(R|A,G) = \prod_{j=1}^{m}\prod_{k=1}^{\ell} G_{A_j(k)}(R_j(k)) \tag{1.4}$$

$$\stackrel{def}{=} \prod_{j} G_{A_j}(R_j) \tag{1.5}$$

It will be useful to rearrange this product so it is over positions instead of over reads:

$$P(R|A,G) = \prod_{i=1}^{n}\prod_{x\in\{a,c,g,t\}} G_i(x)^{U_i(x)} \tag{1.6}$$

where $U_i(x)$ is the number of times nucleotide x is aligned at position i. Note that U is a function of A and R.

1.2.4 A prior on G

We model the genome profile, G, using a hierarchical prior. At the top, are the discrete valued hyper parameters: $H = (H_1, \ldots, H_q, \ldots, H_r)$. The H variables operate on the level of an entire repeat, and allow the sequence of some repeat copies to be more variable than others. In some applications H will be constant, but it will become particularly important when we discuss hyper-editing, where some repeat copies have a large number of A to G variations and other repeats copies have few or none. Each H variable is drawn independently.

At the second level we have the position type, $T = (T_1, \ldots, T_i, \ldots, T_n)$ – also discrete valued. The position type tells us how we expect the genome profile at a given position to differ from the reference genome. For example if the reference nucleotide is a, and the position type is an a/c biallelic SNP, then we would expect the profile to be a mixture of a and c at that position. The probability of each position type depends on the value of H for the relevant repeat. Position types are sampled independently given H. Each position type defines a Dirichlet distribution. The genome profile at position i is then sampled from the Dirichlet defined by T_i. Figure 1.3 shows the prior on genome profiles. Putting it all together, we have the following distribution on H, T, G:

$$P(H,T,G) = \prod_q P(H_q) \prod_i P(T_i|H_{q(i)}) \frac{1}{B(\alpha_{T_i})} \prod_{x \in \{a,c,g,t\}} G_i(x)^{\alpha_{T_i}(x)-1} \tag{1.7}$$

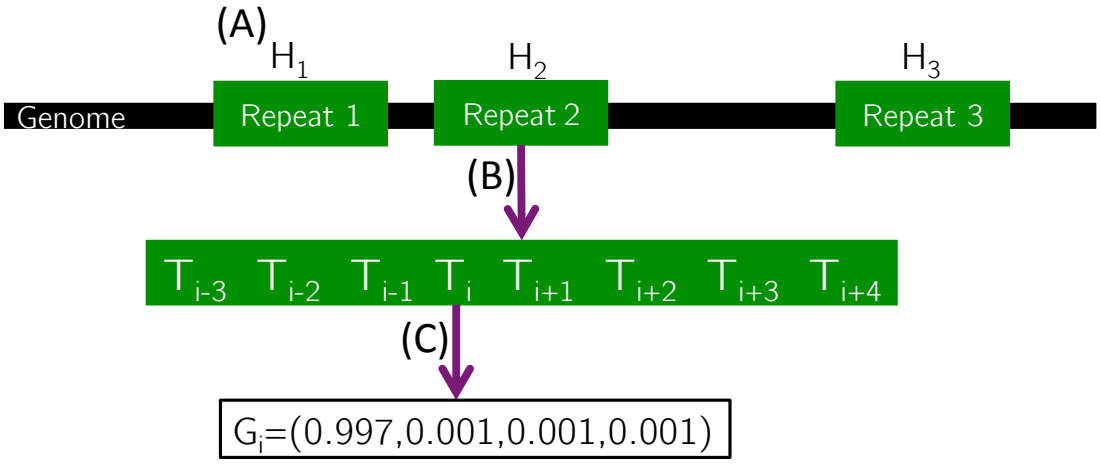

Figure 1.3: The prior over the genome profile, G. (A) A repeat type is sampled for each repeat, indicating the type and amount of variation that is expected in that repeat. (B) Variation types are sampled for each position of each repeat, depending on the corresponding repeat type. (C) Genome profile distributions over $\{a, c, g, t\}$ are sampled at each position, depending on the variation at that position.

1.2.5 The full joint distribution

Putting a uniform prior on X and combining equations 1.3, 1.6 and 1.7, we get the following joint distribution over all of the model variables:

$$\begin{aligned} P(X, A, R, G, H, T) &\propto P(A|X)P(R|A, G)P(H, T, G) \\ &= h(A) \prod_q P(H_q) X_q^{V_q} \prod_i \frac{P(T_i|H_{q(i)})}{B(\alpha_{T_i})} \prod_{x \in \{a,c,g,t\}} G_i(x)^{U_i(x)+\alpha_{T_i}(x)-1} \end{aligned}$$

1.2.6 Pitfalls for this model

The joint model described above includes independence assumptions that are necessary for efficient computation, but are not always true biologically. First, we assume that we can draw read letters from genome profile independently. However because children inherit whole chromosomes from their parents, variations tend to travel together. Parental chromosomes are recombined during the crossover step of meiosis, but only a few cross overs occur per chromosome, so the mixing is not complete. Thus variations that are nearby in genomic space tend to travel together. We also observe dependence between edited positions in section 2.1.3. Furthermore, some variations span multiple genomic positions, contradicting the assumption of position independence. However as long as we have enough informative positions – either learned or in the reference genome – this assumption will not prevent accurate estimation.

The model also assumes that reads are drawn independently. This is approximation is good when the number of fragments in the sample is much larger than the number of reads sequenced. However because the fragments are duplicated before sequencing, if there are too few fragments, the same read may be sequenced multiple times. Fortunately, read sets can be prescreened to see if have a large number of so called PCR duplicates. Such datasets may not be appropriate for the method described in this thesis.

1.3 Expection Maximization

We now have observed read data, R, and unknown random variables A, H, T, G and X. H, T, G and X are biological values of potential interest. H tells how which repeats are most variable and how they vary. T tells us about specific variations, and G tells how prevalent those variations are in the sequence population. The correct alignments, A is a property of the data, so we are only interested in it to the extent that it provides information about the other random variables. Thus we would like to estimate H, T, G and X given the reads, R. We can do this by maximum a posteriori (MAP) estimation. Our goal is then to calculate the following:

$$\begin{aligned}\hat{H}, \hat{T}, \hat{G}, \hat{X} &= \arg\max_{H,T,G,X} P(X, G, H, T | R = r) \\ &= \arg\max_{H,T,G,X} \sum_{A} P(X, A, G, H, T, R = r)\end{aligned}$$

While it is not feasible to maximize this quantity directly, expectation maximization (EM) provides a method to maximize over some random variables, (H, T, G, X), while marginalizing over another random variable, A.

To apply expectation maximization we need to perform the following iteration:

$$\begin{aligned}H^{(t+1)}, T^{(t+1)}, G^{(t+1)}, X^{(t+1)} &= \arg\max_{H,T,G,X} \mathbb{E}_A \left[\log P(X, A, G, H, T, R) | R, H^{(t)}, G^{(t)}, T^{(t)}, X^{(t)} \right] \\ &\stackrel{def}{=} \mathbb{E}_{A|t} \left[\log P(X, A, G, H, T, R) \right]\end{aligned}$$

To calculate the right hand side, we first need to simplify it. First,

$$\begin{aligned}\log P(X, A, G, H, T, R) = \log h(A) + \sum_q \log H_q + \sum_q V_q \log X_q + \sum_i P(T_i | H_{q(i)}) \\ - \sum_i \log B(\alpha_{T_i}) + \sum_i \sum_{x \in \{a,c,g,t\}} (U_i(x) + \alpha_{T_i}(x) - 1) \log G_i(x)\end{aligned}$$

We can then drop any term that does not depend on (H, T, G, X) as it will not be relevant to the argmax. Furthermore we can remove any terms that do not depend on A from the expectation yielding:

$$\mathbb{E}_{A|t}\left[\log P(X, A, G, H, T, R)\right] = f(A) + \sum_q \log H_q + \sum_q \mathbb{E}_{A|t}[V_q] \log X_q + \sum_i P(T_i|H_{q(i)}) - \sum_i \log B(\alpha_{T_i}) + \sum_i \sum_{x \in \{a,c,g,t\}} (\mathbb{E}_{A|t}[U_i(x)] + \alpha_{T_i}(x) - 1) \log G_i(x)$$

Maximizing over H, T, G, X yields:

$$\max_X \left[\sum_q \mathbb{E}_{A|t}[V_q] \log X_q\right] + \max_{H_q} \left[\sum_q \log H_q + \sum_i \max_{T_i} \left[P(T_i|H_{q(i)}) - \log B(\alpha_{T_i}) + \max_{G_i} \left[\sum_{x \in \{a,c,g,t\}} (\mathbb{E}_{A|t}[U_i(x)] + \alpha_{T_i}(x) - 1) \log G_i(x)\right]\right]\right]$$

The maximization over X is a multinomial MLE, so

$$\begin{aligned} X^{(t+1)} &= \arg\max_X \sum_q \mathbb{E}_{A|t}[V_q] \log X_q \\ &= \frac{\mathbb{E}_{A|t}[V_q]}{\sum \mathbb{E}_{A|t}[V_q]} \end{aligned}$$

The maximization over G_i is also a multinomial MLE, but it also depends on T, so we can let:

$$\begin{aligned} \hat{G}_{T,i} &= \arg\max_{G_i} \sum_{x \in \{a,c,g,t\}} (\mathbb{E}_{A|t}[U_i(x)] + \alpha_{T_i}(x) - 1) \log G_i(x) \\ &= \frac{\mathbb{E}_{A|t}[U_i(x)] + \alpha_{T_i}(x) - 1}{\sum \mathbb{E}_{A|t}[U_i(x)] + \alpha_{T_i}(x) - 1} \end{aligned}$$

Because T_i is discrete and assuming only a few values are allowed, we can then calculate

its max for a given $H_{q(i)}$:

$$\hat{T}_{H_{q(i)},i} = \arg\max_{T_i} \left[P(T_i|H_{q(i)}) - \log B(\alpha_{T_i}) + \sum_{x\in\{a,c,g,t\}} (\mathbb{E}_{A|t}[U_i(x)] + \alpha_{T_i}(x) - 1) \log \hat{G}_{T,i}(x) \right]$$
$$\stackrel{def}{=} \arg\max_{T_i} f_i(H_{q(i)}, T_i)$$

Finally, again because H is discrete and can only take on a few values, we can calculate the maximum:

$$H_q^{(t+1)} = \arg\max_{H_q} \log H_q + \sum_{i:q(i)=q} f_i(H_q, T_i)$$

We can then plug H and the T into the formulas for $\hat{T}$ and $\hat{G}$ respectively to get:

$$T_i^{(t+1)} = \hat{T}_{H_{q(i)}^{(t+1)},i}$$

and

$$G_i^{(t+1)} = \hat{G}_{T_i^{(t+1)},i}$$

Now it remains to estimate $\mathbb{E}_{A|t}[V]$ and $\mathbb{E}_{A|t}[U]$. These two are calculated in exactly the same way. The calculate $\mathbb{E}_{A|t}[U]$ simply replace V with U in the following calculation:

$$
\begin{aligned}
\mathbb{E}_{A|t}[V] &= \sum_j \mathbb{E}[V(A_j)] \\
&= \sum_j \sum_{A_j} V(A_j) P(A_j|R_j, G^{(t)}, X^{(t)}) \\
&= \sum_j \sum_{A_j} V(A_j) \frac{P(R_j|A_j, G^{(t)}) P(A_j|X^{(t)})}{\sum_{A_j} P(R_j|A_j, G^{(t)}) P(A_j|X^{(t)})} \\
&= \sum_j \sum_{A_j} V(A_j) \frac{P(R_j|A_j, G^{(t)}) P(A_j|X^{(t)})}{\sum_{A_j} P(R_j|A_j, G^{(t)}) P(A_j|X^{(t)})} \\
&= \sum_j \sum_{A_j} V(A_j) \frac{G^{(t)}(R_j) X_{q(A_j)}}{\sum_{A_j} G^{(t)}(R_j) X_{q(A_j)}}
\end{aligned}
$$

where $V(A_j)$ is the value of V if there is only a single read with alignment A_j. At least naively, summing over all possible alignments for each read is not feasible. A simple strategy that will work in most cases is to choose a smaller number of candidate alignments for each read and only sum over that set. Standard alignments tools such as bwa aln[60] are capable of finding all possible alignments to the reference genome up to a certain number of mismatches. In the next section, I will give a slower but more careful way to approximate the sum over all alignments that is also capable of considering inhomogenous insertion and deletion probabilities.

1.4 Exact calculation of PE alignment probabilities

1.4.1 A Profile HMM model for paired end reads

Equations 1.1 and 1.4 closely resemble the profile HMM model. The profile HMM is a non-homogenous Hidden Markov Model with three hidden states: match, insert and delete. We start in the match state with index $i = 1$. In the match state, a letter is drawn from G_i, and the index proceeds to $i + 1$. In the insert state, a random letter is output and the index remains the same. In the delete state, the index advances to $i + 1$ without

outputting any letter. The process terminates when the index reaches n. The transition matrix for the Markov chain is as follows:

	match	insert	delete
match	$1 - \delta_i^{open} - \iota_i^{open}$	ι_i^{open}	δ_i^{open}
insert	$1 - \iota_i^{ext}$	ι_i^{ext}	0
delete	$1 - \delta_i^{ext}$	0	δ_i^{ext}

The model used here to sample paired end reads, differs in three small ways. Firstly, instead starting at $i = 1$, the process starts at $i = I_0$. The probability that $I_0 = i$ is proportional to $X_{q(i)}p_i$ where $X_{q(i)}$ is the expression level at position i and p_i is calculated recursively as follows to account for the fact that the read is less likely to start at a position that has a high probability of being deleted:

$$p_i = p_{i-1}\delta_{i-1}^{(open)} + (1 - p_{i-1})(1 - \delta_{i-1}^{(ext)})$$

This probability, p_i, is then the probability that a given position is deleted, calculated from a two state Markov model with states: match and delete. Secondly instead of terminating when $i = n$, it terminates when 2ℓ letters have been output. Finally, we must account for the unsequenced positions in the middle of the fragment. We do this by adding m silent match, insert, and delete states. After outputting ℓ letters the state transitions to match(1), insert(1) or delete(1) according to the above transition matrix. In these silent states instead of outputting a letter, there is a p probability that we move the next silent match or insert state (e.g. from match(1) to match(2)). However for the standard profile HMM algorithms to apply we must increase k when transitioning into or out of an insert state. While this limits the learning of long inserts, in most cases we are more interested in deletions. We then have the following transition matrix:

	match(k)	match(k+1)	delete(k)	insert(k+1)
match(k)	$(1-p)(1-\delta_i^{open}-\iota_i^{open})$	$p(1-\delta_i^{open}-\iota_i^{open})$	δ_i^{open}	ι_i^{open}
insert(k)	0	$(1-\iota_i^{ext})$	0	ι_i^{ext}
delete(k)	$(1-p)(1-\delta_i^{ext})$	$p(1-\delta_i^{ext})$	δ_i^{ext}	0

Once we have passed through all of the silent interior states, instead of transitioning to match(k+1) or insert(k+1) we transition back to the standard match and insert states to output the final ℓ letters. This process yields paired end reads with a number of unsequenced interior positions that is distributed as a negative binomial with parameters m and p. The number of silent states m and the transition probability p can be chosen so that the negative binomial distribution approximates the desired normal distribution.

1.4.2 The forward and backward algorithms.

Taken together, the forward and backward algorithms allow us to calculate the marginal probability that the process is in state x at index i after j letters have been output (including the emission from state x if it is not silent.) This allows the calculation of $E_{A|t}[U]$ and $E_{A|t}[V]$ in $\mathcal{O}(mn)$ time, where n is the total number of genomic positions and m is the number reads.

Let $S_k = (\mathtt{x}, i, j)$ indicate that at the k^{th} step, the chain is in state $\mathtt{x}$, the index is i and there have been exactly j emissions, including the emission at step k (if $\mathtt{x}$ is not a silent state.) Furthermore, let $S = \{S_k\}$ be the set of all such triples visited. The forward sum calculates the following:

$$f_{ij}(\mathtt{x}) = P\left(R_{1:j}, (\mathtt{x}, i, j) \in S\right)$$

This quantity is calculated recursively by summing over all possible values $(\mathtt{y}, i', j')$ that

can proceed $(\mathtt{x}, i, j)$. When $\mathtt{x}$ is not a silent state, we have the following recursion:

$$\begin{aligned}
f_{ij}(\mathtt{x}) &= P\left(R_{1:j}, (\mathtt{x}, i, j) \in S\right) \\
&= \sum_k P\left(R_{j:i-1}, S_k = (\mathtt{x}, i, j)\right) \\
&= \sum_k \sum_{\mathtt{y}} P\left(R_{1:j}, S_{k-1} = (\mathtt{y}, i', j-1)\right) P(S_k = (\mathtt{x}, i, j) | S_{k-1} = (\mathtt{y}, i', j-1) P(R_j | S_k = (\mathtt{x}, i, j)) \\
&= \sum_{\mathtt{y}} T(\mathtt{x}, \mathtt{y}) P(\mathtt{x}, i, j \to R_j) \sum_k P\left(R_{1:j}, S_{k-1} = (\mathtt{y}, i', j-1)\right) \\
&= P(\mathtt{x}, i, j \to R_j) \sum_{\mathtt{y}} f_{i',j-1}(\mathtt{y}) T_{i',j-1}(\mathtt{x}, \mathtt{y})
\end{aligned}$$

Where i' is i if $\mathtt{x}$ is an insert state and $i-1$ otherwise, $P(\mathtt{x}, i, j \to R_j)$ is the probability that R_j is the j^{th} letter output if $(\mathtt{x}, i, j) \in S$ and $T_{i,j}(\mathtt{y}, \mathtt{x})$ is the probability of transitioning from $\mathtt{y}$ to $\mathtt{x}$ when the index is i and j letters have been emitted. When $\mathtt{x}$ is a silent state, there is no emission probability and j does not advance, so the recursion is:

$$f_{ij}(\mathtt{x}) = \sum_{\mathtt{y}} f_{i',j}(\mathtt{y}) T_{i',j}(\mathtt{x}, \mathtt{y})$$

Correspondingly, the backward sum calculates

$$P(R_{j+1:2\ell} | (\mathtt{x}, i, j) \in S)$$

by summing across the possible triples $(i', j', \mathtt{y})$ that might follow $(i, j, \mathtt{x})$:

$$\begin{aligned}
b_{ij}(\mathtt{x}) &= P(R_{j+1:2\ell}|(\mathtt{x}, i, j) \in S) \\
&= \sum_{\mathtt{y} \notin \text{silent}} P(R_{j+2:2\ell}|(\mathtt{y}, i', j+1) \in S) P((\mathtt{y}, i', j+1) \in S|(\mathtt{x}, i, j) \in S) P(R_{j+1}|(\mathtt{y}, i', j+1) \in S) \\
&+ \sum_{\mathtt{y} \in \text{silent}} P(R_{j+1:2\ell}|(\mathtt{y}, i', j) \in S) P((\mathtt{y}, i', j) \in S|(\mathtt{x}, i, j) \in S) \\
&= \sum_{\mathtt{y} \notin \text{silent}} b_{i',j+1}(\mathtt{y}) T_{i',j+1}(\mathtt{x}, \mathtt{y}) P(\mathtt{y}, i', j+1 \to R_{j+1}) + \sum_{\mathtt{y} \in \text{silent}} b_{i',j}(\mathtt{y}) T_{i',j}(\mathtt{x}, \mathtt{y})
\end{aligned}$$

where i' is i if $\mathtt{y}$ is an insert state and $i + 1$ otherwise. The equality $P((\mathtt{y}, i', j) \in S|(\mathtt{x}, i, j) \in S) = T(\mathtt{x}, \mathtt{y})$ follows from the fact that no triple that can be reached in one step can also be reached in two or more steps.

We can then calculate $E_{A|t}[U]$ as follows:

$$\begin{aligned}
E[U_i^x(R_j)] &= \sum_{k: r_k = x} P(S_{ik} = \mathtt{m}|R_{1:2\ell} = r_{1:2\ell}) \\
&= \sum_{k: r_k = x} \frac{f_{ik}(\mathtt{m}) b_{ik}(\mathtt{m})}{\sum_{\mathtt{x} \in \{\mathtt{m}, \mathtt{d}, \mathtt{i}\}} f_{i,2\ell}(\mathtt{x})}
\end{aligned}$$

Where $\mathtt{m}$, $\mathtt{d}$, and $\mathtt{i}$ are the match, delete and insert states respectively.

Unfortunately while the forward-backward algorithm does provide significant improvement over a brute force strategy, it is still too slow for large next-generation sequencing datasets. We can however get an accurate estimate of the forward and backward sums that is computationally feasible by using a seeded alignment strategy. For a given read instead of calculating $f_{ij}(\mathtt{x})$ and $b_{ij}(\mathtt{x})$ for all genomic coordinates i, we first find positions in the genome where part of the read exactly matches the most likely sequence given the genome profile. We then use those matches to find a smaller set of coordinates to calculate the forward and backward sums at. We assume that the forward and backward sums are approximately 0 everywhere else.

1.4.3 Learning indel parameters

For HMMs, the Baum-Welch algorithm is used to estimate the transition parameters using EM. At each EM step, the probability of transitioning from state $\mathbb{x}$ to state $\mathbb{y}$ is set to be the ratio between the expected number of times that the $\mathbb{x} \rightarrow \mathbb{y}$ transition occurred over the expected number of times that process entered state $\mathbb{x}$[20]. Thus at step $t+1$,

$$T_i(\mathbb{x}, \mathbb{y}) = \frac{\sum_k \sum_j P_{A|t}((i,j,\mathbb{x}) \in S, (i',j',\mathbb{y}) \in S) | R_{1:2\ell}}{P_{A|t}((i,j,\mathbb{x}) \in S | R_{1:2\ell})}$$

where $\mathbb{x}$ and $\mathbb{y}$ are each one of: all match states, all delete states or all insert states. Thus,

$$\begin{aligned}
\delta_i^{open} &= T_i(\mathbb{m}, \mathbb{d}) \\
\delta_i^{ext} &= T_i(\mathbb{d}, \mathbb{d}) \\
\iota_i^{open} &= T_i(\mathbb{m}, \mathbb{i}) \\
\iota_i^{ext} &= T_i(\mathbb{i}, \mathbb{i}).
\end{aligned}$$

2

Applications

2.1 RNA hyper-editing

2.1.1 Background

While RNA is only transcribed as a single stranded molecule, it can form double stranded RNA (dsRNA) if the strand is self-complementary or if both strands of DNA are transcribed. Transposable elements provide a large source of self-complementary RNA molecules that form dsRNA. For some repeats, such as FB4_DM in *Drosophila melanogaster*, the TE sequence itself is self-complementary. Other repeats are common enough that they appear nearby, but on opposite strands. If both copies are transcribed together then a self-complementary RNA molecule will be formed.

RNA editing occurs when adenosine deaminase acting on RNA (ADAR) enzymes bind to a double stranded RNA target and chemically modify adenosine bases by inserting a wa-

ter molecule and removing an ammonia molecule. The result is that adenosine (A) bases are converted to inosine (I). However inosine is recognized by the ribosome and other cellular machinery as guanosine (G), so for our purposes, we can think of RNA editing as causing A to G changes. When ADAR binds to a long, perfect dsRNA molecule it can convert as many as 50% of the adenosines in the double stranded region into inosine[74]. This high density editing is called hyper-editing. The high concentration of nucleotide changes makes hyper-editing very well suited to the profile/EM method described in this thesis. Each of the many editing sites represents a potential informative position that can be learned and then subsequently used to improve our alignment estimation.

Because dsRNA is involved in many pathways, often with detrimental effect, understanding its interaction with ADAR is of particular biological interest. Long, perfect dsRNA stimulates innate immunity, regulates gene transcription, and has been implicated in a variety of neurological and autoimmune disorders[76,93]. The fate of dsRNA depends on its interaction with RNA binding proteins. Possible fates include cleavage by dicer, leading to gene silencing[4], suppression by TDP-43 related proteins[91], which have been been implicated in neurodegenerative disease[11], and hyper-editing by ADAR enzymes.

Hyper-editing of endogenous RNAs was first reported in human ALU elements, a class of transposable elements comprising about 10% of the human genome sequence and numbering over one million copies[54]. As next-generation sequencing has become cheaper, new studies, using new sequencing methods and analyses, have increased the number of known editing sites and expanded our understanding of ADAR activity[89,83,101,85,107].

However, the ability of these studies to find hyper-editing in long, perfect dsRNA and to accurately estimate the level of editing at a given hyper-edited position is limited, because they must discard reads that have no best alignment to a reference genome or risk widespread false positive predictions. By its nature, long dsRNA, which consists of a sequence followed by its reverse complement, is repetitive. The self-complementarity of dsRNA ensures that a read originating from the interior of such a molecule will align equally well on both the forward and reverse strand. To make matters worse, dsRNA is

most likely to appear when repetitive elements are present, forming when two copies occur nearby but in opposite orientation or within certain self-complementary sequences. Thus, reads originating from within a long, perfect duplex are unlikely to have a single best alignment. Therefore, methods that discard reads with ambiguous alignment cannot provide a full picture of hyper-editing.

2.1.2 Tailoring the model

RNA hyper-editing is predicted using the joint model described in section 1.2 and the EM algorithm described in section 1.3. For each repeat, the repeat type, H, can take on three values: hyper-edited on the forward strand, hyper-edited on the reverse strand or not hyper-edited. *A priori*, each repeat has a 1% chance of being hyper-edited on either strand and a 98% chance of not being hyper-edited. There are five position types: SNP1, SNP2, SNP3, edited and none. Positions with type SNP1 have $\alpha = (1.0, 1.0, 0.01, 0.01)$ where the first coordinate is the reference base, the second is the next one alphabetically (with a following t), etc. This yields genome profiles that have a mixture of the reference base and the next base alphabetically. SNP2 and SNP3 have similar α values but yield a mixture of the reference base and the second and third following bases respectively. No matter the value of H, each position has a 0.5% change of being SNP2 and 0.25% for each of SNP1 and SNP3. These values represent the fact that transition mutations are more frequent than transversions[106]. If the repeat is hyper-edited on the forward strand, each A has a 49.5% chance of being edited. For a repeat that is hyper-edited on the reverse strand, each T has a 49.5% chance of being edited. Edited positions have the same α value as SNP2, yielding genome profiles that are a mixture of the reference base (A/T) and the edited base (G/C). Position that are none have $\alpha = (10.0, 0.01, 0.01, 0.01)$, yielding genome profiles with almost all the probability on the reference base. To prevent EM from getting stuck, we force $G_i(x) \geq 0.001$ for all i and x.

Because we are primarily interested in A to G (T to C) changes, we can approximate the sum over all alignments by considering all alignments that have any number of A/G

(T/C) mismatches but no more than 4 other mismatches on each end. Such a set of alignments can be provided by standard read alignment software. We use bwa aln[60].

Because EM is only guaranteed to find a local maximum rather than the global maximum, it makes sense to try a range of initial conditions. For this application we start with a single initial condition – no SNPs, no edits and uniform expression – but once EM converges we create new initial conditions by systematically removing editing from each repeat one at a time to see if we can find a better estimate. If a better maxima is found, the process is repeated for the new estimate. Simulated data shows EM converging quickly, so we assume that EM has converged after 10 steps.

2.1.3 Results

The above method for predicting RNA hyper-editing, which we called RepProfile, was tested in our recently published paper[68]. This work was a collaboration with members of the Reenan lab. Sequence data was generated by Yiannis Savva and Ali Rezaei. I performed all computation and simulation. RepProfile was used to predict hyper-editing in transposable elements from 2x100bp Illumina sequence reads from whole head *Drosophila melanogaster* RNA. RepProfile was run on each transposable element (TE) family in parallel. This included all repeats in the UCSC genome browser (`genome.ucsc.edu`) repeatmasker track except simple repeats, low complexity repeats, rRNA and satellites. The sequences considered total 29 megabases. Hyper-editing is not limited to TEs, but they are a common source of dsRNA, and RepProfile was designed to find hyper-editing in TEs. Repeat families that are a prefix of other families were merged. Thus, for example, PROTOP, PROTOP_A and PROTOP_B were considered together. Similarly the LTR and interior portions of RNA transposons were merged. RepProfile aligned 8.3 millions reads (totaling 1.66 gigabases) and predicted a total of 30,185 edit sites.

2.1.3.1 Simulations support accuracy of RepProfile

RepProfile and competing methods were tested against three different simulations of hyper-editing. In the first, reads were simulated from a hypothetical repeat family consisting of 24 identical copies of a random 1kb sequence: 20 isolated copies (10 in each orientation) and 2 oppositely oriented pairs that are simulated to be hyper-edited (one on each strand). In the second, reads were simulated from the reference FB4_DM repeat sequences, including editing only at the sites observed in our clone data (see below). In the third simulation, FB4_DM repeats are chosen at random to be hyper-edited. In both FB4_DM simulations, reads are simulated in proportion to observed expression levels.

Reads drawn from the hypothetical repeat family show that RepProfile is able to provide accurate alignment to highly repetitive sequence. Because the genome sequences of these repeats are identical, no read that falls entirely within a repeat has a unique alignment to the hypothetical repeat reference. Nevertheless, RepProfile is able to align 90% of reads that fall entirely within one of the hyper-edited duplexes to the learned profile with mapping quality 30+ (estimated probability of misalignment $\leq 0.1\%$). 99.9% of these reads are aligned correctly, allowing RepProfile to predict editing sites with high sensitivity and PPV (table 2.1).

In addition to RepProfile, we predicted edit sites using the method of Porath et al.[83] and by finding editing enriched regions (EERS)[107], either using all reads (+rep) or only using reads for which at least one end aligns uniquely (uniq). Table 2.1 shows sensitivity and positive predictive value (PPV) for each method applied to each simulation. RepProfile provides the highest sensitivity in all three simulations, while maintaining a PPV of 99% or above. RepProfile also provides accurate estimation of the editing level across all simulations (see figure 2.1).

In the clone simulation, editing occurs at A/G informative positions. Thus hyper-edited reads can align uniquely but incorrectly, explaining the diminished PPV when finding

EERs with unique reads. RepProfile's diminished sensitivity in the random hyper-editing simulation is due to the fact that many simulated edit sites occur in low coverage regions. As repeats accumulate mutations, unique alignments become possible, but the repeats also tend to lose their dsRNA structure. Because edit sites are not limited to dsRNA in the random hyper-editing simulation, more hyper-edited reads align uniquely, allowing the competing methods to perform better.

Exact Rep	Sens.	PPV	**Clone**	Sens.	PPV	**Random**	Sens.	PPV
RepProfile	0.94	0.99	RepProfile	0.95	1.0	RepProfile	0.87	0.99
Porath	0.07	1.0	Porath	0.13	0.97	Porath	0.50	0.98
EER uniq	0.36	0.99	EER uniq	0.55	0.82	EER uniq	0.70	0.98
EER +rep	0.90	0.14	EER +rep	0.91	0.42	EER +rep	0.86	0.74

Table 2.1: Sensitivity and PPV for RepProfile and competing methods in the three different simulations. Left: Simulation from 24 identical copies of a hypothetical repeat. Center: Simulation of editing predicted by clones. Right: Random hyper-editing of FB4_DM.

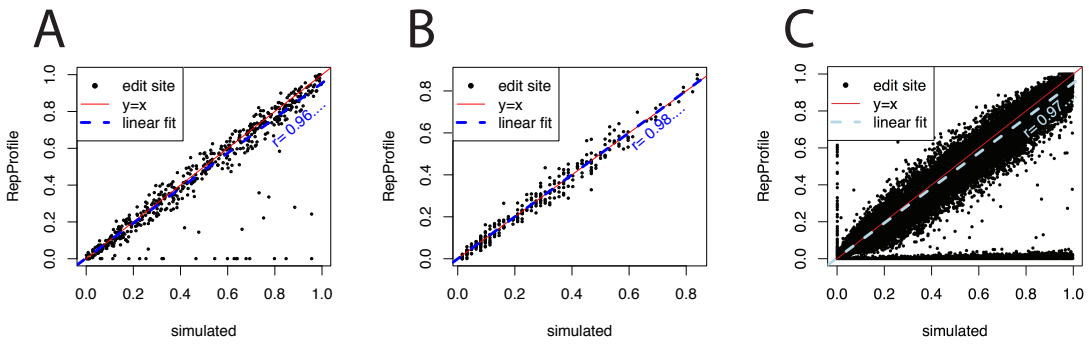

Figure 2.1: Agreement between simulated and predicted RNA-editing levels. Level of RNA-editing in simulation vs prediction for each position that is either simulated or predicted to be edited. A) Edits simulated from hypothetical exact repeat copies. B) Simulation using edits predicted by clone sequence. C) Simulation of hyper-editing at random FB4_DM. Because this simulation reflects the varied coverage of an actual RNA-seq experiment, many simulated edit sites are covered by few of no reads (points along x axis).

2.1.3.2 RepProfile predictions are validated by Sanger clones

A hyper-edited FB4_DM in the gene retinal degeneration A (rdgA) was chosen for validation. Not only is this repeat not unique, it is also internally repetitive, making

alignment particularly challenging. rdgA is expressed almost exclusively in the nervous system[33], as is dADAR protein. My collaborators Yiannis Savva and Ali Rezaei generated 62 validation sequences by cloning RT-PCR amplicons from the sequence spanning chrX:8,928,544-8,929,835 in the dm6 genome assembly (available from the UCSC genome browser: `genome.ucsc.edu`). The cloned sequences showed a deletion spanning chrX:8,928,786-8,929,278 and so the FB4_DM reference was updated to include this deletion.

The Sanger sequences confirm the pattern of hyper-editing predicted at this locus, with each clone displaying a distinct pattern of edited sites. Of the 322 adenosines in the clone region, editing is observed at 280 positions (87%) in at least 1 clone. Each clone is edited at an average of 76.7 adenosines (24%). RepProfile predicts editing at 269 of the 280 positions edited in the clone sequences (sensitivity = 96%). RepProfile also predicts editing at an additional 11 sites, yielding a PPV of 96%. As figure 2.2A shows, RepProfile also accurately predicts the level of editing at a given position. Among positions that show evidence for editing both in RepProfile and in the clones, RepProfile overestimates editing by 6%, with a standard deviation of 11%. Figure 2.2B provides a site-by-site comparison of predicted and validated editing.

Using reads for which at least one end aligns uniquely, the method of EERs[107] predicts only 31% of the cloned edit sites. All predictions made by the EER method are supported by the clone validation. Applying the method of Porath et al.[83] to our data yields 17 editing sites in this FB4_DM element, but fails to find any editing in the cloned region. These sensitivity estimates are lower than those estimated in simulation, indicating that there are additional alignment challenges not included in our simulation. Similarly results from[101] predict 11 editing sites in this element, but none in the cloned region. Rodriguez et al.[89] and Ramaswami et al.[85] fail to predict any editing in this FB4_DM. The lack of FB4_DM hyper-editing in these published lists is unsurprising as they all rely on the unambiguous alignment of short reads.

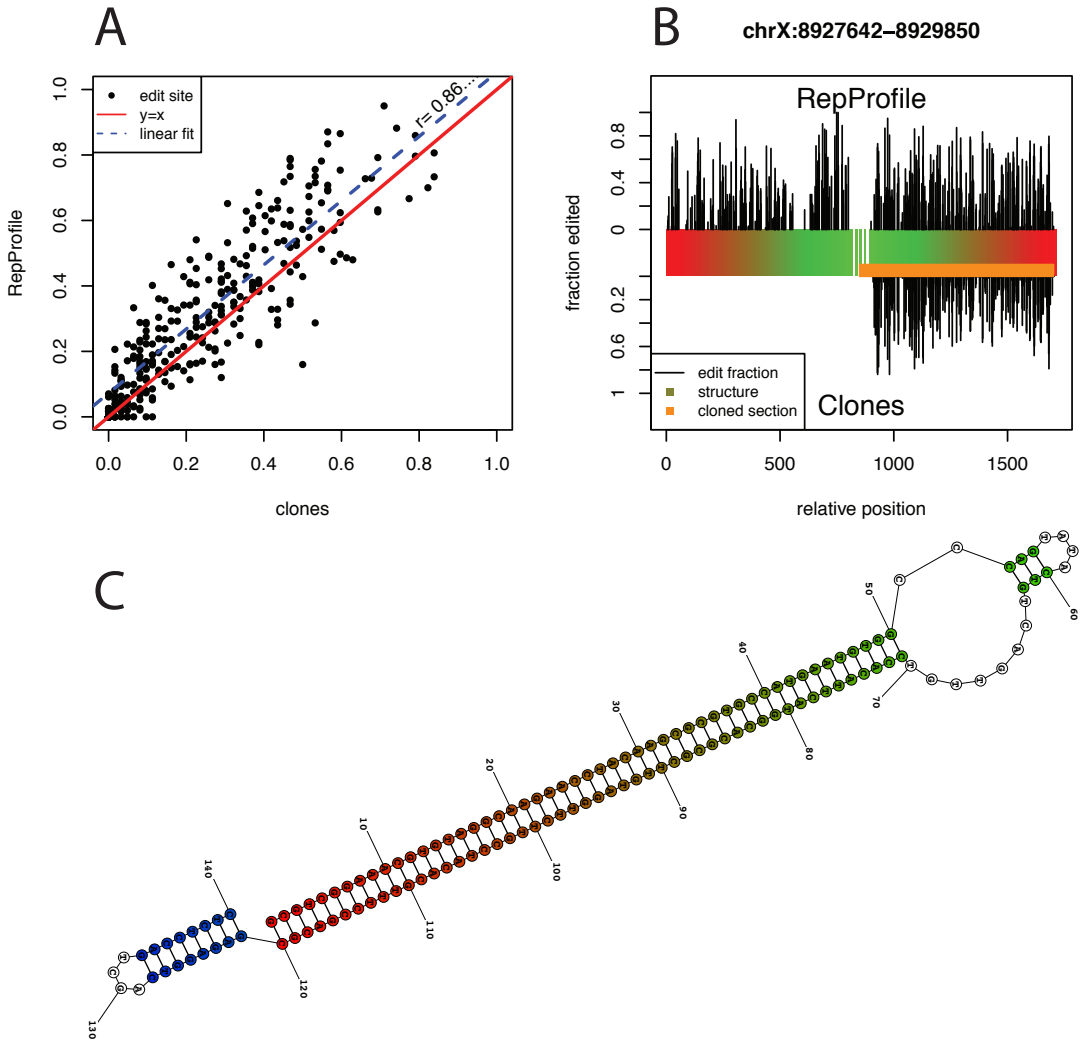

Figure 2.2: Agreement between RepProfile and clone validation. A) Fraction of editing predicted by clones vs fraction predicted by RepProfile. RepProfile and clones show good agreement in the rdgA FB4_DM element. B) RNA-editing predictions aligned to RNA secondary structure as predicted by RNAstructure[65,63]. Positions are colored so that aligned pairs are the same color. Unpaired positions are not colored. Both methods predict hyper-editing only in the helical portion of this repeat, consistent with the fact that ADAR is a dsRNA binding protein. C) Explanation of RNA structure coloring. A gradient from red to green to blue is stretched across the length of the sequence. Paired positions are colored according to the minimum of their position and the position they are paired to. Unpaired position are not colored.

2.1.3.3 FB4_DM repeats are highly hyper-edited

The FB4_DM sequence is almost entirely a perfect inverted repeat, which has the capacity, if transcribed, to fold back and form long, (nearly) perfect dsRNA (see supplementary figure). Thus, transcripts containing FB4_DM in pre-mRNA are potentially excellent ADAR hyper-editing substrates. Indeed, RepProfile predicts frequent hyper-editing of FB4_DM elements.

In addition to the element in rdgA described above, there are four other FB4_DM elements that are predicted to be hyper-edited by RepProfile in every local max founf by EM. These predictions appear in the genes no-long-nerve-cord (nolo), Pur-alpha, Maf1 and rolled (rl). Across these five repeats (including rdgA), 1,681 editing sites are predicted. Interestingly, like rdgA, these genes are known to be involved in proper cell-cell communication in the nervous system, particularly in the correct function of synapses[70,39,95,49]. Two (Pur-alpha[39] and rdgA[33]) are associated with neurodegeneration. Seven more FB4_DM (see table 2.2) are predicted to be edited by RepProfile in the best maxima found by EM. This brings the total number of predicted edit sites to 4,384. Five of these seven are also in genes that have been shown (or are predicted) to play roles in proper neuronal maintenance and function[105,98,55,72,69]. Two examples of FB4_DM hyper-editing are shown in figure 2.3.

There are seven additional genes containing FB4_DM elements that are predicted to form dsRNA, but are not predicted to be edited: CG11873, CG17600, CG42238, kek5, kirre, Pka-R1, vtd. Only two (Pka-R1 and vtd) of these genes are annotated with neuron-related GO terms in flybase[2] (as of January 1, 2017). It is possible that while these genes are edited in neurons, the edited reads are overwhelmed by transcription in cells that do not express ADAR. Alternatively, these RNA duplexes may be targeted by another dsRNA binding protein, making them unavailable to ADAR.

Position	Gene	all local maxima
chr2L:21705310-21707776	nolo	yes
chr2R:1075216-1076448	rl	yes
chr2R:1521733-1522952	Maf1	yes
chr4:560886-562920	Pur-alpha	yes
chrX:8927642-8929850	rdgA	yes
chr3L:4361048-4362834	Cip4	no
chr3L:8019759-8024110	nmo	no
chr3R:1971337-1972661	Myo81F	no
chr3R:21609064-21611636	inR	no
chr3R:22450601-22453179	CG34376	no
chrX:11645168-11648276	Ptp10D	no
chrX:2132105-2133551	ph-p	no

Table 2.2: FB4_DM that are predicted to be hyper-edited. Predictions with a yes in the last column are RepProfile's most confident predictions. If there is a no in the last column, RepProfile was able to align reads without predicting hyper-editing at this repeat, but not predicting hyper-editing at these repeats yielded a lower posterior probability.

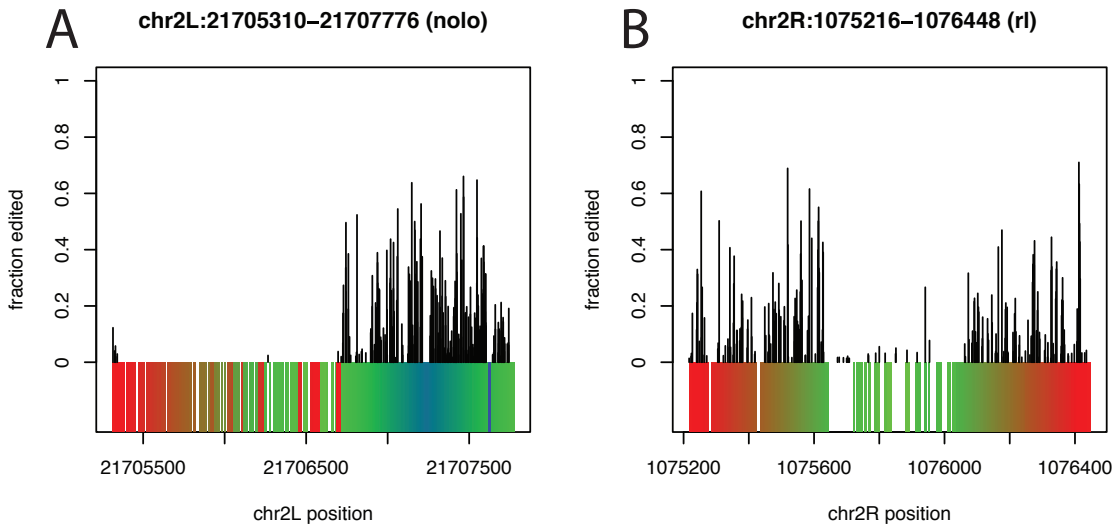

Figure 2.3: RepProfile predictions for two FB4_DM. Alignment of RepProfile predictions to RNAstructure[65,63] secondary structure predictions. Secondary structure is colored as in figure 2.2B/C. A) An FB4_DM element in the gene nolo. B) An FB4_DM element in the gene rolled (rl).

2.1.3.4 DNAREP1_DM repeats form imperfect helices that are partially edited

While shorter (up to 500 nt) the 5,802 DNAREP1_DM elements in the fly genome play an analogous role to that of ALU repeat elements in the human genome. As with ALU re-

peats, it is not uncommon for two DNAREP1_DM elements to be oriented in opposite directions in the same gene, or even to be opposite and adjacent. However DNAREP1_DM instances tend to be quite divergent, and so these DNAREP1_DM form imperfect helices that are edited to a lesser extent than FB4_DM. Figure 2.4 shows the hyper-editing and structural predictions for two pairs of DNAREP1_DM that are adjacent and opposite in orientation. Table 2.3 lists all the DNAREP1_DM that are predicted to be hyper-edited. RepProfile predicts 685 edited positions in DNAREP1_DM. Many of these genes are also relevant to the nervous system[8,50,111,46].

Position	Gene	dsRNA
chr2L:22996546-22999081	intergenic	adjacent +/-
chr2R:1305412-1314667	intergenic	unknown
chr2R:2245576-2248758	CG17684	adjacent +/-
chr2R:3107642-3109943	dpr21	same gene +/-
chr2R:4900040-4902165	CG44102	same gene +/-
chr2R:6823192-6825246	intergenic	unknown
chr3L:23034337-23037599	nrm	adjacent +/-
chr3L:24175963-24178195	Snap25	same gene +/-
chr3L:24183640-24186213	Snap25	same gene +/-
chr3L:24224373-24226438	snap25	same gene +/-
chr3L:25653710-25655922	CG45782	same gene +/-
chr3R:1453181-1455383	Myo81F	same gene +/-
chr3R:1593187-1595257	Myo81F	same gene +/-
chr3R:1627526-1629596	Myo81F	same gene +/-
chr3R:647512-650915	Myo81F	adjacent +/-
chr3R:888752-890953	Myo81F	same gene +/-
chr4:1147092-1149608	CG32017	adjacent +/-
chr4:859109-861284	CG11148	same gene +/-
chrX:142931-144983	tyn	same gene +/-

Table 2.3: DNAREP1_DM that are predicted to be hyper-edited. Adjacent +/- indicates that there is a DNAREP1_DM within 2kb that is in the opposite orientation. Same gene +/- indicates that there is a DNAREP1_DM in the same gene that is in the opposite orientation. Unknown means that neither of these two conditions apply.

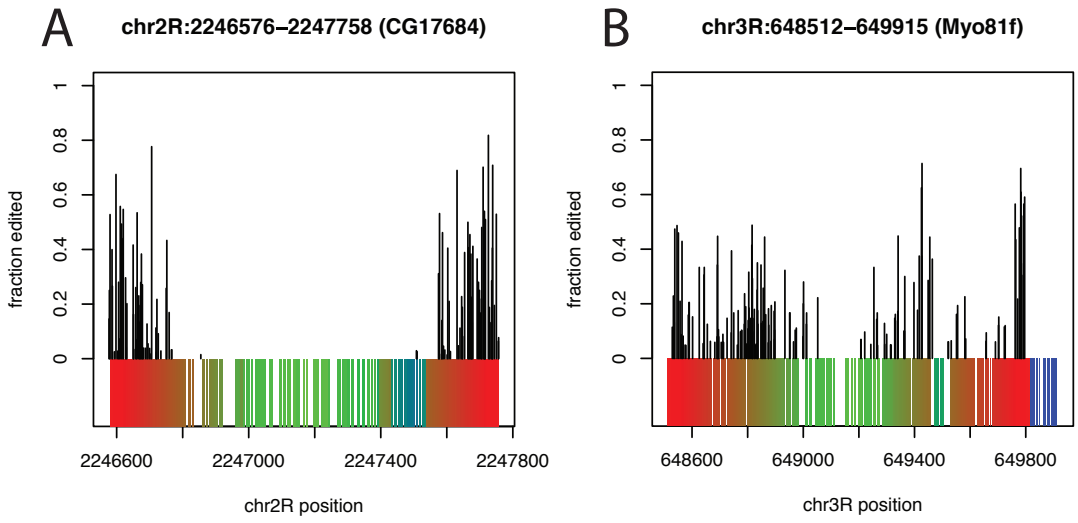

Figure 2.4: RepProfile predictions for two DNAREP1_DM. Alignment of RepProfile predictions to RNAstructure[65,63] secondary structure predictions. Secondary structure is colored as in figure 2.2B/C. A) A complementary pair of DNAREP1_DM elements in the gene CG17684. B) A complementary pair of DNAREP1_DM elements in the gene Myo-81f.

2.1.3.5 Alignments to PROTOP show hyper-editing of the previously described Hoppel killer element

The most probable solution found by RepProfile includes 38 hyper-edited PROTOP, PROTOP_A and PROTOP_B elements containing a total of 4,326 edit sites. Here we focus on the five copies that are predicted to be hyper-edited at all local maxima found by EM. All five are pairs of PROTOP(A/B) that are adjacent but opposite in orientation. These repeats contain a total of 973 predicted edit sites, 697 of which are in the Hoppel killer (Hok) element[92].

My collaborators previous work[92] demonstrated that dADAR proteins, as well as other dsRNA-binding proteins, localize to the Hok element in vivo, but, using methods that require unambiguous alignment, they only found a small number of editing sites in Hok[101]. However with RepProfile we are able to predict drastically more editing – a result that is more in line with the strong evidence of ADAR activity at this locus. Figure 2.5 shows predicted editing for Hok. Hok contains three highly similar PROTOP_A elements, so

there may be structural conformations other than the one illustrated in figure 2.5.

Position	Gene	dsRNA
chr3L:28002423-28003664	CG17514	adjacent +/-
chr3R:2420941-2423648	Myo81F	adjacent +/-
chr3R:3788494-3789401	intergenic	adjacent +/-
chr4:1257060-1260307	cadps	adjacent +/-
chr3L:24327899-24328803	nvd	adjacent +/-

Table 2.4: PROTOP(A/B) that are predicted to be hyper-edited at the highest confidence. Adjacent +/- indicates that there is a PROTOP(A/B) within 2kb that is in the opposite orientation. Same gene +/- indicates that there is a PROTOP(A/B) in the same gene that is in the opposite orientation. Unknown means that neither of these two conditions apply.

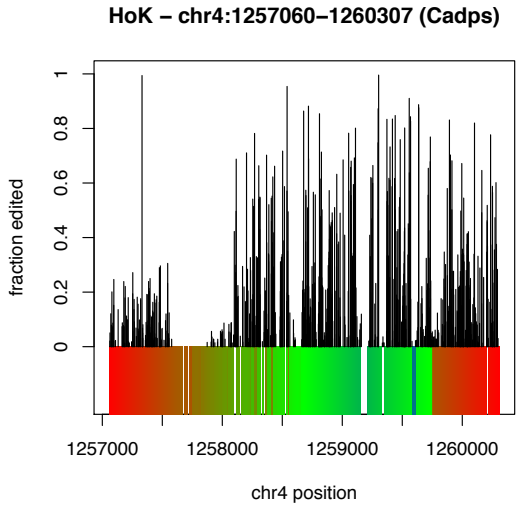

Figure 2.5: RepProfile predictions for the PROTOP_A elements that form Hok. Alignment of RepProfile predictions for Hok to the RNAstructure[65,63] secondary structure prediction. Secondary structure is colored as in figure 2.2B/C.

2.1.3.6 Hyper-editing in other repeats

RepProfile predicted 1,086 edit sites in BEL repeats. Predicted edit sites are in the ranges chr2R:18575852-18581982 (GEFmeson intron) and chrX:3415072-3421199 (CG12535 intron).

RepProfile predicted 2,084 edited sites in Copia repeats. However the only predictions that appear in every local maxima were 62 edit sites in a region (chr3R:1698684-1699615) of a Myo81f intron. Other predictions were in intergenic and poorly assembled sequence.

RepProfile predicted 2,693 edit sites in DMCR1A. More than half of these appear in every local maxima. 401 edit sites are predicted in chr3R:3450703-3456350, where there are DMCR1A elements on both strands. 1,145 edit sites are predicted in chr2R:1306411-1333449, where there are also DMCR1A on both strands. Neither of these regions overlap annotated genes.

RepProfile predicted 2,742 edit sites in FW_DM. More than 90% of these predictions are in three genomic regions. 1,223 edit sites are predicted in chr3R:660095-666560 which includes two adjacent oppositely oriented FW_DM repeats inside a Myo81f intron. Another 716 sites are predicted in Myo81f at chr3R:1606987-1623276, which includes FW_DM elements both strands. 558 sites are predicted in an rl intron: chr2R:1097537-1101863. Of these three regions, only chr3R:1606987-1623276 is predicted to be hyper-edited in every local maxima.

RepProfile predicted 3,435 edit sites in Gypsy family repeats. 13% are in the range spanning chr3R:3204734-3214018. This region is part of a Pzl intron and includes Gypsy repeats on both strands. 84% are in the region spanning chr2R:2808821-2839705 – part of a CG41378 intron that contains Gypsy repeats on both strands. Both regions are hyper-edited an all local maxima.

RepProfile predicted 1,316 edit sites in ROO. 96% of the predicted sites are in the genomic region chr3R:1596105-1626676 – a section of a Myo81f intron that contains ROO elements on both strands. This region is predicted to be hyper-edited in all local maxima.

RepProfile predicted 1,386 edit sites in ROVER. All edit sites are in the range chr2R:580416-591861, a section of a CG45781 intron that contains ROVER elements on

both strands. This region is predicted to be hyper-edited in all local maxima.

RepProfile predicted 1,332 edit sites in S_DM. Like FB4_DM, S_DM elements are self-complementarity. Predictions that appear in all local maxima are made in S_DM elements are listed in table 2.5

Position	Gene
chr3R:4683231-4684935	CG43427
chr3R:611446-613180	Myo81f
chr3R:1343650-1345383	Myo81f
chr3R:1642493-1642860	Myo81f
chr2R:2194058-2195781	CG17684
chr4:420027-421232	intergenic
chr3L:25608432-25610027	CG45782
chr3L:25414448-25416178	CG45782
chr3L:27280020-27281201	Dbp80
chr3L:27387332-27389063	Dbp80

Table 2.5: S_DM element predicted to be hyper edited.

2.1.3.7 ADAR edits in short runs

FB4_DM sequences contain long strings of consecutive adenosines, sometimes more than ten adenosines long. We analyzed the hyper-editing of consecutive adenosines on a read-by-read basis to understand how ADAR edits long, perfect dsRNA. When analyzing runs of edited adenosines that are followed by another base, we do not not know whether the run would have continued had there been more adenosines to edit. Thus we have (a discrete version of) the lifespan estimation from censored data problem analyzed by Kaplan and Meyer[41]. Runs of edited adenosines that are followed by a base that is not adenosine are considered to be censored, as the run may have continued were there more adenosines to edit. The hazard function,

$$P(\text{run length} = n | \text{run length} \geq n) \approx \frac{\#n \text{ long, uncensored}}{(\#n \text{ long, uncensored}) + (\#> n \text{ long})}$$

is shown in figure 2.6. Short runs are less likely to end than would be predicted from context alone. However as the run gets longer the probability that the run will end increases. This is consistent with the theory that as ADAR edits, it disrupts the dsRNA structure, introducing I-U mispairs and making future editing less likely.

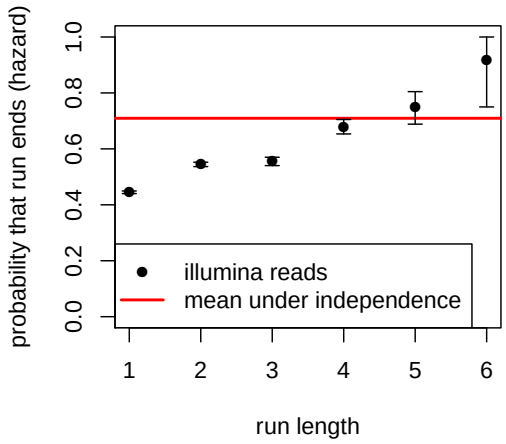

Figure 2.6: Estimated hazard function for length of editing run. The estimated probability that a run of FB4_DM editing will end after a given number of edited A's, with 95% binomial confidence intervals. The red line is the marginal probability that an A following an A at helix depth at least 25 will not be edited in a particular read. The estimates indicate that as a run of edits gets longer, it is more likely to end.

2.1.3.8 Short helices are rarely edited; the longest helices are edited most

To measure the affect of dsRNA structure on editing, we measure the length of helices in the five most confident FB4_DM hyper-editing predictions (table 2.2) and the five adjacent +/- DNAREP1_DM repeats that are predicted to be hyper-edited (table 2.3). Helices are allowed to include bulges of up to two bases on one or both sides of the helix, as small bulges have been shown not to interrupt hyper-editing[57]. Structure predictions are by RNAstructure[65,63].

We find that editing is rare in short helices and that it is most frequent in long helices.

RepProfile predicts editing at only 13% of adenosines in helices that are fewer than 18 basepairs long, but at 80% of adenosines in helices that are longer than 64 basepairs. This is consistent with in vitro evidence that ADAR does not bind to helices shorter than 15-20 basepairs and is most efficient when editing helices longer than 100 basepairs[74]. Figure 2.7 shows editing binned by helix size.

In the 819 basepair rdgA FB4_DM helix (the longest in this analysis), 84.1% of adenosine positions are predicted to be edited, but in individual transcripts only 27.7% of adenosines are edited on average. This editing of 27.7% of adenosines is less than the $50 - 60\%$ seen in vitro[74]. This difference could be because the editing reaction is not allowed to complete in vivo, because transcripts are sequenced before they are fully edited, or because some copies are sequenced from cells with low levels of ADAR.

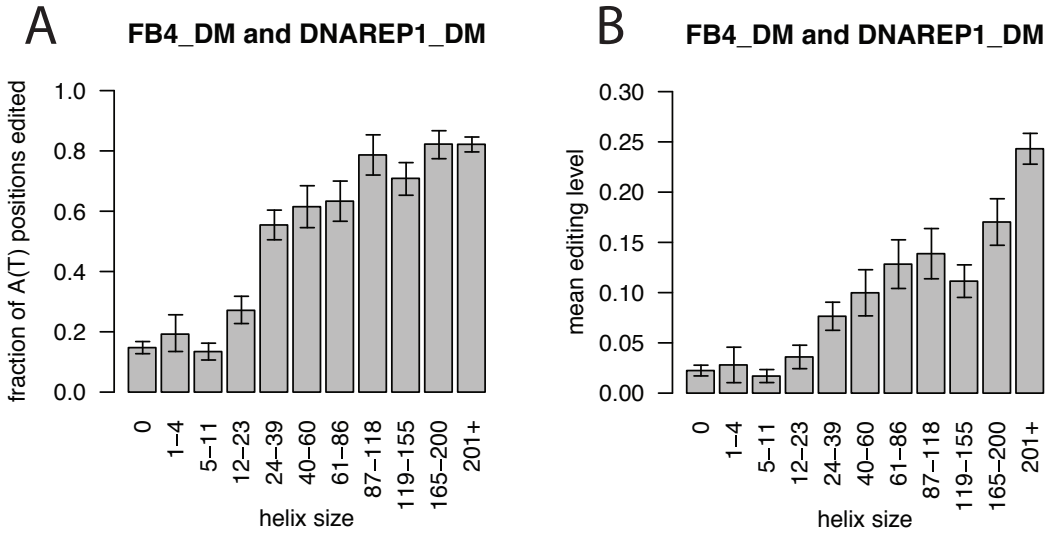

Figure 2.7: Probability that editing will be predicted as a function of helix size. A)The fraction of A positions at which any amount of editing is predicted, binned according the helix size, allowing bulges of of up to two bases for the five most confident FB4_DM hyper-editing predictions (table 2.2) and the five adjacent +/- DNAREP1_DM repeats that are predicted to be hyper-edited (table 2.3). Error bars are 95% binomial confidence intervals. B) Mean fraction of editing, binned according the helix size as in part A. Error bars are 95% normal approximation confidence intervals. Note that in both A and B, larger bins are supported by many positions but only a few helices, so confidence intervals may be overly tight. Structure predictions by RNAstructure[65,63].

2.1.3.9 The nucleotide context of our predictions reflects known ADAR preferences.

We investigated the effect of preceding and following bases on the fraction of editing at an adenosine (A) position. We consider only positions that are in the five FB4_DM predictions that are consistent across local maxima and are also at least 25 bases away from the nearest unpaired position – a total of 865 adenosines. Figure 2.8 shows the fraction of editing for sites in each three-base context.

The preceding base has a strong effect on the fraction of editing. Predicted edit sites following T are edited in the highest fraction of reads (mean = 0.35). Sites following A are slightly less likely to be edited (mean = 0.29). Sites following C are much less likely to be edited (mean = 0.08) and sites following G are rarely edited (mean = 0.03.) Each pairwise comparison has t-test BHY[3] FDR less than 0.003. Consistent with evidence that ADAR edits in runs, this 5' preference affects the following base: Adenosines preceded by AA or TA are edited more often than adenosines preceded by CA or GA (pairwise FDRs all less than 0.015.)

The following base has a smaller effect on the fraction of editing. Adenosines followed by G are edited most often (mean=0.35), followed by A (mean=0.26), C (mean = 0.24) and T (mean = 0.21.) However the only statistically significant result is that adenosines followed by G are more likely to be edited (p value = 0.0018.) Our results for 3' and 5' base preferences agree with those found in previous studies[82,58,45,22].

2.1.3.10 Run time per RepProfile step scales with the number of candidate alignments; the number of steps depends on the amount of editing.

For most repeats, calculating the probability of each candidate alignment in each E step forms a bottleneck. Thus the time to complete a single EM step scales with the number

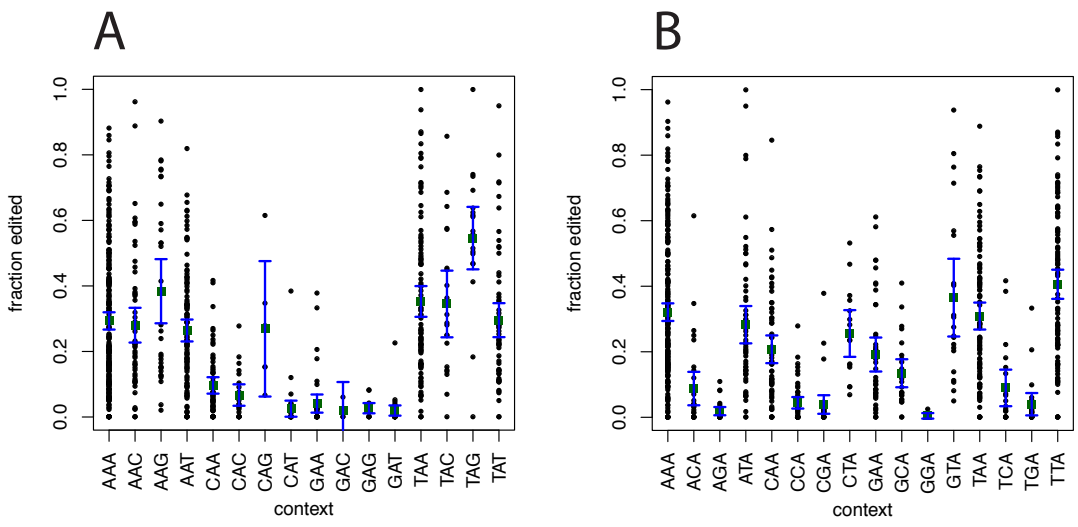

Figure 2.8: Fraction of editing at predicted edit sites as a function of three-letter context. A) The three-letter context centered at the edited base. The 5' base is much more predictive of editing level than the 3' base. B) The three-letter context ending at the edited base. The base two back from an edited position has only a small effect.

of candidate alignments (figure 2.9A). This makes run time difficult to predict *a priori* as the number of candidate alignments depends not only on the number of reads, but also on the number of candidate alignments per read. For example, there are five times as many reads that align to DNAREP1_DM repeats as there are reads that align to FW_DM repeats. However predicting hyper-editing in DNAREP1_DM takes less than half as long per step, because FW_DM repeats are much more similar to one another than DNAREP1_DM repeats are to each other. Alignment probabilities are calculated independently, making the E step highly parallelizable. For FB4_DM, a single EM step takes 1353, 664, 365, 215, and 131 seconds on 1, 2, 4, 8, and 16 cores, respectively.

While EM usually converges in a small number of steps, as it runs, RepProfile suggests new initial conditions to explore alternate hyper-editing solutions. At minimum one initial condition is tried for each hyper-edited repeat (figure 2.9B). However RepProfile will continue trying new initial conditions if a more likely solution is found with a different set of hyper-edited repeats. Most TEs (about 80%) run in 30 minutes or less on 8 cores, but six TEs required run times of a day or more. The most computationally intensive repeat,

PROTOP(A/B), required about 4 days of run time with 815 steps. Running RepProfile on all TEs required a total of 16 node-days at 8 cores per node.

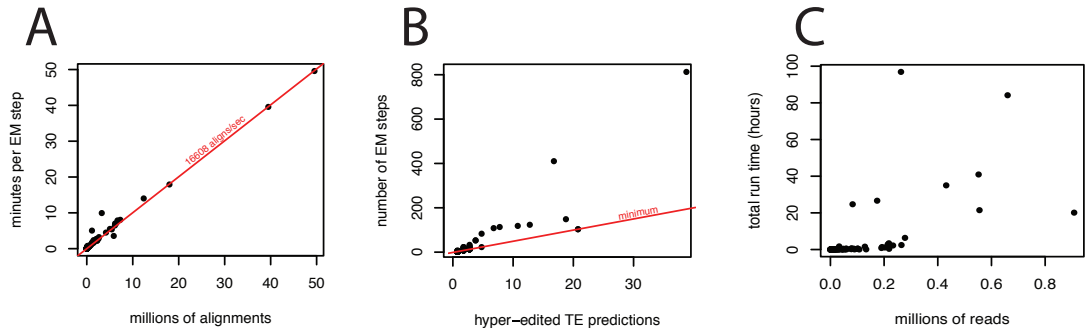

Figure 2.9: EM step run time. EM run time on 8 cores. A) The RepProfile bottleneck occurs during the E step where about 16,608 candidate alignments are processed per second. B) The red line indicate the minimum number of steps run by RepProfile. More steps are run if many local maxima are found. C) RepProfile runs time vary widely.

2.1.3.11 Summary

RepProfile provides accurate RNA hyper-editing predictions that are validated both by simulated data and by individual sequence clones. This analysis reveals hyper-editing that is not found by other methods. In particular, RepProfile predicts the hyper-editing of long, perfect dsRNA formed by FB4_DM elements – a repeat whose relevance to hyper-editing was not previously known – in the introns of genes with synaptic function. It also estimates the level of editing – something that other methods do not do. In addition to finding many hyper-editing events in long, perfect dsRNA, these results show that ADAR often edits a run of adjacent adenosines; that editing is rare in helices less than 20 base pairs long but becomes more frequent as helix length increases; and that, consistent with previous findings, adenosines are more likely to be edited if they follow A or T than if they follow C or G.

2.2 Genomic variation within repeats

2.2.1 Background

Transposable elements are a significant source of genomic variation between individuals, both because novel germline insertions create new heritable variations and because repetitive sequence is less well conserved than protein coding regions. On average, two humans differ by about 285 insertions of LINE1 TEs – the only TE known to be active in healthy humans. In *Drosophila melanogaster*, the drosophila genome reference panel found an average of 1,175 TE insertions in each lineage, with most TE insertions being lineage specific[64]. Even when they do not directly interfere with coding sequence, TEs insertions still effect the expression of nearby genes. In *Drosophila melanogaster* Cridland et al.[14] found that TE insertions with 500 basepairs of a gene decrease expression by an average of 0.67 standard deviations.

A range of tools exist to find novel TE insertion sites[80,43,88,26]. These methods find reads for which one read end matches the TE sequence and the other end aligns somewhere in unique sequence. Reads for which one read end aligns entirely to TE sequence and the other read end aligns entirely to unique sequence indicate that insertion occurred somewhere near the unique alignment. The exact site can be found by looking at read ends that span the breakpoint – reads ends that align partially to unique sequence and partially to TE sequence. Fewer methods exist that look for variations within a repeat or try to reconstruct the sequence of novel repeat insertion by learning how the novel sequence differs from an archetypal TE copy. The profile and EM method described in this thesis provides a way to find these variations.

2.2.2 Tailoring the model

For the prior on the genome profile, G, we use a simplified version of the prior described in subsection 2.1.2. H is constant, with no repeat being hyper-edited on either strand.

Each position, T_i, can either be one of the three types of SNP or match the reference. The model parameters are unchanged from subsection 2.1.2. Because many TE copies have significant deletions, we also need to learn insertions and deletions. To include indels, we estimate $\mathbb{E}_{A|t}[U]$ using the profile HMM based method described in section 1.4 and learn indel parameters using the additional EM calculations described in that section. Because we are looking for genomic variation, it makes sense to use genomic DNA sequence data. Thus we do not have to consider expression. In this subsection, X is a constant, with X_q proportional to the length of repeat q, yielding a distribution in which each read starting position is equally likely. The initial condition includes no SNPs and has the following indel parameters:

$$\delta_i^{open} = \iota_i^{open} = 0.001$$

$$\delta_i^{ext} = 0.999$$

$$\iota_i^{ext} = 0.9$$

for all $1 \leq i \leq n$. The indel extension parameters are chosen so that if a genome profile match exists for both ends of a split read end, it is much more likely to be align as a deletion than as an insertion. Initial conditions with other indel extension parameters were considered, but they converged to inferior local maxima.

2.2.3 Results

2.2.3.1 Accurate SNP identification is possible when reads overlap an average of two or more SNPs or informative positions.

When predicting RNA hyper-editing in section 2.1, we found editing at about one in four adenosines, meaning that on average a single read included more than 10 edited bases. Using so many edited positions as new informative positions, allowed us to be confident

in the MLE estimate. However genomic variations occur far less densely, so we need to know how frequently variations must occur in order for the profile method to accurately predict SNPs.

To measure the ability of the profile method described in this thesis to predict SNPs, 150,000 100 basepair paired-end reads were simulated from a hypothetical repeat that includes 100 identical copies of a 4kb sequence, each with 1kb of unique flanking sequence. Simulations were done with unknown SNPs every 25, 35, 50, 70, 100, 125, 150, 175, 200, 300, 400 and, 500 positions. Sensitivity and PPV for each of these tests is shown in figure 2.10.

Sensitivity and PPV begin to drop when SNPs occur less frequently than one in every 100 positions. Because each read includes 200 sequenced positions, accurate estimate requires that each read must overlap at least two SNPs on average. When a read overlaps two SNPs, it indicates that both SNPs occur in the same repeat. If one read end overlaps unique sequence and the other end overlaps a SNP, we know which repeat copy that SNP occurs in. Thus if we have many reads that overlap two or more SNPs, we can string the SNPs together and join them to the unique flanking sequence, allowing us to correctly identify SNP locations. On the other hand if no repeat that overlaps a particular SNP also overlaps another SNP, we have no way of knowing which repeat copy that SNP belongs to.

2.2.3.2 The genomic DNA model can accurately reconstruct the sequence of a novel insertion.

The ability of this method to accurately predict the sequence of a novel insertion was tested by simulating read data from the reference genome, while treating one reference TE copy as if it were a novel insertion. The tests were performed on the *Drosophila melanogaster* POGO repeats. POGO was chosen because it is a small repeat family with a high degree of similarity between repeats. Thus reconstructing POGO sequences

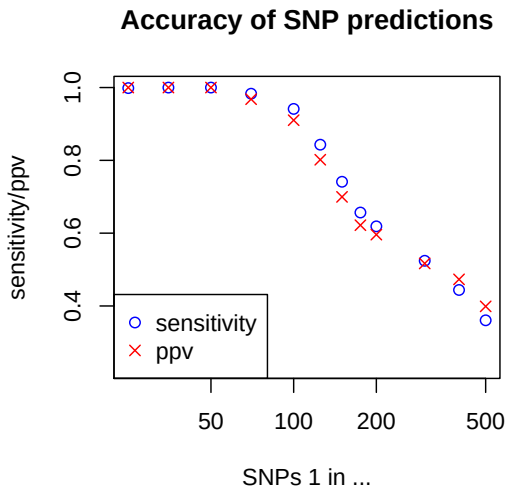

Figure 2.10: Sensitivity at PPV as a function of SNP density. The profile method needs SNPs to occur about 1 in every 100 positions for accurate estimation. As SNPs become less frequent sensitivity and PPV drop.

remains theoretically challenging, but is computationally somewhat less intensive.

First, reads were simulated (without added variation) from the reference POGO elements, together with 1000 bases of flanking sequencing. Next, one POGO element was removed and replaced with a copy of the archetypal POGO element. This simulates the information provided by one of the standard insertion prediction methods. We know the location of the TE insert, but we do not know anything about its sequence. Thus the archetypal repeat sequence represents our best initial guess of the sequence. Finally, we can use the model described in section 2.2.2 to learn a profile with indels for the novel repeat copy. In this simulation, the sequences of other repeat copies are assumed to be correct. To test the accuracy of the prediction, the most likely sequence given the learned profile is aligned to the correct reference sequence using BLAT[44]. Some reference POGO elements sequences are fragments whose flanking sequence is not known. These elements were not tested. Furthermore POGO elements that are small, highly divergent fragments less than 500 basepairs long were not tested as they do not provide a good model of novel insertion. Finally, one locus that has three elements separated by less than 1000 base pairs was not considered, as the flanking sequences for these elements would overlap. The other

24 POGO elements were all tested. The reconstruction accuracy can be found in table 2.6.

For nine of the sequences, the reconstructed sequence is exactly correct. For all but one sequence, the reconstructed sequence is at least 95% accurate. The method fails to accurately reconstruct one POGO element (located at chr3L:26653-29800). This POGO element includes a deletion that is identical to deletions in two other POGO copies and is too far away from the flanking sequence for a read to span both the deletion and the flanking sequence. This makes that element particularly difficult to accurately reconstruct.

Position	Matches	Estimated Length	Actual Length	%Matches
chr2L:2933353-2935475	2060	2060	2060	100.00%
chr2L:7959094-7960450	1353	1368	1356	98.90%
chr2L:17912011-17914134	2123	2123	2123	100.00%
chr2L:21539459-21540930	1471	1471	1471	100.00%
chr2R:16117-17444	1324	1328	1327	99.70%
chr2R:4761999-4763490	1489	1490	1491	99.87%
chr2R:11472701-11473843	1140	1141	1142	99.82%
chr2R:17782415-17783563	1142	1149	1148	99.39%
chr3L:27653-28800	1146	2123	1147	53.38%
chr3L:11864606-11865846	1240	1240	1240	100.00%
chr3L:20257384-20258451	1033	1064	1067	96.81%
chr3L:24616744-24618090	1346	1346	1346	100.00%
chr3R:12022937-12025059	2122	2122	2122	100.00%
chr3R:13135880-13138000	2120	2120	2120	100.00%
chr3R:22234736-22236000	1255	1298	1264	96.69%
chr3R:29760414-29761560	1097	1136	1146	95.72%
chrX:15421073-15423429	1456	1456	1456	100.00%
chrX:23036435-23037781	1346	1346	1346	100.00%
chrUn_DS484865v1:1836-2789	951	952	953	99.79%

Table 2.6: Accuracy of POGO elements sequence reconstruction. Estimated length is the length of the reconstructed sequence, and actual length is the length of the reference element. % matches is the ratio between the number of matches and the larger of estimated length and actual length.

2.2.3.3 Analysis of genomic reads shows evidence for the deletion found in FB4_DM clone sequences

In addition to showing wide spread editing, the validation sequences discussed in subsection 2.1.3.2 showed a 492 basepair deletion. Thus these Sanger validation sequences can also be used to test the method described in this section. The EM/profile method described in this section was applied to *Drosophila melanogaster* genomic DNA generated by Yiannis Savva in the Reenan lab. Unfortunately finding indels in repeats is a much more computationally intensive task that finding single position variations, making it less suitable for larger repeat families. For the FB4_DM repeats, about one week of compute time was required to run 10 EM iterations – far short of the number of iterations that would be required to achieve convergence. Nevertheless through these 10 steps, EM puts increasing weight on the presence of a deletion in this FB4_DM element. The probability of a deletion starting at chrX:8,928,793 increases at each step: 0.001, 0.050, 0.112, 0.199, 0.314, 0.458, 0.574, 0.659, 0.727. No other position has a probability over 50% of starting a deletion.

2.2.3.4 Within TE indel variation is widespread in POGO elements.

Next the same *Drosophila melanogaster* genomic DNA was used to predict SNPs and indels in POGO elements. Because there are fewer POGO elements than there are FB4_DM elements, 100 EM steps only requires about 35 hours of run time on 16 cores, achieving a solution that is near convergence. Across all the POGO elements, SNPs that occur in at least half the flies occur at about 1% positions, and fixed SNPs occur at approximately 0.66% of positions. These numbers comparable to SNP rates observed genome wide in the Drosophila Genome Reference Panel (DGRP)[37]. Because DGRP reports variation between a host of fly strains, and we only consider variations between one strain and the reference, more careful comparison is not possible.

However insertions and deletions showed large variation from the reference, dwarfing the

variation due to single nucleotide difference. For the POGO elements tested in table 2.6, the ratio between the expected number of indel positions and the total number of POGO positions in the reference genome is 0.203. While this ratio is large, it is consistent with the fact that 38.1% of bases were deleted in the section of FB4_DM that was sequenced for editing validation. Figure 2.11 shows the indel ratio for each individual element.

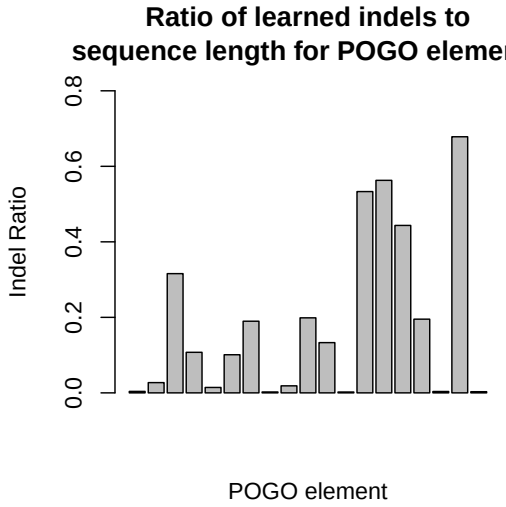

Figure 2.11: Indel ratio for individual POGO elements. Bar heights are ratio between the expected number of indels in an element and the length of that element. Only the elements listed in table 2.6 are plotted. The order in the table and the figure are identical.

2.3 Repeat expression

2.3.1 Background

Which transposable elements are being expressed and at what level is a question of particular interest. Under certain conditions, such as aging[77], neurodegenerative disease[86] and cancer[7], TE expression has been shown to increase. However it is not known if TE expression increases uniformly, or if there are a few copies that are individually activated under these conditions. The profile/EM method described in this thesis can also be applied to this problem. While learning expression does not give us new informative positions, it does still provide information that can be used to refine read alignments. If we

learn that TE x is expressed ten times more than TE y, then a read that aligns equally well to both to both x and y would be ten times more likely to have come from x than y.

EM has been used extensively to estimate expression levels. Most commonly to estimate the relative expression of isoforms of the same gene[79,5,78]. Most eukaryotic genes consist of several exons separated by introns. Depending on how the exons are spliced together, different mRNAs can be made from the same genomic sequence. Because these alternate splice isoforms share many of the same exons, it is often impossible to tell which isoform a particular read comes from. However EM based methods allow us the find the relative expression levels that make the observed read data most likely. The TEtranscripts package[40] also applies this method to repeats, but it does not use any of the profile methods outlined in this thesis. As a result TEtranscripts is only capable of estimating the expression level of all the copies of a particular TE together rather than the expression level of individual TEs.

In this subsection I use the profile/EM method outlined in this thesis to estimate the expression of individual TE copies. However because we do not have a large number of RNA editing sites as in section 2.1 or large deletions as in section 2.2 we also have to make sure that the expression parameters we are estimating are actually identifiable by maximum likelihood estimation.

2.3.2 Tailoring the model

In this section, we assume that the genome profile, G, is known or has already been estimated. G could be derived from the reference genome, assuming that it is approximately correct, or it could be estimated from genomic data as in section 2.2. Thus G is a constant value (with no prior). The initial guess for X is the uniform distribution. $\mathbb{E}_{A|t}$ is estimated by summing over all candidate alignments with 2 or fewer mismatches per read end.

2.3.3 The question of identifiability

The consistency of maximum likelihood estimation depends on an assumption of identifiability. In the case of estimating repeat expression, this means that if two expression profiles are not equal ($X \neq X'$) then the corresponding likelihoods must also be different: $f(R|X) \neq f(R|X')$. Unfortunately this assumption is likely to be violated. Some TE copies will be expressed as part of an intron, including flanking sequence that will make it possible to differentiate them from other, identical copies. However some TE copies, especially retrotransposons, are transcribed from their own promotors, yielding transcripts that include only the TE copy. It will not be possible to distinguish the transcription of such copies if they are identical.

If the expression level of two repeats, X_i and X_j, are indistinguishable, the log likelihood function, $\log f(x) = P(R|X = x)$, will be constant along lines of the form

$$X_i + X_j = c.$$

We can look for such pairs by considering a second order Taylor expansion about the maximum likelihood estimate. This is precisely the idea behind Fisher information and observed information[21]. Observed information is the Hessian of the log likelihood function evaluated at the maximum likelihood estimate. Fisher information is the expected value of the observed information. For this problem it makes sense to consider observed information as we do not know whether anchored reads will be present in an RNAseq experiment. If a repeat is transcribed as part of an intron, then we would expect to sequence the entire intron, yielding anchored reads. However if a repeat is transcribed from its own promotor we only expect to sequence the read itself, and we would not have anchored reads. Calculating Fisher information with or without anchored reads would potentially yield two very different results, with no clear way to decide between the two. Therefore if we want to know whether X_i and X_j are sufficiently distinguishable, we

would like to know if

$$D^2_{u_i-u_j} \log f(\hat{x}) = \left.\frac{\partial^2 \log f}{\partial x_i \partial x_i}\right|_{x=\hat{x}} - 2\left.\frac{\partial^2 \log f}{\partial x_i \partial x_j}\right|_{x=\hat{x}} + \left.\frac{\partial^2 \log f}{\partial x_j \partial x_j}\right|_{x=\hat{x}} > 0 \tag{2.1}$$

where u_i and u_j are unit vectors and $\hat{x}$ is the MLE estimate of X.

To calculate 2.1, we need to find the Hessian matrix of the log likelihood. Recall that the likelihood function is as follows:

$$\begin{aligned} P(R|G,X) &= \sum_A P(R,A|G,X) \\ &= \sum_A \prod_{j=1}^{m} X_{A_j} \prod G_{A_j}(R_j) \\ &= \prod_{j=1}^{m} \sum_{A_j} X_{A_j} G_{A_j}(R_j) \end{aligned}$$

Thus the loglikelihood is:

$$\log P(R|G,X) = \sum_{j=1}^{m} \log \sum_{A_j} X_{A_j} G_{A_j}(R_j)$$

First we can take the derivative with respect to X_{ℓ_1}:

$$\begin{aligned} \frac{\partial}{\partial X_{\ell_1}} \log P(R|G,X) &= \frac{\partial}{\partial X_{\ell_1}} \sum_{j=1}^{m} \log \sum_{A_j} X_{A_j} G_{A_j}(R_j) \\ &= \sum_{j=1}^{m} \frac{\partial}{\partial X_{\ell_1}} \log \left[\left(\sum_{A_j \in \ell_1} G_{A_j}(R_j) \right) X_{\ell_1} + \sum_{A_j \notin \ell_1} X_{A_j} G_{A_j}(R_j) \right] \\ &= \sum_j \frac{\sum_{A_j : A_j \in \ell_1} G_{A_j}(R_j)}{\sum_{A_j} X_{A_j} G_{A_j}(R_j)} \end{aligned}$$

where $A_j \in \ell_1$ indicates that A_j is alignment to repeat ℓ_1. Now we take the derivative

with respect to X_{ℓ_2}:

$$\begin{aligned}\frac{\partial^2}{\partial X_{\ell_1}\partial X_{\ell_2}}\log P(R|G,X) &= \frac{\partial}{\partial X_{\ell_2}}\sum_j \frac{\sum_{A_j:A_j\in\ell_1} G_{A_j}(R_j)}{\sum_{A_j} X_{A_j}G_{A_j}(R_j)} \\ &= \sum_j \frac{\partial}{\partial X_{\ell_2}} \frac{\sum_{A_j:A_j\in\ell_1} G_{A_j}(R_j)}{\left(\sum_{A_j\in\ell_1} G_{A_j}(R_j)\right) X_{\ell_1} + \sum_{A_j\notin\ell_1} X_{A_j}G_{A_j}(R_j)} \\ &= \sum_j \frac{\left(\sum_{A_j:A_j\in\ell_1} G_{A_j}(R_j)\right)\left(\sum_{A_j:A_j\in\ell_2} G_{A_j}(R_j)\right)}{\left(\sum_{A_j} X_{A_j}G_{A_j}(R_j)\right)^2}\end{aligned}$$

Thus, returning to equation 2.1:

$$D^2_{u_{\ell_1}-u_{\ell_2}}\log f(\hat{x}) = \frac{\partial^2 \log f}{\partial x_{\ell_1}\partial x_{\ell_1}}\bigg|_{x=\hat{x}} - 2\frac{\partial^2 \log f}{\partial x_{\ell_1}\partial x_{\ell_2}}\bigg|_{x=\hat{x}} + \frac{\partial^2 \log f}{\partial x_{\ell_2}\partial x_{\ell_2}}\bigg|_{x=\hat{x}} \tag{2.2}$$

$$= \sum_j \frac{\left[\left(\sum_{A_j:A_j\in\ell_1} G_{A_j}(R_j)\right) - \left(\sum_{A_j:A_j\in\ell_2} G_{A_j}(R_j)\right)\right]^2}{\left(\sum_{A_j} X_{A_j}G_{A_j}(R_j)\right)^2} \tag{2.3}$$

We can conclude the following from equation 2.3: X_{ℓ_1} and X_{ℓ_2} are indistinguishable if and only if every read aligns equally well to repeat ℓ_1 and ℓ_2.

Proof: If some read j aligns better to ℓ_1 or ℓ_2 then:

$$\sum_{A_j:A_j\in\ell_1} G_{A_j}(R_j) \neq \sum_{A_j:A_j\in\ell_2} G_{A_j}(R_j)$$

and so $D^2_{u_{\ell_1}-u_{\ell_2}}\log f(\hat{x}) > 0$. If each read aligns equally well to ℓ_1 and ℓ_2 then the repeats must be indistinguishable.

2.3.4 Results

2.3.4.1 Simulation shows that expression levels of Gypsy repeats are measured accurately.

To test the ability of the EM/profile method described in this thesis, 17,000 reads were generated from the Gypsy repeat sequence in *Drosophila melanogaster* without flanking

sequence. 17,000 reads were chosen to reflect the coverage in the real RNAseq data used in section 2.1.3. The relative expression of each element in the simulation was 1.1^i where i is a randomly assigned index. In other words, the Gypsy repeats were randomly permuted and then each sequence was assigned an expression level that is 10% higher than the previous sequence. Note that Gypsy elements vary in length, so the 10% increase is not carried over to the read counts plotted in figure 2.12.

MAP estimates after 10 EM steps are shown in figure 2.12. A test was also run with 50 EM steps, but it provided no improvement in accuracy. For the MAP estimate after 10 EM steps, about 1000 reads (6%) would need to be reassigned to get the correct simulated read counts. About two thirds of these misaligned reads are assigned to one of three unidentifiable Gypsy elements. Since it is not possible to tell which of the three a read should align to, these three elements were merged into a single read count (figure 2.12 right side). After merging only about 325 (2%) of the reads need to be realigned to get the correct simulated read counts.

2.3.4.2 Increase in Gypsy expression in published ALS study is due to an increase at three intact Gypsy loci.

In Krug et al.[48] the authors express human TDP-43 protein, which is associated with ALS, in *Drosophila melanogaster* glial cells. They observed an increase in Gypsy expression, but could not pinpoint the loci responsible. Figure 2.13 shows EM/profile expression estimates for the data analyzed in Krug et al.[48]. The largest increases in Gypsy expression occur at three intact Gypsy loci: chr3L:20,477,216-20,483,944, chr3L:27,225,300-27,232,027 and chr3R:31,074,691-31,081,418. All three instances include open reading frames encoding gag, pol and env proteins. A BLAT search indicates that there are five other intact elements, but these do not show an increase in expression with glial hTDP-43. The expression level of all five intact elements are distinguishable.

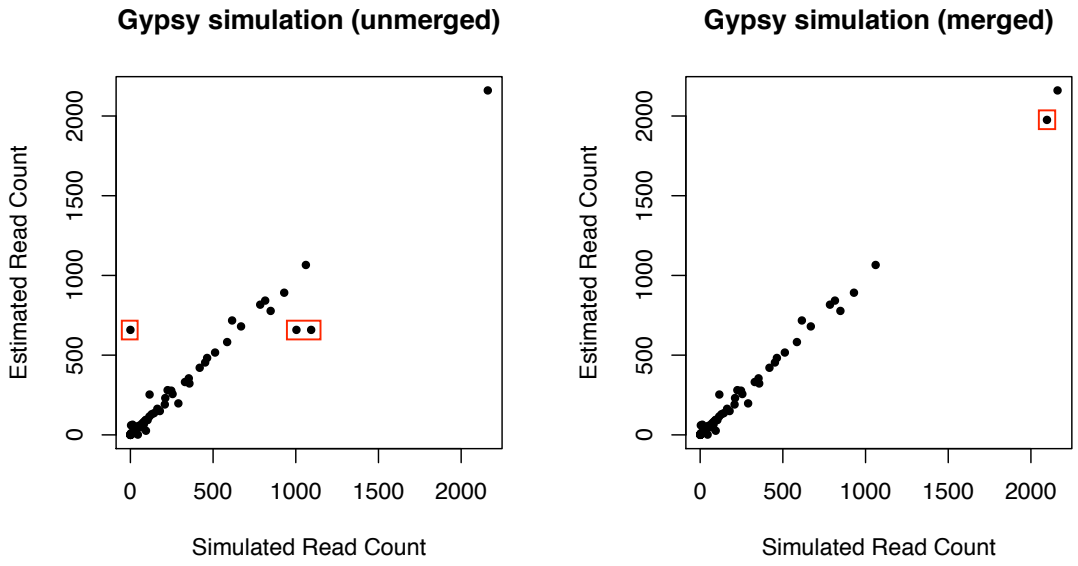

Figure 2.12: Simulated and estimated expression levels. Left, without merging unidentifiable elements. Right, after merging unidentifiable elements. Highlighted points on the left are not identifiable and merged into a single point on the right.

2.4 Summary

Next-generation sequencing platforms, such as Illumina, make it possible to sequence billions of nucleotides for a few thousand dollars. This has caused a "big data" explosion in molecular biology. However the repetitive nature of transposable elements makes it difficult to apply this data to questions about the role of TEs in disease and in normal cellular function. Probabilistic modeling and MAP estimation by the EM algorithm makes it possible to bridge this gap. This method provides accurate estimates of RNA hyper-editing, genomic variation and expression in repetitive sequence.

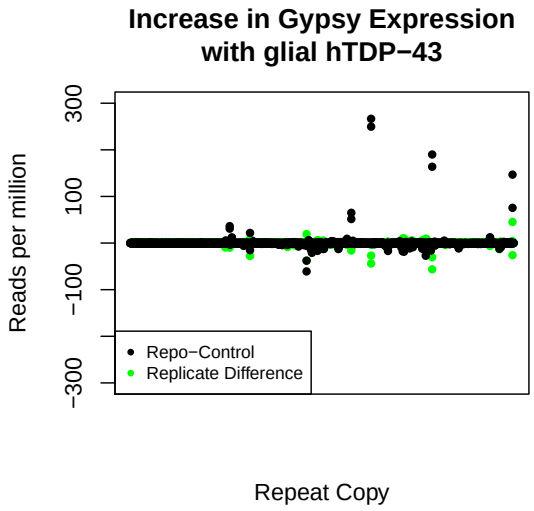

Figure 2.13: Increase in Gypsy Expression in glial hTDP-43 data from[48]. Expression of Gypsy elements was estimated for two replicates with human TDP-43 expressed and *Drosophila melanogaster* glial cells and for two control replicates. The difference in expression at each Gypsy locus is plotted. The difference between TDP-43 and control is plotted in black. For comparisons the difference between replicates of the same type is plotted in green.

3

RNA structure

3.1 Introduction

3.1.1 RNA secondary structure

As noted in section 2.1.1 while RNA is transcribed as a single stranded molecule it is still capable of forming basepairs between pairs of nucleotides within a single RNA molecule. This internal basepairing fold the RNA molecule into structures that range from the simple double stranded structure described in section 2.1 to complex structures, such as the ribosomal RNAs, that have specific, carefully controlled cellular functions. Even randomly generated RNA sequences will form internal basepairs. The set of basepairs formed by an RNA molecule is called the secondary structure (SS). This is in relation to the primary structure (the sequence) and the tertiary structure (how the molecule is folded in 3D space.)

The ability for RNA molecules to fold into specific shapes allows some RNA molecules to have enzymatic or other functional activities. Perhaps most notably, the ribosome – the structure responsible for translating mRNA into protein – is made of several RNA subunits[102,30,28]. These subunits must fold correctly for the ribosome to function properly. Furthermore, tRNA – the molecules that carry amino acids to the ribosome – are also functional RNAs[99]. Some introns – called group I and II introns – form structures that allow the intron to cut itself out of an RNA molecule during the formation of mRNA[15,71]. Ribonuclease P (RNAp) is a functional RNA that cleaves other RNA molecules[6]. This is far from an exhaustive list of function RNAs. Others include signal recognition particles (SRP)[53] and telomerase – a molecule that is responsible for repairing chromosome tips that are shortened during replication[81]. However even protein coding mRNAs show evidence of structural selection, with certain mutations that affect RNA structure but not the encoded protein having observable effects on the rate of protein translation[51].

The relationship between RNA structure and function makes understanding and predicting RNA structure a key problem in computational biology. The nearest neighbor model, described below in section 3.1.2, makes it possible to find the single most stable RNA structure or to calculate the partition function for the Boltzmann ensemble, allowing for the exploration of alternate structures. Exploring alternate structures is particularly important for molecules that take on multiple conformations[75] or for molecules where the predicted most stable structure does not accurately represent the native structure found in vivo[66]. Recently there has been particular interested in understand how nucleotide changes affect RNA structure. In many cases naturally occurring mutations will not affect protein coding sequencing but will still affect phenotype by modifying RNA structures. These mutations have been dubbed "riboSNitches"[31,104,97]. New CRISPR technologies make it possible to specifically target individual bases in an RNA molecule for editing[13]. This creates a need to understand how these edits affect RNA structure and to find edits that will change the conformation of an RNA molecule in a desirable way.

3.1.2 Nearest neighbor model

The most popular model for RNA secondary structure is the nearest neighbor model. There are several software packages that implement the model[35,113,19,87]. In this chapter we focus on the model implemented in the RNAstucture package[87]. The nearest neighbor model provides a function for the free energy of a given secondary structure that can be calculated, maximized and summed efficiently.

For a sequence $S_{1:n}$, we notate the secondary structure as $X = \{X_{ij}\}_{1 \leq i,j \leq n}$, where $X_{ij} = 1$ if i and j are paired and 0 otherwise. The nearest neighbor model has three rules regarding which basepairs can be present.

1. (Watson-Crick and Wobble). Only A-U, G-C and G-U pairs are allowed: If $X_{ij} = 1$ then $\{S_i, S_j\} \in \{A, U\}, \{G, C\}, \{G, U\}\}$.
2. (No triples). Each position is only allowed be pars of at most one basepair: $\sum_j X_{ij} \leq 1$ for all i.
3. (No pseudoknots). Basepairs must be nested or non-overlapping: $X_{ij} + X_{kl} \leq 1$ for all $i < k < j < l$.

The largest contribution to free energy in the nearest neighbor model comes from the favorable stacking of adjacent basepairs. Two pairs $\{i, j\}$ and $\{k, \ell\}$ are stacked if $k = i+1$ and $\ell = j - 1$. A single basepair on its own is not favorable, but adding a new basepair stacked on one that is already present is favorable. A set of stacked basepairs is called a helix or a stem. In general, a helix with one or two base pairs is not favorable, but a helix with three or more basepairs is. However the favorability of the helix also depends on the types of basepairs present. A G-C pair is held by three hydrogen bonds and is the most stable. An A-U pair, held by two hydrogen bonds, is also stable. A G-U pair, sometimes called a wobble pair is not very stable and would only appear as part of a longer helix. Under the nearest neighbor model, the contribution of a pair $\{i, j\}$ to the free energy of a structure depends on the letters in the pair and whether the nearest neighbor $\{i-1, j+1\}$ is paired and, if it is paired, the letters in that pair.

Because the free energy contribution of a pair only depends on its nearest neighbor and because the no pseudoknots requirement prevents the formation of a pair that connects the subsequence between the paired nucleotides to the subsequence outside, efficient dynamic programming algorithms are possible for this model. Suppose that we want to find the minimum free energy (MFE) for a given sequence $S_{1:n}$. We will denote the MFE as follows: $M(1:n|\emptyset)$. The $1:n$ indicates that we are considering the MFE of the entire sequence. The $|\emptyset$ indicates that we are not including a nearest neighbor. We can then break apart $M(1:n|\emptyset)$ by considering each of the possible positions – $\{2,\ldots,n-1\}$ – that position 1 might pair with, along with the possibility that 1 might not be paired. If $\{1,j\}$ is a pair that we can minimize, over $M(j+1:n|\emptyset)$ and $M(2:j-1|\{1,j\})$ separately, where $M(2:j-1|\{1,j\})$ indicates the minimum free energy of the sequence $S_{2:j-1}$ with the nearest neighbor pair $\{i,j\}$. Thus we have the following recursion to find the minimum free energy.

$$M(i:j|\mathcal{X}) = \min\Bigg(\Big(M(i+1:j|\{i,\emptyset\}) + E(\{i,\emptyset\}|\mathcal{X})\Big), \tag{3.1}$$

$$\min_{k\in\{i+1,\ldots,j-1\}}\Big(M(i+1:k-1|\{i,k\}) + E(\{i,k\}|\mathcal{X}) + M(k+1:j|\emptyset)\Big)\Bigg) \tag{3.2}$$

The software that we use to calculate free energy in this chapter, RNAstructure[66], includes a few additional terms that provide added accuracy when predicting the secondary structure adopted by a given RNA sequence. The package also considers the effect that different types of bulges and loops have on the free energy of an RNA secondary structure. Loops are a string of unpaired nucleotides $i:j$, for which $\{i-1,j+1\}$ are paired. A bulges is also a stretch, $i:j$, of unpaired nucleotides, but in the case of a bulge, $i-1$ and $j+1$ are paired but not to each other. RNAstructure also allows for some coaxial stacking: In some cases the structure of an RNA molecule is able to twist in 3D space so that two pairs that are not nearest neighbors can still stack into a helix.

3.1.3 The Boltzmann ensemble

As noted above, for many RNA molecules and many applications, finding the single most stable RNA structure is not sufficient. Thus in many cases it makes sense to look at the equilibrium distribution over RNA secondary structure to get a more complete picture. This equilibrium distribution is call the Boltzmann ensemble and has the following distribution function:

$$P(X|S) = \frac{\exp -E(X|S)}{\sum_{X'} \exp -E(X'|S)}$$

for structure X and a sequence S. $E(X|S)$ is the nearest neighbor energy function described in section 3.1.2. The dynamic programming algorithm described above (equation 3.1) can be modified to calculate $\sum_{X'} \exp -E(X'|S)$ efficiently and to sample structures from the Boltzmann ensemble.

Many RNA structures have Boltzmann ensembles with many centers of mass, potentially predicting structures with little overlap. This has let to efforts to cluster structures and provide a simplified representation of the Boltzmann distribution. Methods include standard clustering algorithms[17,10,18] and strategies tailored to RNA SS. RNAshapes[27,100,38] finds structures that share a common "shape", and Rogers et al.[90] group structures that share common helices in a process called "profiling". Both of these strategies simplify the RNA folding problem by abstracting from individual basepairs to the helices that define RNA SS. While grouping structures based on common features does allow for a simplified description, such methods do not provide insight into the underlying features – conflicting basepairs – that drive multimodality in the Boltzmann distribution.

3.1.4 Most informative basepair

In this chapter we make use of the most informative basepair (MIBP) defined by Luan Lin in her thesis and described in our BMC bioinformatics paper[62]. We equate the com-

plexity (or simplicity) of a distribution with how unsure we are about the value of the corresponding random variable. The uncertainty in a basepair $\{i, j\}$ is measured using information entropy:

$$H[X_{ij}] = -p_{ij} \log_2 p_{ij} - (1 - p_{ij}) \log_2 (1 - p_{ij}) \tag{3.3}$$

where $p_{ij} = P(X_{ij} = 1)$, $0 \log 0 = 0$ and the units of H are in bits. When we are less sure about a basepair, p_{ij} is closer to 1/2 and $H[X_{ij}]$ is larger. Conversely when we are more sure about a basepair, p_{ij} is closer to 0 or 1 and $H[X_{ij}]$ is smaller.

Now if we condition on another basepair X_{kl}, we have two conditional distributions: $P(X_{ij}|X_{kl} = 1)$ and $P(X_{ij}|X_{kl} = 0)$, each with corresponding entropies: $H[X_{ij}|X_{kl} = 1]$ and $H[X_{ij}|X_{kl} = 0]$. The conditional entropy is defined to be

$$H[X_{ij}|X_{kl}] = (1 - p_{kl}) H[X_{ij}|X_{kl} = 0] + p_{kl} H[X_{ij}|X_{kl} = 1] \tag{3.4}$$

and is the average uncertainty in X_{ij} after we learn the value of X_{kl}. Therefore the amount of information that X_{kl} tells us about X_{ij} is

$$I(X_{ij}; X_{kl}) = H[X_{ij}] - H[X_{ij}|X_{kl}] \tag{3.5}$$

This value is referred to as the mutual information between X_{ij} and X_{kl} and it is symmetric[12]. By equations 3.4 and 3.5, on average the distribution of X_{ij} conditioned on X_{kl} is $I(X_{ij}; X_{kl})$ bits simpler than the unconditioned distribution. We can then measure the amount of information that a basepair provides about the rest of the secondary structure by adding up its mutual information with each other basepair. We can then condition on the basepair that has the greatest sum of mutual information to get a less complex conditional distribution. We call this basepair the most informative basepair:

Definition: The most informative basepair (MIBP) is the basepair that has the largest

sum of mutual information:

$$\text{MIBP} = \underset{kl}{\text{argmax}} \sum_{ij} I(X_{ij}; X_{kl})$$

Calculating mutual information requires the joint probability of every pair of basepairs, a computationally intensive task. However we can quickly estimate the mutual information from sampled structures. Structures can be sampled from the Boltzmann ensemble for a sequence of length n in $\mathcal{O}(n^3)$ time using RNAstructure or a similar tool. We can find the MIBP from sampled structures as follows:

Algorithm:

1. Get 1000 samples from the Boltzmann distribution
2. Considering only pairs that appear in at least 10 and less than 990 samples, estimate the joint probability for each pair of pairs.
3. Use the estimates to calculate mutual information for each pair of basepairs.
4. Sum the mutual information of each basepair.
5. Find the basepair that has the greatest sum.

Basepairs that appear in fewer than 10 or more than 990 samples will have low entropy and so will not make a significant contribution to mutual information. Thus we can improve computational efficiency without sacrificing accuracy by ignoring them. In general we find that 1000 samples is enough to get an accurate estimate of the base pairing probabilities.

3.2 MIBP clustering algorithm

3.2.1 Tree Based Clustering

In our forthcoming paper, we greedily employ the MIBP algorithm to build a binary tree that clusters structures based on the presence of MIBPs. In this work, I honed MIBP

algorithms created by Luan Lin and Bryce Richards, and applied them to biologically interesting questions. The clustering algorithm first splits the space into a cluster that includes the MIBP and one that does not. We then find the conditional MIBP in each of the clusters and split those clusters in two. We continue this process until the product of the cluster probability and estimated mutual information falls below 2 bits.

Algorithm:

1. Label ensemble cluster ''.
2. For all clusters x:
   (a) Calculate MIBP
   (b) If $P(x) * \text{MI} > 2$ bits: Split x into cluster $x0$ without MIBP and cluster $x1$ with MIBP.
3. Repeat step 2 until no new clusters are made.

The algorithm creates a binary tree of nested clusters, where each branch corresponds to conditioning on the presence or absence of a particular MIBP. The leaves of this tree are then an exhaustive set of clusters. An html file is created that draws the tree and plots the marginal basepairing probabilities at each node. See the section 3.3 for examples. Algorithms that use mutual information to create binary trees, such as ID3 and C4.5[84], are used widely in classification problems. This algorithm employs a similar concept, but it uses the mutual information between basepairs as there is no natural labelling for RNA secondary structures as would be the case in a classification problem.

3.2.2 Conflicting Basepairs and Other Cluster Calculations

Once we have found MIBPs and divided the space, we can examine the individual clusters. First we look for basepairs that conflict with the MIBP. For each MIBP split, we first find the basepair that conflicts with the MIBP and is most probable. Because the MIBP and the conflicting pair cannot be present in the same structure, the most probable

conflicting pair is also the conflicting basepair that has the most pairwise mutual information with the MIBP (see claim below). We then continue in a greedy fashion, repeatedly finding the most probable basepair that conflicts with the MIBP and all previous conflicting pairs. We stop once we have found a total of five conflicting pairs. When conflicting pairs are present, they provide an explanation for the presence of divergent clusters.

We can also use RNAstructure to calculate the marginal probability of each basepair in each cluster and use that information to calculate the conditional entropy in each cluster. Finally, we can calculate the number of structures in each cluster by setting the free energy, $E(x)$, equal to 0 for all structures x and then calculating the partition function. Note that RNAstructure counts structures with the same basepairs but different coaxial stacking separately.

Claim: For fixed p_{ij}, if X_{ij} and X_{kl} conflict, then $I(X_{ij}; X_{kl})$ is a monotonic increasing function of p_{kl}.

Proof: Using the formula for $I(X;Y)$ given in[12] and the fact that $P(X_{ij} = 1, X_{kl} = 1) = 0$:

$$\begin{aligned} I(X_{ij}; X_{kl}) &= (1 - p_{ij} - p_{kl}) \log_2 \frac{1 - p_{ij} - p_{kl}}{(1 - p_{ij})(1 - p_{kl})} + p_{ij} \log_2 \frac{p_{ij}}{p_{ij}(1 - p_{kl})} + p_{kl} \log_2 \frac{p_{kl}}{p_{kl}(1 - p_{ij})} \\ &= (1 - p_{ij} - p_{kl}) \log_2(1 - p_{ij} - p_{kl}) - (1 - p_{ij}) \log_2(1 - p_{ij}) - (1 - p_{kl}) \log_2(1 - p_{kl}) \end{aligned}$$

Taking the derivative with respect of p_{kl} yields:

$$\begin{aligned} \frac{dI(X_{ij}; X_{kl})}{dp_{kl}} &= \frac{1}{\log 2} \left[-2 \log(1 - p_{ij} - p_{kl} + 2 \log(1 - p_{kl}) \right] \\ &= \frac{2}{\log 2} \left[\log \frac{1 - p_{kl}}{1 - p_{ij} - p_{kl}} \right] > 0 \end{aligned}$$

Thus the claim is proved.

3.3 Results

To test our algorithm, we used a set of sequences and corresponding native structures provided to us by David Mathews. This data is used by the Mathews group to test the RNAstructure software package. The sequences are compiled from the following publications:[102,29,28,15,71,6,53,81,114,99]. The test set includes sequences from ten families whose secondary structures have been verified by comparative analysis: 5s, 16s and 23s rRNA, group 1 and 2 introns, RNAse P (RNAp), signal recognition particle (SRP), telomerase, tmRNA and tRNA. The 16s and 23s rRNA sequences were divided into four and six folding domains respectively[29,28] to make the computation more tractable. For 5s, 16s, SRP, telomerase, tmRNA and tRNA, we considered only 10 randomly selected sequences from each family as the test set included a large number of sequences from these families. A list of all sequences considered can be found at the visualization site described in the next subsection.

3.3.1 Visualizations

Visualizations of the Boltzmann ensembles of these sequences can be found at

http://ccmbweb.ccv.brown.edu/wmckerro/MIBP/

Visualizations were designed for the firefox browser, but may work with other browsers as well. Longer sequences may be slow to load. The visualizations draw the binary tree described in the section 3.2. Clicking on a node in the tree reveals the conditional probability of each basepair in the corresponding cluster, showing how the presence or absence of MIBPs affects the structure. Figure 3.1 shows the circle diagrams arranged in a tree for an example RNA molecule.

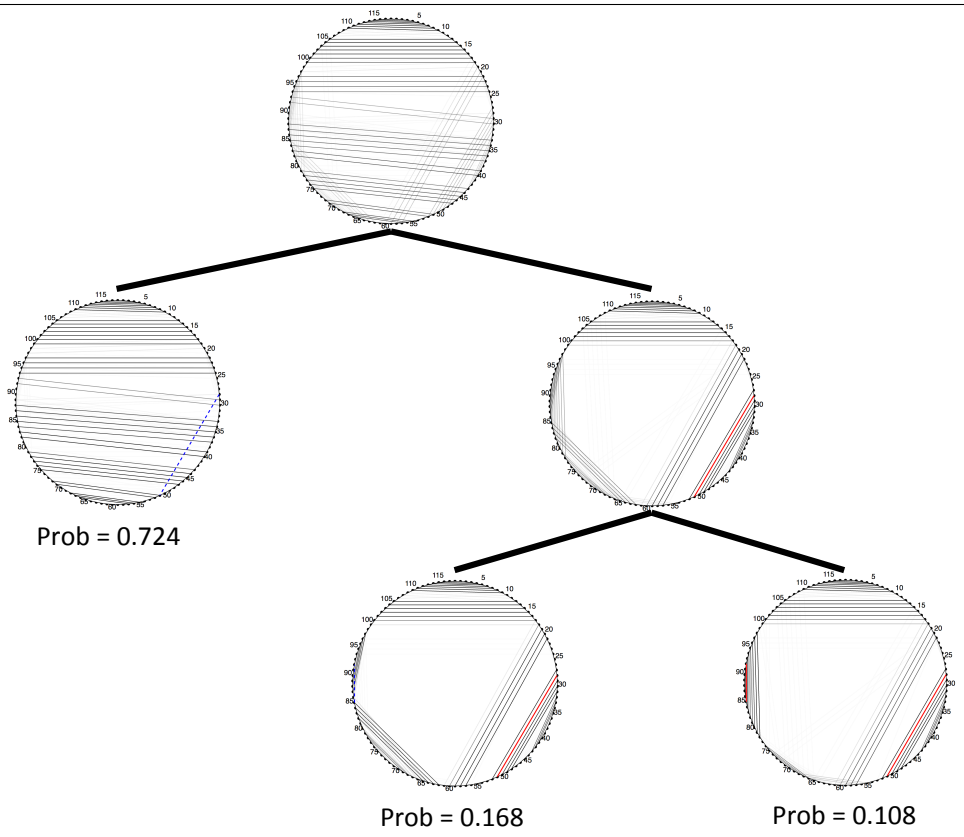

Figure 3.1: Visualization of *Tremella encephala* 5s rRNA (5s3_201 in the test set). Nucleotides are arranged around the edge of a circle and basepairs are drawn as chords connecting the paired bases. MIBPs that are constrained to be present are highlighted in red. Those that are absent are highlighted in blue. The darkness of a plotted basepair is proportional to its probability.

3.3.2 Entropy reduction

To see how adding new clusters affects the conditional entropy, we ran our algorithm for 100 steps on one sequence from each family, yielding 101 clusters for each sequence. As a function of the number of clusters, the entropy closely follows a power law with an exponent that varies from -0.1 for the *Chinchilla brevicaudata* telomerase (AF221937.99-545 in the test set) to -0.4 for the *Clostridium perfringens* tmRNA sequence (Clos.perf._CP000246_1-361). (See figure **??**.)

We also test how the default 2 bit cutoff (see methods) affects the conditional entropy. Across all the sequences tested, the 2 bit cutoff yielded an average of 3.4 clusters with average entropy reduction of nearly a half (See figure 3.3). This reduction is slightly larger

than would be predicted by the power law because the first couple splits often provide greater entropy reduction. It would be possible to use a smaller cutoff, yielding more clusters, but the power law functions indicate that this would likely yield only a modest decrease in ensemble entropy.

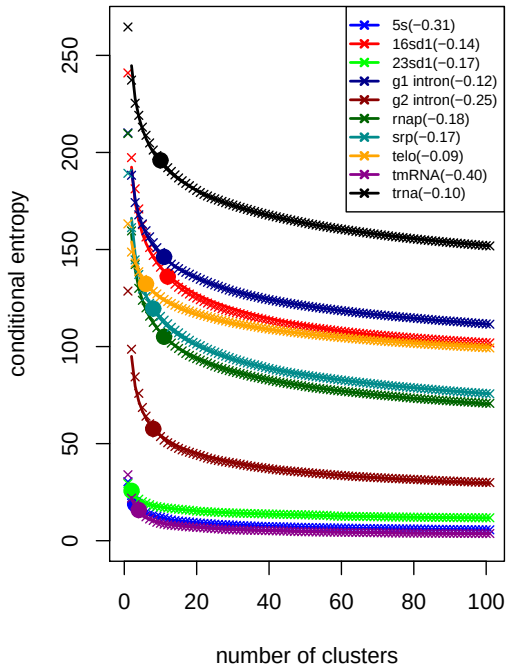

Figure 3.2: Entropy as a function of number of clusters for one sequence from each of the ten families. Power law functions of the form $y = ax^b$ are estimated by linear least squares regression after log-log transform and plotted as lines. The value of b is given parenthetically. The two bit cutoff is highlighted by a filled circle.

3.3.3 Entropy constraints and number of structures

Constraining basepairs with low entropy excludes most of the possible structures, but retains most of the probability mass. This is consistent with concentration of measure phenomena often seen in high dimensional probability distributions[103]. Basepairs with entropy less than 0.002 bits were constrained to be unpaired if they have probability near 0 and paired if they have probability near 1. For every sequence, the entropy constraints

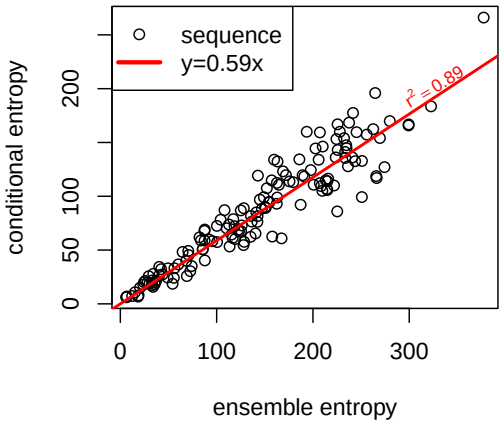

Figure 3.3: Ensemble entropy vs conditional entropy. Running the MIBP algorithm with a 2 bit cutoff yields an entropy reductions of nearly a half. For example the Chlamydomonas 5s rRNA (5s_13 in the test set) has an ensemble entropy of 49 bits, but after conditioning on the MIBPs, only 25.5 bits of uncertainty remains. Each point represents one of the sequences from one of the ten families described at the beginning of the results section.

removed less than 5% of the probability mass but resulted in about a one fourth reduction in the orders of magnitude for the total number of possible structures. For a short sequence such as *Spirocodon saltatrix* 5s rRNA (5s3_220) this means a reduction of 8 orders of magnitude from 4×10^{31} to 6×10^{23}. However for a longer sequence such as a *Saccharomyces cerevisiae* group II intron (ya1) the entropy constraints result in a reduction of almost 50 orders of magnitude: from 1×10^{172} to 9×10^{126} (see figure **??**.)

3.3.4 Cluster with native structure

Consistent with previous observations[17] the fact that a cluster contains more probability does not necessarily mean that it will contain the native structure. In fact, for the sequences tested, the probability of the native cluster is not significantly larger than the probability of a cluster chosen uniformly at random (permutation p-value = 0.47). This is likely due to the fact that secondary structure prediction algorithms struggle to provide accurate predictions for some of the longer sequences considered. If we rerun the analy-

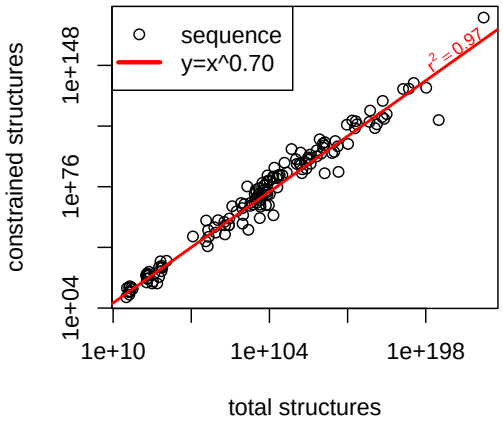

Figure 3.4: Number of structures before and after basepairs with entropy less than 0.002 bits are constrained. Constraining basepairs reduces the orders of magnitude by about one fourth. Each point represents one of the sequences from one of the ten families described at the beginning of the results section.

sis on the 309 5s RNA sequences in the RNAstructure test set, we find that the average native cluster size is 41.5% – significantly higher than the mean random cluster size of 37.0% (permutation p-value = 0.02). However it is still far short of the mean expected cluster size of 51.9%. This indicates that, at least for smaller sequences, the native structure is more likely to be found in a higher probability cluster, but that it less likely to be found in such a cluster than the Boltzmann ensemble indicates. Permutation tests were done in R using the coin package with 10,000 samples.

3.3.5 Conflicting basepairs

We find that many MIBPs are part of a pair of conflicting basepairs, but we also find that in many cases the MIBP is part of a set of more than two mutually conflicting basepairs. Each pair in the set of mutually conflicting pairs is somewhat probable on its own, but due to the no pseudoknots and no triples constraints, only one can be present in a given structure.

For 40% of MIBPs, the MIBP represents a binary choice between two basepairs. In such cases there is a conflicting base that is present in at least 90% of sampled structures that do not include the MIBP. In other cases the MIBP is one choice among a set of mutually conflicting pairs. For 84% of MIBPs, 90% of samples that do not include the MIBP include one member of a set of up to five basepairs that conflict with each other and the MIBP.

3.3.6 Mutations to the MIBP nucleotides

In this subsection we consider a 118 nucleotide 5s rRNA from the freshwater alga *Hydrurus foetidus* (5s3_71 in the test set). The MIBP algorithm finds one most informative basepair – $(17, 61)$ – dividing the Boltzmann space into two classes. 82% of structures that do not contain the MIBP contain the conflicting pair $(29, 107)$. This implies that mutating the sequence so that one of these two basepairs cannot form would bias the structure to fall into one class over the other. Indeed, editing the 17th nucleotide from a C to a A yields a Boltzmann distribution that is similar to conditioning on the absence of the MIBP. Editing the 29th position from a C to a G yields a structure that is similar to conditioning on the presence of the MIBP. However a different set of basepairs constitute one of the helices (see figure 3.5.)

3.3.7 Mutual information and RiboSNitches

Woods and Laederach[110] use SHAPE data to classify mutations into three categories based on whether they cause (i) "no differences or small differences", (ii) "local differences", or (iii) "global differences" to the RNA secondary structure. We focus on one of the sequences considered by Woods and Laedarach: a 16s rRNA 4 way junction (16SFWJ_1M7_0001 in RMDB: https://rmdb.stanford.edu/). While the ends of the MIBP (146,216) are not mutated, nearby positions that are likely to form a helix with the MIBP are mutated: a G to C mutation at position 125 causes local differences, and C to

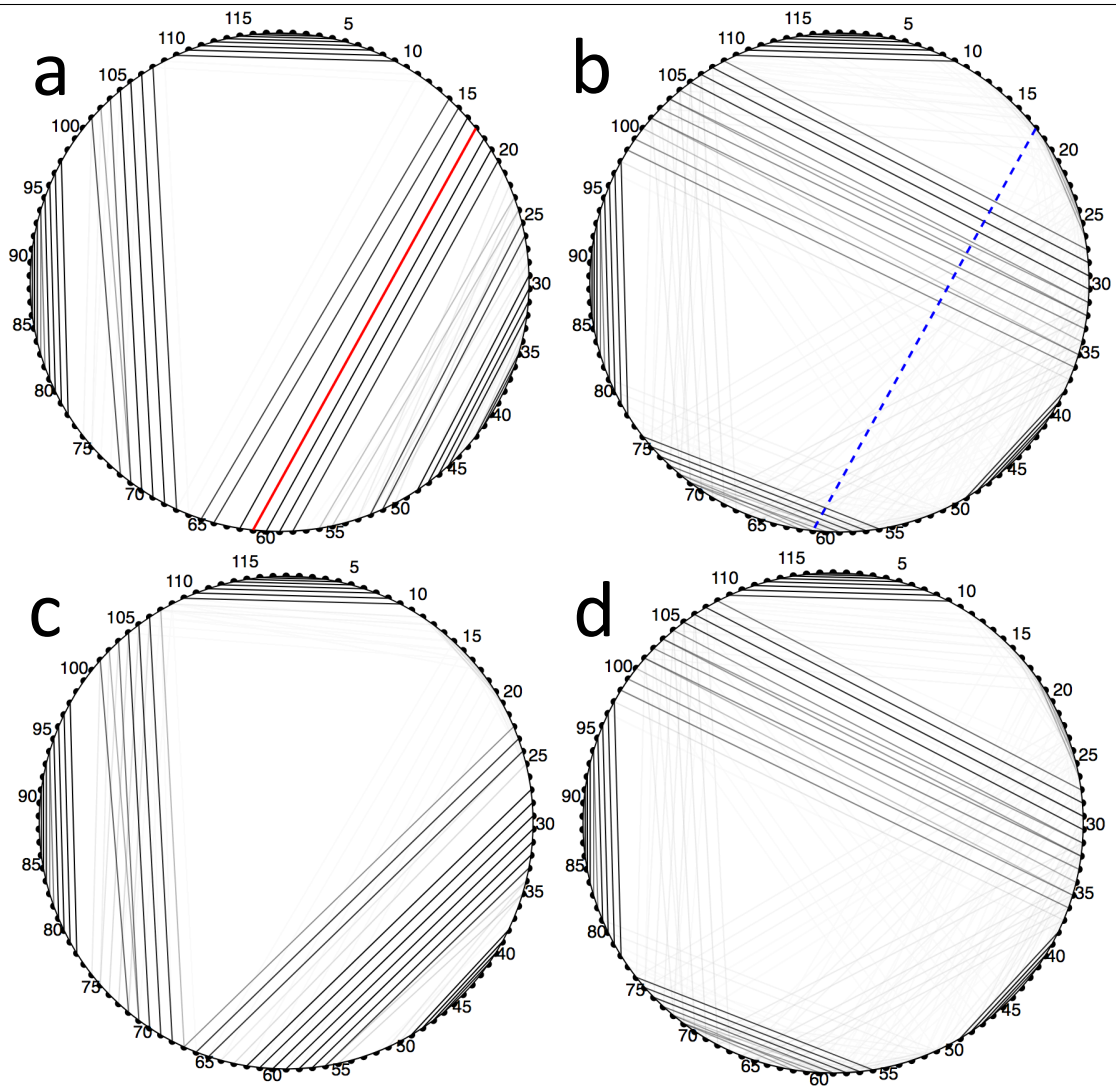

Figure 3.5: Mutating the ends of the MIBP or conflicting pair has a large effect on the resulting RNA SS. a) Basepair probabilities conditioned on the presence of MIBP. b) Basepair probabilities conditioned on the absence of MIBP. c) Basepair probabilities when conflicting pair is mutated. d) Basepair probabilities when MIBP is mutated. 5s rRNA from the freshwater alga *Hydrurus foetidus* (5s3_71 in the test set) is the sequence considered.

G mutations at positions 214 and 221 causes global differences. Most of the mutations considered (74%) cause little or no difference to the RNA SS. The third mutation that affects global change, a G to C mutation at position 177, forms the conflicting pair for one of the three additional MIBPs found with the standard 2 bit cutoff.

We also compare the mutation category to the maximum sum of mutual information for a basepair originating from the mutated position. The mean mutual information for mu-

tations that cause global changes in RNA SS (9.85 bits) is greater than the mean MI for positions local changes (7.36 bits) and much greater than the mean MI for positions that cause little or no change (3.06 bits). Figure 3.6 shows the mutual information at all the mutated positions. The html visualization for this molecule can be accessed at: http://ccmbweb.ccv.brown.edu/wmckerro/MIBP/16SFWJ_1M7_0001.html.

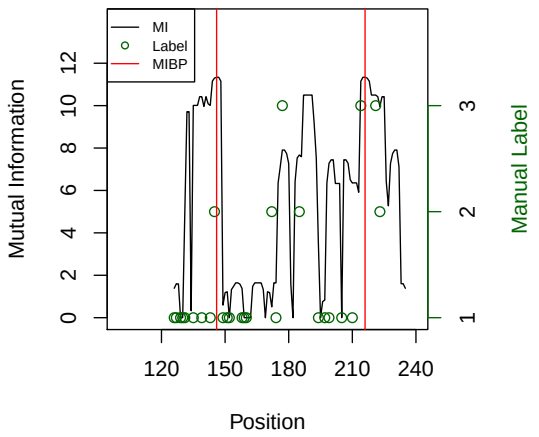

Figure 3.6: Mutual information sum and label assigned by Woods and Laederach[110] for the 16s rRNA four way junction (16SFWJ_1M7_0001 in RMDB: https://rmdb.stanford.edu/). A label of 1 indicates that the mutation causes little or no change in RNA SS. A label of 2 indicates that the mutation causes significant but primarily local changes to the RNA SS. A label of 3 indicates that the mutation causes global changes to the RNA SS.

3.3.8 A viral RNA with a key alternate conformation

Hepatitis Delta Virus (HDV) normally adopts a rod shaped configuration, but HDV Genotype III must also form a branched structure in order to undergo an essential RNA editing event[9]. We ran our MIBP algorithm on a section of the HDV Genotype III (reverse complement of GenBank: HF679406.1, nucleotides 499-1097). Most sampled structures form a rod shaped structure (figure 3.7a), but some form the branched structure described in[9] (figure 3.7b). The MIBP algorithm shows that the branched structure forms when the MIBP – $(1020, 1086)$ – is present and the rod structure forms when it is absent.

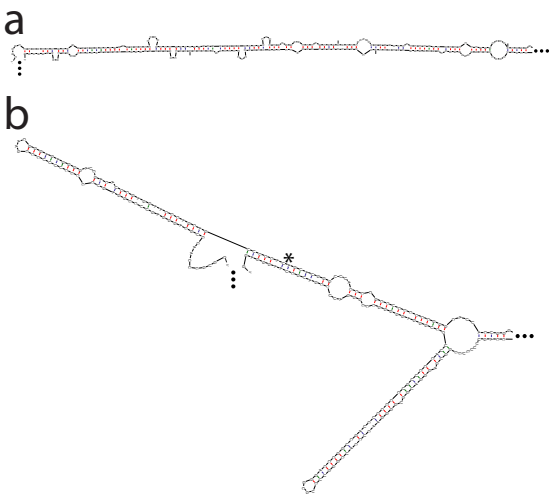

Figure 3.7: Structure predictions for Hepatitis Delta Virus Genotype III. a) Without MIBP, a rod shaped structure forms. b) With the MIBP, the branched structure described in[9] forms. The edited position is indicated with an asterisk. Structures were drawn using the mfold webserver[113].

3.4 Summary

While RNA molecules are transcribed as single stranded polymers, they readily fold into complex structures by forming internal basepairs. Many methods exist to efficiently estimate the free energy of an RNA secondary structure, but focusing only on the most stable structure fails to fully elucidate RNA structure. Instead we need to consider the RNA secondary structure Boltzmann distribution. However because the Boltzmann distribution covers all possible configurations, it is too complex to consider directly. The MIBP clustering method divides the Boltzmann space into a set of collectively exhaustive, mutually exclusive clusters that are built around the conflicting basepairs that divide the Boltzmann distribution. Importantly, the MIBP method also provides insight into which nucleotides, if mutated, will result in large scale changes to the RNA secondary structure.

4

Conclusion

Next generation sequencing technology has led to an explosion of biological sequence data. However significant challenges remain when applying this data to to the problem of understanding the role of noncoding sequence. Firstly, many noncoding sequences are repetitive, making it difficult to align short reads. Secondly, the properties of RNA molecules are intimately bound up with their secondary structures, which remains difficult to predict from sequence alone. In the first part of this thesis, I describe a profile and EM method that can surmount the ambiguous alignment challenge, providing accurate estimates of RNA editing, genomic variations and TE expression. In the final chapter, I show how mutual information can be used to understand the RNA secondary structure Boltzmann ensemble.

The applications in this thesis show the power of probabilistic modeling and Bayesian inference for sequence analysis in general and for next generation sequencing data in par-

ticular. Probabilistic modeling requires that we have an understanding of how the data was generated, but does not require us to pick out specific statistics or features. This is particularly important for the problem of ambiguous alignment to repetitive sequence. Sequence variation in transposable elements has subtle effects on the read data that can be hard to predict. However a probabilistic model allows us to compare the probability of the data when the variation is present to the probability of the data when the variation is absent, giving us a simple way to decide which variations are likely. The challenge for this type of method, especially in the era of quickly growing datasets, tends to be computational feasibility. It is often easy to write down a model, but computation within that model may not be feasible. Thus more work is always needed to streamline computation. This includes computational techniques like building efficient data structures and parallelizing when possible, mathematical techniques that simplify the expression to be calculated, and clever approximations that speed up computation without sacrificing accuracy.

References

[1] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3), 403–410.

[2] Attrill, H., Falls, K., Goodman, J. L., Millburn, G. H., Antonazzo, G., Rey, A. J., Marygold, S. J., & the FlyBase consortium (2015). Flybase: establishing a gene group resource for drosophila melanogaster. *Nucleic Acids Research*.

[3] Benjamini, Y. & Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4), 1165–1188.

[4] Bernstein, E., Caudy, A. A., Hammond, S. M., & Hannon, G. J. (2001). Role for a bidentate ribonuclease in the initiation step of rna interference. *Nature*, 409(6818), 363–366.

[5] Bray, N. L., Pimentel, H., Melsted, P., & Pachter, L. (2016). Near-optimal probabilistic rna-seq quantification. *Nature Biotechnology*, 34, 525 EP –.

[6] Brown, J. W., Haas, E. S., Gilbert, D. G., & Pace, N. R. (1994). The Ribonuclease P database. *Nucleic Acids Research*, 22(17), 3660–3662.

[7] Burns, K. H. (2017). Transposable elements in cancer. *Nature Reviews Cancer*, 17, 415 EP –.

[8] Carrillo, R. A., Özkan, E., Menon, K. P., Nagarkar-Jaiswal, S., Lee, P. T., Jeon, M., Birnbaum, M. E., Bellen, H. J., Garcia, K. C., & Zinn, K. (2015). Control of synaptic connectivity by a network of drosophila igsf cell surface proteins. *Cell*, 163(7), 1770–1782.

[9] Casey, J. L. (2002). RNA editing in hepatitis delta virus genotype III requires a branched double-hairpin rna structure. *Journal of Virology*, 76(15), 7385–7397.

[10] Chan, C. Y., Lawrence, C. E., & Ding, Y. (2005). Structure clustering features on the Sfold web server. *Bioinformatics*, 21(20), 3926–3928.

[11] Chen-Plotkin, A. S., Lee, V. M.-Y., & Trojanowski, J. Q. (2010). Tar dna-binding protein 43 in neurodegenerative disease. *Nature reviews. Neurology*, 6(4), 211–220.

[12] Cover, T. & Thomas, J. (1991). *Elements of Information Theory*. Hoboken, NJ: John Wiley.

[13] Cox, D. B. T., Gootenberg, J. S., Abudayyeh, O. O., Franklin, B., Kellner, M. J., Joung, J., & Zhang, F. (2017). RNA editing with CRISPR-Cas13. *Science*.

[14] Cridland, J. M., Thornton, K. R., & Long, A. D. (2015). Gene expression variation in drosophila melanogaster due to rare transposable element insertion alleles of large effect. *Genetics*, 199(1), 85–93.

[15] Damberger, S. H. & Gutell, R. R. (1994). A comparative database of group I intron structures. *Nucleic Acids Research*, 22(17), 3508–3510.

[16] de Koning, A. P. J., Gu, W., Castoe, T. A., Batzer, M. A., & Pollock, D. D. (2011). Repetitive elements may comprise over two-thirds of the human genome. *PLoS Genetics*, 7(12), e1002384.

[17] Ding, Y., Chan, C. Y., & Lawrence, C. E. (2005). RNA secondary structure prediction by centroids in a boltzmann weighted ensemble. *RNA*, 11(8), 1157–1166.

[18] Ding, Y., Chan, C. Y., & Lawrence, C. E. (2006). Clustering of RNA secondary structures with application to messenger RNAs. *Journal of Molecular Biology*, 359(3), 554–571.

[19] Ding, Y. & Lawrence, C. E. (2003). A statistical sampling algorithm for RNA secondary structure prediction secondary structure prediction. *Nucleic Acids Research*, 31(24), 7280–7301.

[20] Durbin, R. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

[21] Efron, B. & Hinkley, D. V. (1978). Assessing the accuracy of the maximum likelihood estimator: Observed versus expected fisher information. *Biometrika*, 65(3), 457–483.

[22] Eggington, J. M., Greene, T., & Bass, B. L. (2011). Predicting sites of adar editing in double-stranded rna. *Nature communications*, 2, 319.

[23] Erwin, J. A., Marchetto, M. C., & Gage, F. H. (2014). Mobile dna elements in the generation of diversity and complexity in the brain. *Nature reviews. Neuroscience*, 15(8), 497–506.

[24] Feschotte, C. (2008). The contribution of transposable elements to the evolution of regulatory networks. *Nature reviews. Genetics*, 9(5), 397–405.

[25] Fire, A., Xu, S., Montgomery, M. K., Kostas, S. A., Driver, S. E., & Mello, C. C. (1998). Potent and specific genetic interference by double-stranded rna in caenorhabditis elegans. *Nature*, 391, 806 EP –.

[26] Gardner, E. J., Lam, V. K., Harris, D. N., Chuang, N. T., Scott, E. C., Pittard, W. S., Mills, R. E., Consortium, T. . G. P., & Devine, S. E. (2017). The mobile element locator tool (melt): population-scale mobile element discovery and biology. *Genome Research*, 27(11), 1916–1929.

[27] Giegerich, R., Voß, B., & Rehmsmeier, M. (2004). Abstract shapes of RNA. *Nucleic Acids Research*, 32(16), 4843–4851.

[28] Gutell, R. R. (1994). Collection of small subunit (16s- and 16s-like) ribosomal RNA structures. *Nucleic Acids Research*, 22(17), 3502–3507.

[29] Gutell, R. R., Gray, M. W., & Schnare, M. N. (1993). A compilation of large subunit (23s and 23s-like) ribosomal RNA structures. *Nucleic Acids Research*, 21(13), 3055–3074.

[30] Gutell, R. R., Power, A., Hertz, G. Z., Putz, E. J., & Stormo, G. D. (1992). Identifying constraints on the higher-order structure of RNA: continued development and application of comparative sequence analysis methods. *Nucleic Acids Research*, 20(21), 5785–5795.

[31] Halvorsen, M., Martin, J. S., Broadaway, S., & Laederach, A. (2010). Disease-associated mutations that alter the RNA structural ensemble. *PLoS Genetics*, 6(8), e1001074.

[32] Hancks, D. C. & Kazazian, H. H. (2016). Roles for retrotransposon insertions in human disease. *Mobile DNA*, 7, 9.

[33] Harris, W. A. & Stark, W. S. (1977). Hereditary retinal degeneration in drosophila melanogaster. a mutant defect associated with the phototransduction process. *The Journal of general physiology*, 69(3), 261–91.

[34] Havecker, E. R., Gao, X., & Voytas, D. F. (2004). The diversity of ltr retrotransposons. *Genome Biology*, 5(6), 225.

[35] Hofacker, I. L. (2003). Vienna RNA secondary structure server. *Nucleic Acids Research*, 31(13), 3429–3431.

[36] Hsu, K., Chang, D.-Y., & Maraia, R. J. (1995). Human signal recognition particle (srp) alu-associated protein also binds alu interspersed repeat sequence rnas characterization of human srp9. *Journal of Biological Chemistry*, 270(17), 10179–10186.

[37] Huang, W., Massouras, A., Inoue, Y., Peiffer, J., Ràmia, M., Tarone, A. M., Turlapati, L., Zichner, T., Zhu, D., Lyman, R. F., Magwire, M. M., Blankenburg, K., Carbone, M. A., Chang, K., Ellis, L. L., Fernandez, S., Han, Y., Highnam, G., Hjelmen, C. E., Jack, J. R., Javaid, M., Jayaseelan, J., Kalra, D., Lee, S., Lewis, L., Munidasa, M., Ongeri, F., Patel, S., Perales, L., Perez, A., Pu, L., Rollmann, S. M., Ruth, R., Saada, N., Warner, C., Williams, A., Wu, Y.-Q., Yamamoto, A., Zhang, Y., Zhu, Y., Anholt, R. R., Korbel, J. O., Mittelman, D., Muzny, D. M., Gibbs, R. A., Barbadilla, A., Johnston, J. S., Stone, E. A., Richards, S., Deplancke, B., & Mackay, T. F. (2014). Natural variation in genome architecture among 205 drosophila melanogaster genetic reference panel lines. *Genome Research*, 24(7), 1193–1208.

[38] Janssen, S. & Giegerich, R. (2010). Faster computation of exact RNA shape probabilities. *Bioinformatics*, 26(5), 632–639.

[39] Jin, P., Duan, R., Qurashi, A., Qin, Y., Tian, D., Rosser, T. C., Liu, H., Feng, Y., & Warren, S. T. (2007). Pur a binds to rcgg repeats and modulates repeat-mediated neurodegeneration in a drosophila model of fragile x tremor/ataxia syndrome. *Neuron*, 55(4), 556–564.

[40] Jin, Y., Tam, O. H., Paniagua, E., & Hammell, M. (2015). Tetranscripts: a package for including transposable elements in differential expression analysis of rna-seq datasets. *Bioinformatics*, 31(22), 3593–3599.

[41] Kaplan, E. L. & Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282), 457–481.

[42] Kazazian Jr, H. H., Wong, C., Youssoufian, H., Scott, A. F., Phillips, D. G., & Antonarakis, S. E. (1988). Haemophilia a resulting from de novo insertion of l1 sequences represents a novel mechanism for mutation in man. *Nature*, 332, 164 EP –.

[43] Keane, T. M., Wong, K., & Adams, D. J. (2013). Retroseq: transposable element discovery from next-generation sequencing data. *Bioinformatics*, 29(3), 389–390.

[44] Kent, W. J. (2002). Blat—the blast-like alignment tool. *Genome Research*, 12(4), 656–664.

[45] Kim, D. D., Kim, T. T., Walsh, T., Kobayashi, Y., Matise, T. C., Buyske, S., & Gabriel, A. (2004). Widespread rna editing of embedded alu elements in the human transcriptome. *Genome Research*, 14, 1719–1725.

[46] Kim, M., Semple, I., Kim, B., Kiers, A., Nam, S., Park, H. W., Park, H., Ro, S. H., Kim, J. S., Juhász, G., & Lee, J. H. (2015). Drosophila gyf/grb10 interacting gyf protein is an autophagy regulator that controls neuron and muscle homeostasis. *Autophagy*, 11(8), 1358–1372.

[47] Koonin, E. V. & Krupovic, M. (2014). Evolution of adaptive immunity from transposable elements combined with innate immune systems. *Nature Reviews Genetics*, 16, 184 EP –.

[48] Krug, L., Chatterjee, N., Borges-Monroy, R., Hearn, S., Liao, W.-W., Morrill, K., Prazak, L., Rozhkov, N., Theodorou, D., Hammell, M., & Dubnau, J. (2017). Retrotransposon activation contributes to neurodegeneration in a drosophila tdp-43 model of als. *PLoS Genetics*, 13(3), e1006635.

[49] Kumar, J. P., Hsiung, F., Powers, M. A., & Moses, K. (2003). Nuclear translocation of activated map kinase is developmentally regulated in the developing drosophila eye. *Development*, 130(16), 3703–14.

[50] Kurusu, M., Cording, A., Taniguchi, M., Menon, K., Suzuki, E., & Zinn, K. (2008). A screen of cell-surface molecules identifies leucine-rich repeat proteins as key mediators of synaptic target selection. *Neuron*, 59(6), 972–985.

[51] Lackey, L., Coria, A., Woods, C., McArthur, E., & Laederach, A. (2018). Allele-specific shape-map assessment of the effects of somatic variation and protein binding on mrna structure. *RNA*.

[52] Langmead, B., Trapnell, C., Pop, M., & Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biology*, 10(3), R25.

[53] Larsen, N. & Zwieb, C. (1996). The signal recognition particle database (SRPDB). *Nucleic Acids Research*, 24(1), 80–81.

[54] Lasda, E. & Parker, R. (2014). Circular rnas: diversity of form and function. *RNA*, 20(12), 1829–1842

[55] Lee, H. K., Cording, A., Vielmetter, J., & Zinn, K. (2013). Interactions between a receptor tyrosine phosphatase and a cell surface ligand regulate axon guidance and glial-neuronal communication. *Neuron*, 78(5), 813–826.

[56] Lee, S.-I. & Kim, N.-S. (2014). Transposable elements and genome size variations in plants. *Genomics & Informatics*, 12(3), 87–97.

[57] Lehmann, K. A. & Bass, B. L. (1999). The importance of internal loops within rna substrates of adar1. *Journal of Molecular Biology*, 291, 1–13.

[58] Lehmann, K. A. & Bass, B. L. (2000). Double-stranded rna adenosine deaminases adar1 and adar2 have overlapping specificities. *Biochemistry*, 39(42), 12875–12884.

[59] Li, H. & Durbin, R. (2009a). Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, 25(14), 1754–1760.

[60] Li, H. & Durbin, R. (2009b). Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, 25(14), 1754–1760.

[61] Li, W., Lee, M.-H., Henderson, L., Tyagi, R., Bachani, M., Steiner, J., Campanac, E., Hoffman, D. A., von Geldern, G., Johnson, K., Maric, D., Morris, H. D., Lentz, M., Pak, K., Mammen, A., Ostrow, L., Rothstein, J., & Nath, A. (2015). Human endogenous retrovirus-k contributes to motor neuron disease. *Science Translational Medicine*, 7(307), 307ra153.

[62] Lin, L., McKerrow, W. H., Richards, B., Phonsom, C., & Lawrence, C. E. (2018). Characterization and visualization of rna secondary structure boltzmann ensemble via information theory. *BMC Bioinformatics*, 19(1), 82.

[63] Lu, Z. J., Gloor, J. W., & Mathews, D. H. (2009). Improved rna secondary structure prediction by maximizing expected pair accuracy. *RNA*, 15(10), 1805–1813.

[64] Mackay, T. F. C., Richards, S., Stone, E. A., Barbadilla, A., Ayroles, J. F., Zhu, D., Casillas, S., Han, Y., Magwire, M. M., Cridland, J. M., Richardson, M. F., Anholt, R. R. H., Barrón, M., Bess, C., Blankenburg, K. P., Carbone, M. A., Castellano, D., Chaboub, L., Duncan, L., Harris, Z., Javaid, M., Jayaseelan, J. C., Jhangiani, S. N., Jordan, K. W., Lara, F., Lawrence, F., Lee, S. L., Librado, P., Linheiro, R. S., Lyman, R. F., Mackey, A. J., Munidasa, M., Muzny, D. M., Nazareth, L., Newsham, I., Perales, L., Pu, L.-L., Qu, C., Ràmia, M., Reid, J. G., Rollmann, S. M., Rozas, J., Saada, N., Turlapati, L., Worley, K. C., Wu, Y.-Q., Yamamoto, A., Zhu, Y., Bergman, C. M., Thornton, K. R., Mittelman, D., & Gibbs, R. A. (2012). The drosophila melanogaster genetic reference panel. *Nature*, 482, 173 EP –.

[65] Mathews, D. H. (2004). Using an rna secondary structure partition function to determine confidence in base pairs predicted by free energy minimization. *RNA*, 10(8), 1178–1190.

[66] Mathews, D. H., Disney, M. D., Childs, J. L., Schroeder, S. J., Zuker, M., & Turner, D. H. (2004). Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences of the United States of America*, 101(19), 7287–7292.

[67] McClintock, B. (1951). Chromosome organization and genic expression. volume 16 (pp. 13–47).: Cold Spring Harbor Laboratory Press.

[68] McKerrow, W. H., Savva, Y. A., Rezaei, A., Reenan, R. A., & Lawrence, C. E. (2017). Predicting rna hyper-editing with a novel tool when unambiguous alignment is impossible. *BMC Genomics*, 18(1), 522.

[69] Merino, C., Penney, J., González, M., Tsurudome, K., Moujahidine, M., O'Connor, M. B., Verheyen, E. M., & Haghighi, P. (2009). Nemo kinase interacts with mad to coordinate synaptic growth at the drosophila neuromuscular junction. *Journal of Cell Biology*, 185(4), 713–725.

[70] Meyer, S., Schmidt, I., & Klämbt, C. (2014). Glia ecm interactions are required to shape the drosophila nervous system. *Mechanisms of Development*, 133, 105–116.

[71] Michel, F., Kazuhiko, U., & Haruo, O. (1989). Comparative and functional anatomy of group II catalytic introns – a review. *Gene*, 82(1), 5–30.

[72] Nahm, M., Kim, S., Paik, S. K., Lee, M., Lee, S., Lee, Z. H., Kim, J., Lee, D., Bae, Y. C., & Lee, S. (2010). dcip4 (drosophila cdc42-interacting protein 4) restrains synaptic growth by inhibiting the secretion of the retrograde glass bottom boat signal. *Journal of Neuroscience*, 30(24), 8138–8150.

[73] Needleman, S. B. & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3), 443–453.

[74] Nishikura, K., Yoo, C., Kim, U., Murray, J. M., Estes, P. A., Cash, F. E., & Liebhaber, S. A. (1991). Substrate specificity of the dsrna unwinding/modifying activity. *The EMBO Journal*, 10(11), 3523–3532.

[75] Nudler, E. & Mironov, A. S. (2004). The riboswitch control of bacterial metabolism. *Trends in biochemical sciences*, 29(1), 11–17.

[76] O'Connell, M. A., Mannion, N. M., & Keegan, L. P. (2015). The epitranscriptome and innate immunity. *PLOS Genetics*, 11(12), e1005687.

[77] Orr, W. C. (2016). Tightening the connection between transposable element mobilization and aging. *Proceedings of the National Academy of Sciences*, 113(40), 11069–11070.

[78] Patro, R., Duggal, G., Love, M. I., Irizarry, R. A., & Kingsford, C. (2017). Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods*, 14, 417 EP –.

[79] Patro, R., Mount, S. M., & Kingsford, C. (2014). Sailfish enables alignment-free isoform quantification from rna-seq reads using lightweight algorithms. *Nature biotechnology*, 32(5), 462–464.

[80] Platzer, A., Nizhynska, V., & Long, Q. (2012). Te-locate: A tool to locate and group transposable element occurrences using paired-end next-generation sequencing data. *Biology*, 1(2), 395–410.

[81] Podlevsky, J. D., Bley, C. J., Omana, R. V., Qi, X., & Chen, J. J.-L. (2008). The telomerase database. *Nucleic Acids Research*, 36(Database issue), D339–D343.

[82] Polson, A. G. & Bass, B. L. (1994). Preferential selection of adenosines for modification by double-stranded rna adenosine deaminase. *The Embo Journal*, 12(23), 5701–5711.

[83] Porath, H. T., Carmi, S., & Levanon, E. Y. (2014). A genome-wide map of hyper-edited rna reveals numerous new sites. *Nature Communications*, 5(4726).

[84] Quinlan, J. R. (1986). Induction of decision trees. *Mach. Learn.*, 1(1), 81–106.

[85] Ramaswami, G., Zhang, R., Piskol, R., Keegan, L. P., Deng, P., O'Connell, M. A., & Li, J. B. (2013). Identifying rna editing sites using rna sequencing data alone. *Nature Methods*, 10(2), 128–131.

[86] Reilly, M. T., Faulkner, G. J., Dubnau, J., Ponomarev, I., & Gage, F. H. (2013). The role of transposable elements in health and diseases of the central nervous system. *The Journal of Neuroscience*, 33(45), 17577–17586.

[87] Reuter, J. S. & Mathews, D. H. (2010). RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics*, 11, 129–129.

[88] Robb, S. M. C., Lu, L., Valencia, E., Burnette, J. M., Okumoto, Y., Wessler, S. R., & Stajich, J. E. (2013). The use of relocate and unassembled short reads to produce high-resolution snapshots of transposable element generated diversity in rice. *G3: Genes|Genomes|Genetics*, 3(6), 949–957.

[89] Rodriguez, J., Menet, J. S., & Rosbash, M. (2012). Nascent-seq indicates widespread cotranscriptional rna editing in drosophila. *Molecular Cell*, 47, 27–37.

[90] Rogers, E. & Heitsch, C. E. (2014). Profiling small RNA reveals multimodal substructural signals in a Boltzmann ensemble. *Nucleic Acids Research*, 42(22), e171–e171.

[91] Saldi, T. K., Ash, P. E., Wilson, G., Gonzales, P., Garrido-Lecca, A., Roberts, C. M., Dostal, V., Gendron, T. F., Stein, L. D., Blumenthal, T., Petrucelli, L., & Link, C. D. (2014). Tdp-1, the caenorhabditis elegans ortholog of tdp-43, limits the accumulation of double-stranded rna. *The EMBO Journal*, 33(24), 2947–2966.

[92] Savva, Y. A., Jepson, J. E. C., Chang, Y.-J., Whitaker, R., Jones, B. C., StLaurent, G., Tackett, M. R., Kapranov, P., Jiang, N., Du, G., Helfand, S. L., & Reenan, R. A. (2013). Rna editing regulates transposon-mediated heterochromatic gene silencing. *Nature Communications*, 4, 2745.

[93] Savva, Y. A., Rezaei, A., StLaurent, G., & Reenan, R. A. (2016). Reprogramming, circular reasoning and self versus non-self: One-stop shopping with rna editing. *Frontiers in Genetics*, 7, 100.

[94] Slotkin, R. K. & Martienssen, R. (2007). Transposable elements and the epigenetic regulation of the genome. *Nature Reviews Genetics*, 8, 272 EP –.

[95] Smith, K. R., Oliver, P. L., Lumb, M. J., Arancibia-Carcamo, I. L., Revilla-Sanchez, R., Brandon, N. J., Moss, S. J., & Kittler, J. T. (2010). Identification and characterisation of a maf1/macoco protein complex that interacts with gabaa receptors in neurons. *Mol Cell Neurosci*, 44(4), 330–41.

[96] Smith, T. F. & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1), 195–197.

[97] Solem, A. C., Halvorsen, M., Ramos, S. B. V., & Laederach, A. (2015). The potential of the ribosnitch in personalized medicine. *Wiley Interdisciplinary Reviews: RNA*, 6(5), 517–532.

[98] Song, J., Wu, L., Chen, Z., Kohanski, R. A., & Pick, L. (2003). Axons guided by insulin receptor in drosophila visual system. *Science*, 300(5618), 502–505.

[99] Sprinzl, M. & Vassilenko, K. S. (2005). Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Research*, 33(Database Issue), D139–D140.

[100] Steffen, P., Voß, B., Rehmsmeier, M., Reeder, J., & Giegerich, R. (2006). RNAshapes: an integrated RNA analysis package based on abstract shapes. *Bioinformatics*, 22(4), 500–503.

[101] StLaurent, G., Tackett, M. R., Nechkin, S., Shtokalo, D., Antonets, D., Savva, Y. A., Maloney, R., Kapranov, P., Lawrence, C. E., & Reenan, R. A. (2013). Genome-wide analysis of a-to-i rna editing by single-molecule sequencing in drosophila. *Nature Structural and Molecular Biology*, 20(11), 1333–1339.

[102] Szymanski, M., Specht, T., Barciszewska, M. Z., Barciszewski, J., & Erdmann, V. A. (1998). 5s rRNA data bank. *Nucleic Acids Research*, 26(1), 156–159.

[103] Talagrand, M. (1996). A new look at independence. *The Annals of Probability*, 24(1), 1–34.

[104] Wan, Y., Qu, K., Zhang, Q. C., Flynn, R. A., Manor, O., Ouyang, Z., Zhang, J., Spitale, R. C., Snyder, M. P., & Segal, E. (2014). Landscape and variation of RNA secondary structure across the human transcriptome. *Nature*, 505(7485), 706–709

[105] Wang, J., Lee, C. H. J., Lin, S., & Lee, T. (2006). Steroid hormone-dependent transformation of polyhomeotic mutant neurons in the drosophila brain. *Development*, 133(7), 1231–1240.

[106] Wang, J., Raskin, L., Samuels, D. C., Shyr, Y., & Guo, Y. (2015). Genome measures used for quality control are dependent on gene function and ancestry. *Bioinformatics*, 31(3), 318–323.

[107] Whipple, J. M., Youssef, O. A., Aruscavage, P. J., Nix, D. A., Hong, C., Johnson, W. E., & Bass, B. L. (2015). Genome-wide profiling of the c. elegans dsrnaome. *RNA*, 21(5), 786–800.

[108] Wood, J. G. & Helfand, S. L. (2013). Chromatin structure and transposable elements in organismal aging. *Frontiers in Genetics*, 4, 274.

[109] Wood, J. G., Jones, B. C., Jiang, N., Chang, C., Hosier, S., Wickremesinghe, P., Garcia, M., Hartnett, D. A., Burhenn, L., Neretti, N., & Helfand, S. L. (2016). Chromatin-modifying genetic interventions suppress age-associated transposable element activation and extend life span in drosophila. *Proceedings of the National Academy of Sciences of the United States of America*, 113(40), 11277–11282.

[110] Woods, C. T. & Laederach, A. (2017). Classification of rna structure change by 'gazing'at experimental data. *Bioinformatics*, 33(11), 1647–1655.

[111] Yu, W., Kawasaki, F., & Ordway, R. W. (2011). Activity-dependent interactions of nsf and snap at living synapses. *Molecular and Cellular Neurosciences*, 47(1), 19–27.

[112] Zhang, L. & Rong, Y. S. (2012). Retrotransposons at drosophila telomeres: host domestication of a selfish element for the maintenance of genome integrity. *Biochimica et Biophysica Acta*, 1819(7), 771–775.

[113] Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13), 3406–3415.

[114] Zwieb, C., Gorodkin, J., Knudsen, B., Burks, J., & Wower, J. (2003). tmRDB (tmRNA database). *Nucleic Acids Research*, 31(1), 446–447.

This thesis was typeset using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 11 point Egenolff-Berner Garamond, a revival of Claude Garamont's humanist typeface. The above illustration, "Science Experiment 02", was created by Ben Schlitter and released under CC BY-NC-ND 3.0. A template that can be used to format a PhD thesis with this look and feel has been released under the permissive MIT (X11) license, and can be found online at github.com/suchow/Dissertate or from its author, Jordan Suchow, at suchow@post.harvard.edu.