NTNU – Trondheim
Norwegian University of
Science and Technology

Studying differential isomiRs in high-throughput sequencing data identifies miRNA end-reads as a novel, putative miRNA degradation product

Jan-Preben Silver Mossin

Master of Science in Computer Science
Submission date: June 2014
Supervisor: Pål Sætrom, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Jan-Preben Silver Mossin

Supervisor: **Pål Sætrom**

Studying differential isomiRs in high-throughput sequencing data identifies miRNA end-reads as a novel, putative miRNA degradation product

Master of Science in Computer Science
Submission date: June 2014

Algorithms, HPC and Graphics Group
Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering

Abstract

With the advancement of deep-sequencing technologies, it has become clear that individual miRNAs show varying degrees of heterogeneity in the expressed mature sequence. Recent evidence has suggested isomiR variants may be biologically relevant, and that small sequence variations can affect e.g. miRNA stability and RISC loading.

The goal of this work was to investigate and characterize such isomiRs to determine whether isomiRs of the same miRNA show biologically relevant differences. The work can further be divided into two sub-studies.

I first present a support-vector machine classifier for predicting expression changes for isomiRs in Ago2-knockout cells. The classifier is constructed from miRNA sequence features, and is trained and tested using the observed read data for individual isomiRs, thus handling the possibility that highly similar isomiRs can experience different expression changes. The classifier is not successful, suggesting expression changes in Ago2-knockout is determined by other factors. I also look more closely at whether there are isomiR variants that experience significantly different expression changes, but find no strong evidence of this.

Working with the Ago2-knockout sequencing data, a second line of work was set off from the discovery of a group of short $\sim$ 10 nt reads that align to the 3' end of mature miRNAs. I initially searched for such reads with the goal of finding products of Ago2 cleavage, but interestingly the reads are found in both wildtype and Ago2-knockout samples. I present an analysis of these reads and their corresponding miRNAs, and suggest they are produced by a previously undescribed regulated miRNA degradation process. In light of this, I also study the project data for isomiRs with non-templated 3' A/U-tails, which previously has been reported to affect miRNA degradation. I find both A- and U-tailing is common in the studied samples, but find neither a strong correlation nor a lack of overlap between miRNAs targeted by 3' tailing and miRNAs with corresponding short 3'-aligning reads.

Preface

I would like to thank my supervisor Pål Sætrom for valuable help and guidance throughout the semester, and for introducing me to the exciting field of microRNA research.

Jan-Preben Silver Mossin
Bergen, June 10, 2014

Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

MicroRNAs (miRNA) are a group of short non-coding RNA molecules that since their discovery in the early 2000s have been found to play a key role in post-transcriptional gene-regulation [Bartel, 2009]. After being processed from longer precursor sequences, mature ~ 22 nt miRNAs are loaded by a multiprotein complex called the RNA-induced silencing complex (RISC), where miRNAs act as guide molecules by binding to target messenger RNA (mRNA) of partial complementarity. Having been guided to a target, RISC can silence the gene either by site-specific cleavage, mRNA degradation or translational repression.

RISC silencing of mRNAs is carried out by a member of the Ago protein family. It is also an Ago protein that binds and loads the miRNA guide, and a minimal RISC molecule has been reported to consist of only a single Ago and the bound miRNA [Cenik and Zamore, 2011]. Many species encode multiple variants of Ago proteins, some of which are known to be functionally distinct. In mammals there are four Agos, Ago1-4, and all four can act as the Ago component of RISC. Of these, Ago2 is known to be functionally distinct, being the only mammalian Ago with slicer activity, and is vital for animal development. But the significance of Ago1, Ago3 and Ago4 is not well understood [Wang et al., 2012].

The importance of Ago2 for miRNA activity was recently studied in a deep-sequencing experiment conducted at NTNU's Department of Cancer Research and Molecular Medicine, where the expression of short RNA sequences were measured for mouse Ago2-knockout and wildtype cells. In [Mossin, 2013], I worked with a dataset derived from this project that contained expression levels and knockout fold-changes for miRNAs, and tried to create a classifier for predicting whether a miRNA, based on features of its sequence, would be up- or down-regulated in Ago2-knockout cells. The sequence features were computed using reference miRNA sequence data (i.e. each miRNA was represented by its reference sequence), and although the classifier seemed to have some success, I argued that a better result could likely be obtained by taking advantage of the fact that most miRNAs show some degree of sequence heterogeneity.

With the development of deep-sequencing technologies, it has become clear that there is no single sequence representing each mature miRNA, but that there often are several expressed variants of the mature miRNA product [Neilsen et al., 2012], differing in which nucleotides are included from the precursor sequence at the 5' and/or 3' end. These sequence variants are called isomiRs. While isomiRs have been shown to be differentially expressed between cell-types, and some reports have indicated isomiR sequence variations may affect mRNA targeting, sequence stability and RISC loading, it is currently not clear to what degree the expression of isomiR variants is a regulated process [Neilsen et al., 2012]. A goal for my project has been to work with sequencing data at the isomiR level, taking into consideration, and also investigating, the possibility that isomiRs may have relevant biological differences.

Part of this report will describe my attempt at constructing a support-vector machine classifier for predicting up/down-regulation in Ago2-knockout cells, where instead of representing each miRNA by a single reference sequence I will take observed isomiRs into consideration. Specifically, the analysis starting point will be the raw sequencing read output, and each isomiR sequence will mostly be treated individually. In summary, the classifier is not successful, which suggests Ago2-knockout expression changes are not determined by sequence properties. But an interesting observation is that some isomiRs seem to experience quite different knockout expression changes, and so I spend some time trying to find common features that differ between these otherwise very similar sequences. This analysis also give no strong results, and I discuss that the difference seen between isomiRs of the same miRNA appear to be due to a natural variation.

In addition to the work on differential expression, I also used the sequencing data from the Ago2 experiment to try to find Ago2-cleavage products; short ~ 10 nt sequences that align to the 3' or 5' end of mature miRNAs. The initial goal was to try to see if Ago2 targets specific miRNAs (or isomiRs) for cleavage, and find common features among the targets. The search revealed a number of short reads aligning to the 3' end of mature miRNA sequences (for the sake of convenience I will denote these reads as “end-reads” below), but interestingly, the reads are found both in wildtype and Ago2-knockout samples. This implies that the end-reads cannot have been produced by Ago2-cleavage. The natural suggestion is then that they result from a different miRNA degradation process. In light of this, I also study the sequencing data for a previously reported mechanism affecting miRNA degradation, namely non-templated 3' additions of A- or U tails. If end-reads and A/U additions represent different degradation mechanisms, it could be expected that individual miRNAs are targeted primarily by only one of the respective processes, but I find no evidence of this.

There are extensive differences in both which miRNAs have corresponding end-reads, and which miRNAs experience non-templated 3' additions. And interestingly, in both cases there are also significant differences between isomiRs of the same miRNA. I have performed a statistical analysis to try to find sequence features that cause the observed differences. For end-reads I find no such features, while whether a miRNA experiences non-templated 3' additions seems to be affected by the 3' end sequence composition.

The outline of this report is as follows. In chapter 2 I give an introduction to the theory underlying the work of the report, which include topics from both biology and computer science. Chapter 3 gives a brief summary of previous studies related to my work. Chapter 4 first gives a description of the project data, before describing the main tools and techniques used to derive the results of the project, which are presented in chapter 5. The experiments and results of chapter 5 can broadly be grouped in two: those dealing with differential expression in Ago2-knockout, presented in sections 5.1 and 5.2, and those dealing with end-reads and non-templated isomiRs, presented in sections 5.3 and 5.4. Finally, in chapter 6 I give a summary of the presented work and results, and discuss possible areas for future work.

CHAPTER 2

Background

This chapter gives an introduction to the main background material necessary for an understanding of the work presented, and basic knowledge of the various topics covered here will later be assumed. Bioinformatics is by nature an interdisciplinary field, and this chapter will cover a wide range of topics. I start with giving a brief and general biology introduction, before discussing microRNAs in some more depth in section 2.2. For the biologist, the first section will be elementary, but I have included it to give someone with only a computer science background a better understanding of the later material. To give some context on the nature of the datasets used, section 2.3 gives an introduction to the basics of RNA-sequencing. Section 2.4 covers some techniques for processing the output of an RNA-sequencing exeriment, including aligning the reads to a genome. As much of this report deals with analysing differentially expressed miRNAs, I discuss statistical methods for determining differential expression in section 2.5. Finally, in section 2.6 I give an introduction to classification using support-vector machines, and discuss how the performance of a classifier can be estimated.

2.1 Cells, DNA, RNA, and proteins

All living beings are made up of cells, which are the smallest replicating units of life [Sung, 2011]. From single-celled organisms to complex animals, cells are the basic functional and structural units; they perform the processes necessary for life, and are the building blocks for more complex structures within multicellular organisms. Each cell of an organism contains the complete genome of the organism stored as double stranded DNA. DNA molecules are built from chains of four smaller molecules called nucleotides. These nucleotides are known as adenine (A), cytosine (C), guanine (G) and thymine (T), and DNA sequences can be represented as strings over the alphabet {A, C, G, T}. In a DNA double helix an adenine base will only bind to a thymine base from the other strand (and vice versa), and cytosine will only bind with guanine, so that knowing one strand completely determines the other, meaning the two strands code the same genetic information. An example double-stranded DNA molecule is shown in Figure 2.1. During cell division, the process by which a cell divides into two children cells, the strands are unfolded and replicated to give each child an identical copy of the genome.

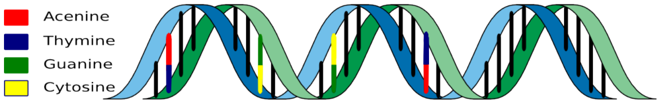


Figure 2.1: A simple illustration of a double-stranded DNA molecule

Throughout the life cycle of a cell its DNA is used to produce protein molecules, which perform a large number of functions within the cell. In this process, illustrated in Figure 2.2, a sequence of DNA is first transcribed to a complementary RNA sequence called messenger RNA (mRNA), which is then translated to produce a protein. RNA molecules are also built out of nucleotide sequences, but with thymine replaced by Uracil (U). The production of a functional product from a strand of DNA (a gene) is called gene expression. The product can be either a protein, or as discussed below, a functional RNA molecule.

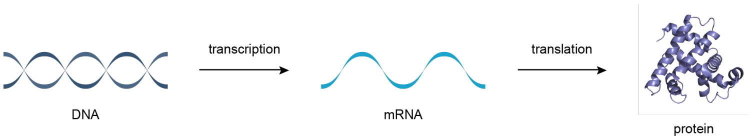


Figure 2.2: DNA is transcribed into messenger RNA, which is translated to protein.

Not all RNA molecules are transcribed into proteins, but rather perform other functions in the cell. Such RNA are collectively known as non-coding RNA (ncRNA), and it has recently become increasingly clear that ncRNA molecules perform a wide range of biological functions [Mattick and Makunin, 2006]. In the next section I will describe microRNAs, which are ncRNA that have been found to play an important role in gene expression regulation.

2.2 MicroRNAs

The RNAs of interest for this report are miRNAs, and in this section I will go through their biogenesis, function, and other miRNA specific details relevant for my work. Except for where otherwise noted, the information given will be based on material from [Soifer et al., 2007], [Bartel, 2009] and [Cenik and Zamore, 2011].

MicroRNAs are a group of short ncRNA consisting of approximately 21-24 nucleotides that play a key role in post-transcriptional regulation of gene expression in both plants and animals. As described in more detail below, miRNAs mediate post-transcriptional gene silencing by binding to target mRNAs. The first description of miRNA-induced gene silencing was given in [Lee et al., 1993], where they suggested a short RNA product from the lin-4 gene regulates lin-14 mRNA translation via an antisense RNA-RNA interaction. Since being recognized as a separate class of conserved ncRNA in 2001 [Lee and Ambros, 2001] a large number of mRNAs have been found to be regulated by miRNA-induced silencing, and abnormal miRNA expression levels have been linked to a variety of diseases, including myocardial infarction and some types of cancer.

There are some differences in both the biogenesis and function of miRNAs between animals and plants, as well as between different species. In the following sections I will focus on the miRNA pathway as found in mammals, as it is the most relevant for my work.

2.2.1 MicroRNA biogenesis

The miRNA biogenesis process is illustrated in Figure 2.3. In the standard pathway, shown in the top left corner of the figure, miRNAs are transcribed from their own genes, first resulting in a longer molecule known as a primary miRNA (pri-miRNA). A pri-miRNA is then cleaved by the microprocessor complex, consisting of Drosha and DGCR8, to produce a precursor-miRNA (pre-miRNA). Precursor-miRNAs consist of about 70 nucleotides folded in a characteristic hairpin or stem-loop structure, and contain the mature miRNA candidates. An example pre-miRNA hairpin is shown in Figure 2.4.

In an alternative pathway, known as the mirtron pathway, miRNAs are transcribed from short introns of host genes. In this process the resulting transcript is not processed by the microprocessor complex; instead the pre-miRNA hairpin is

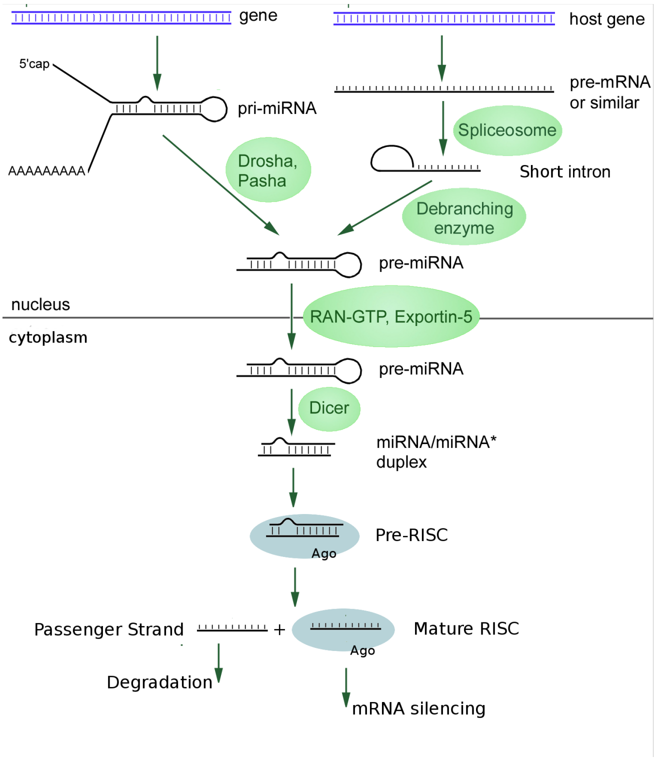


Figure 2.3: The miRNA biogenesis process. Modified from original image courtesy of Daniel Ramsköld, under CC BY-SA 3.0 license.

produced by the mRNA splicing machinery and lariat-debranching enzyme [Okamura et al., 2007]. After the production of a pre-miRNA the two paths merge, and the next step is the transportation of the pre-miRNA from the nucleus to the cytoplasm through a process involving the Exportin-5 protein. In the cytoplasm the pre-miRNA is cleaved by the Dicer enzyme to produce a 21-24 nt miRNA/miRNA* duplex, corresponding to the section marked in blue in Figure 2.4. One of the two duplex strands will form the mature miRNA product and take part in mRNA silencing. This process will be described in the next section.

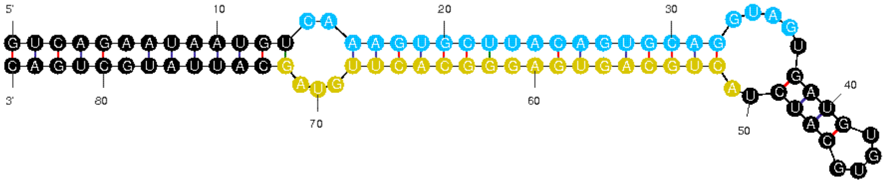


Figure 2.4: The pre-miRNA hairpin structure for mmu-mir-17. The most commonly found 5' (CAAAGUGCUUACAGUGCAGGUAG) and 3' (ACUGCAGUGAGGGCACUUGUAG) mature sequences are marked in blue and yellow, respectively. Figure created using Mfold [Zuker, 2003], with modifications

2.2.2 MicroRNA-mediated gene regulation

MicroRNAs mediate post-transcriptional gene silencing by taking part in a multiprotein complex known as the RNA-induced silencing complex (RISC). The catalytic component of the RISC is a member of the Ago protein family, which first binds the miRNA/miRNA* duplex to form what is called a pre-RISC. There exists many variations of Ago proteins, and most species produce more than one version. Mammals produce four Agos, Ago1-4, all of which can act as the Ago component in a RISC. Of these, Ago2 is known to be functionally distinct, being the only one to cleave target RNAs. This cleaving of targets is called slicer activity. It is not known whether the other three have differing functions.

Of the two miRNA-miRNA* strands, only one will remain to form a mature RISC and take part in the mRNA silencing process. This strand, called the guide strand, has a bias towards the strand with the thermodynamically less stable 5' end. The other strand, called the passenger strand, is evicted from the pre-RISC and ultimately degraded. The eviction and degradation of the passenger strand depends on the type of the Ago protein. Ago2 and other Agos showing slicer activity are thought to release the passenger by cleaving it, while for slicer-independent Agos the process is not fully understood [Kawamata et al., 2009].

The guide strand leads the RISC to target mRNA by binding to a section of the target sequence of at least partial complementary, usually in the 3' untranslated region of the mRNA. Silencing of the mRNA is executed by the Ago protein, and can happen either by site-specific cleavage, mRNA degradation, or translational repression. In mammals, cleavage of the mRNA can only be carried out by Ago2, and additionally requires a high degree of miRNA/mRNA complementarity in the guide seed region (nucleotides 2-8). Ago2 then cleaves the target at the phosphodiester bond connecting the nucleotides laying across from the 10th and 11th nucleotides of the guide strand. The two other methods of silencing can be carried out by RISC containing any of the four mammalian Agos [Gu and Kay, 2010].

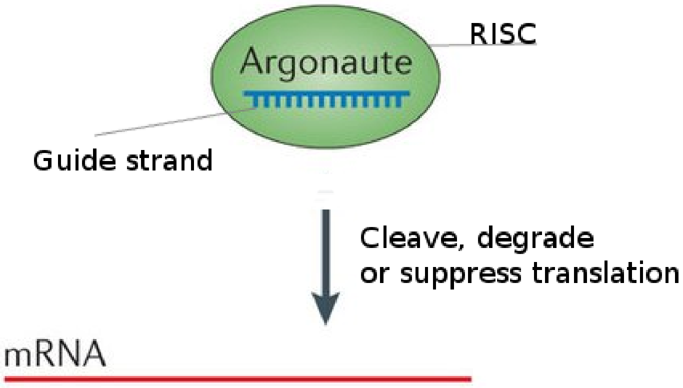


Figure 2.5: A RISC molecule with a loaded guide strand. The miRNA guide leads the RISC molecule and its Ago protein to a target miRNA.

Another class of ncRNA, known as small interfering RNAs (siRNA), are also known to associate with Ago and act as guide strands in RISC. The general process of either miRNAs or siRNAs mediating gene silencing by taking part in RISC is known as RNA interference (RNAi).

2.2.3 IsomiRs

An important detail relevant for this report is that most mature miRNAs show some degree of variation in the expressed sequence [Neilsen et al., 2012]. These different sequences for the same mature miRNA are known as isomiRs. One possible way for isomiRs to arise is through imprecise Drosha and/or Dicer cleavage, resulting in differences in the miRNA/miRNA* duplex product and thus in differences between the mature miRNAs. Another possibility is through post-transcriptional removal or addition of nucleotides at the ends, with the latter of these possibly resulting in a sequence not matching the parent gene. An isomiR is said to be *templated* or *non-templated* depending on whether its sequence is found in the pre-miRNA. Figure 2.6 shows some example isomiRs for mmu-miR-17-5p, for which the reference sequence is also shown in blue in Figure 2.4.

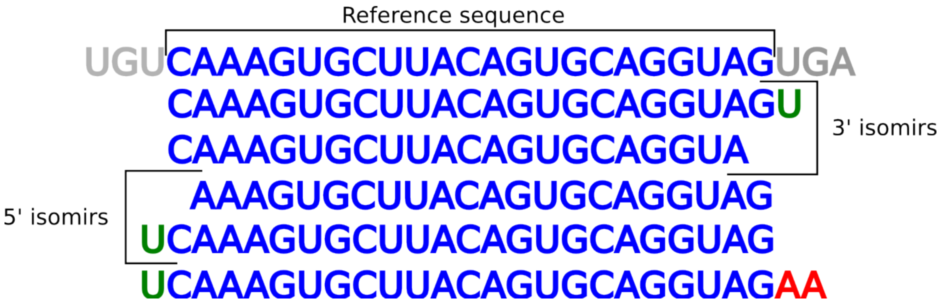


Figure 2.6: The reference sequence for mmu-miR-17-5p with some example isomiRs. The grey colored ends of the reference sequence show the hairpin bases. The 3’ and 5’ isomiRs differ with respect to the reference sequence at their 3’ and 5’ ends, respectively, while the last isomiR differs at both ends. The 3’ end of the last isomiR also does not have a match in the pre-miRNA hairpin sequence, and is said to be a non-templated isomiR.

IsomiRs have been shown to associate with Ago proteins, and are likely to be functional in mRNA silencing [Cloonan et al., 2011]. But it is not clear to what extent they are functionally significant, and whether their biogenesis is a regulated process or the result of “errors” in e.g. Dicer cleavage.

2.3 RNA sequencing

The complete set of RNA transcripts present in a cell or group of cells is known as the transcriptome [Wang et al., 2009]. Transcriptomes change over time and under different physiological conditions, and comparing the measurements obtained from cell samples representing different states of the transcriptome is a common type of experiment, e.g. comparing samples from healthy and sick individuals, or when trying to measure the effect of an external change such as a gene knockout.

RNA-seq is a method for transcriptome profiling based on recent advances in high-throughput DNA sequencing technologies (next-generation sequencing). Several different next-generation sequencing platforms can be used for RNA-seq profiling, but all experiments follow a similar preparation process regardless of the specific sequencing technology used. The main steps are illustrated in Figure 2.7 and briefly described below.

The first step is the isolation of total RNA from the sample cell group. An adapter sequence is then ligated to the 3’ end of the transcripts. To reduce the

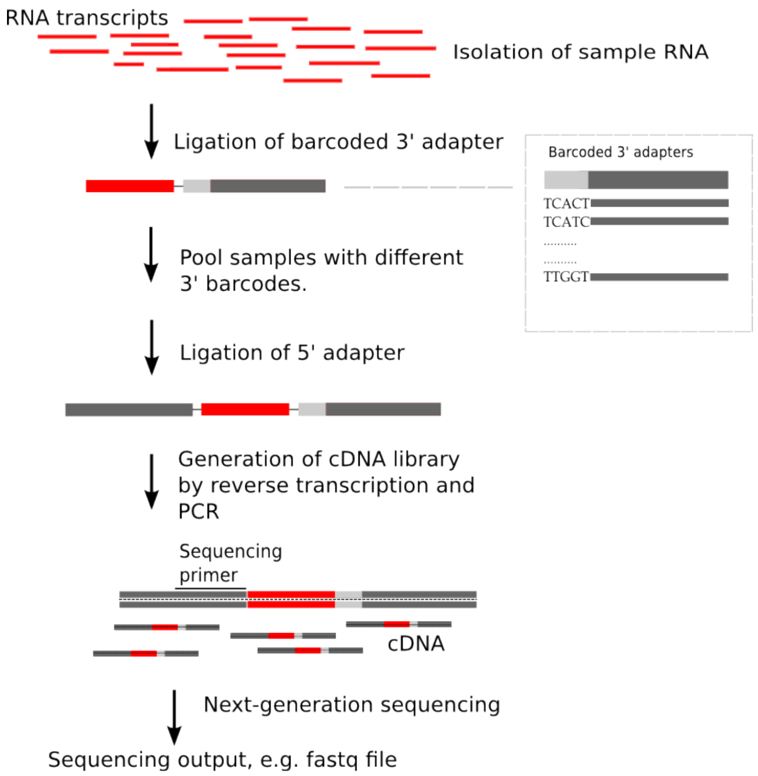
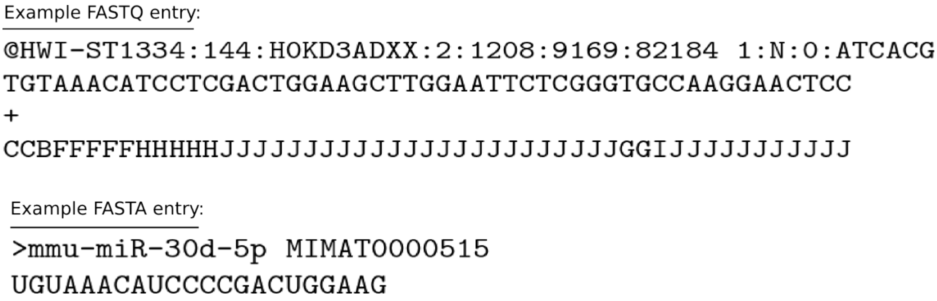

RNA transcripts
Isolation of sample RNA
Ligation of barcoded 3' adapter
Barcoded 3' adapters
TCACT
TCATC
....
....
TTGGT
Pool samples with different 3' barcodes.
Ligation of 5' adapter
Generation of cDNA library by reverse transcription and PCR
Sequencing primer
cDNA
Next-generation sequencing
Sequencing output, e.g. fastq file

Figure 2.7: The main steps of a cDNA library generation process for RNA-seq profiling. Figure adapted and simplified from [Farazi et al., 2012]

cost and overhead of profiling multiple samples, a barcode unique to each sample can be included as part of the 3' adapter [Farazi et al., 2012]. For example, using a barcode of 5 nucleotides, 20 samples can then later be processed together by using the barcodes to map reads back to samples after sequencing. An adapter containing a sequencing primer at its 3' end is then ligated to the 5' end of the transcripts. The RNA molecules are then reverse transcribed into complementary DNA (cDNA) molecules, and amplified using PCR to generate what is known as a cDNA library. The cDNA library represents the relative expression levels of the original RNA transcripts, and is used as input to the sequencing platform. If only short RNA, such as miRNA, are of interest for the experiment, the cDNA library is filtered on sequence length so that only shorter sequences are kept. Finally, the sequencing will result in an output of the reads observed, typically in the form of a FASTQ file, a common text format for storing the output of high-throughput sequencing. A FASTQ file contains one entry per read, as shown in Figure 2.8. The same figure also describes the FASTA format, a simpler text-representation that I will make use of later in the report.



Example FASTQ entry:

```
@HWI-ST1334:144:H0KD3ADXX:2:1208:9169:82184 1:N:0:ATCACG
TGTAAACATCCTCGACTGGAAGCTTGGAATTCTCGGGTGCCAAGGAACTCC
+
CCBFFFFFHHHHHJJJJJJJJJJJJJJJJJJJJJJJJGGIJJJJJJJJJJJ
```

Example FASTA entry:

```
>mmu-miR-30d-5p MIMAT0000515
UGUAAACAUCCCCGACUGGAAG
```

Figure 2.8: The FASTQ and FASTA formats. In FASTQ the first line is a read identifier, the second line contains the read sequence, the third line starts with a '+' character that is optionally followed by the read identifier, and the fourth line contains read quality information (one read quality character per sequence character). See e.g. [Cock et al., 2010] for more information. In the FASTA format, each entry starts with a single-line description, identified by the ">" character, followed by an arbitrary number of lines giving the sequence.

The number of reads for the different sequences give relative expression levels, and not the actual amount of RNA in the original sample. But if wanted, absolute expression can be obtained by adding known amounts of certain calibrator sequences to the sample.

As illustrated in Figure 2.7, the 5' adapter contains the sequencing primer at its 3' end termini, so that the reads obtained run from the position corresponding to the 5' end of the original RNA. The probability of a sequencing error, i.e. that the wrong nucleotide is reported, increases with distance from the 5' end, so that the reads closer to the 5' generally will have higher quality scores. The reads typically consist of approximately 50 nucleotides or more, so that for short RNA a part of the 3' adapter will be included in the reads. This is discussed in the next section, where I describe some common preprocessing steps after obtaining the raw read output.

2.4 Preprocessing of sequencing data

In this section I will look at ways to perform some common first steps in the analysis of RNA-seq data. First, to obtain the original RNA transcripts the 3' adapters must be removed. I will then describe a method for aligning the reads against a reference genome, before looking at identifying differentially expressed sequences.

2.4.1 Adapter trimming

As mentioned above in section 2.3, the length of the reads output from RNA-seq experiments are normally longer (51 nt for the data I have worked with) than short RNA sequences such as miRNA. Reads of such sequences will therefore include part or all of the 3' adapter. Since the adapter is assumed to be attached to the 3' end of the insert RNA there are two possibilities to consider, shown in Figure 2.9. Either the full adapter is included in the read with additional filler nucleotides at the 3', or the read runs into the adapter. For the first case, the full adapter and any nucleotides following it should be removed. For the latter, the part of the read matching the adapter should be removed.

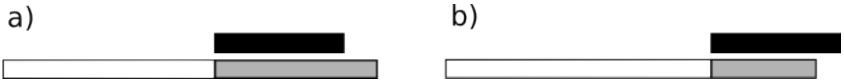


Figure 2.9: The two possible 3' adapter alignments. The black rectangle represents the adapter and its alignment to the read, and the grey rectangle shows the removed suffix.

Due to the possibility of sequencing errors, an exact match against the adapter (or a prefix of it) would possibly discard valid reads. One approach to deal with this is to align each sequence against the adapter using semi-global alignment (see e.g. Durbin [1998] for the dynamic programming algorithm and initial conditions). To force that the adapter should not overlap the 5' end of the read in the alignment, a penalty is added for initial gaps in the read sequence. This is the approach implemented by *cutadapt* [Martin, 2011], the program I have used for my work.

The standard dynamic programming algorithm for global alignment runs in time $O(nm)$ for strings of length n and m. Aligning k reads of length n to an adapter of length m is then $O(knm)$, making this a computationally heavy operation.

2.4.2 Read alignment

To filter out and identify previously annotated sequences the reads can be aligned against a reference genome or set of sequences. For example, to identify miRNAs, we can match the reads against known miRNA precursor sequences. In this section I will outline the ideas behind *Bowtie*, a fast and memory efficient short read aligner [Langmead et al., 2009]. *Bowtie* is based on building an index from the Burrows-Wheeler transform (BWT) [Burrows et al., 1994] of the reference genome/sequence(s). Every read is then aligned to the reference sequence by searching the index for a possible match. The original algorithm for using the BWT for exact search was first described in [Ferragina and Manzini, 2000]. *Bowtie* augments this algorithm with a backtracking procedure to handle indels and mismatches.

The Burrows-Wheeler transform

The Burrows Wheeler transform is a reversible permutation of the characters of a string. Originally developed for lossless compression, as it will tend to group equal characters together, it can also be used to create an efficient index for searching in the input string. For a string S over the alphabet A, BWT(S) is defined as follows: First add a character \$ that is not part of A to the end of S, and define \$ to be lexicographically smaller than any element of A. Then form all cyclical shifts of the resulting string and sort them lexicographically to form a matrix M. BWT(S) is then the last column of M, as illustrated in Figure 2.10. In the following I will denote the first and last column of M as F and L, respectively.

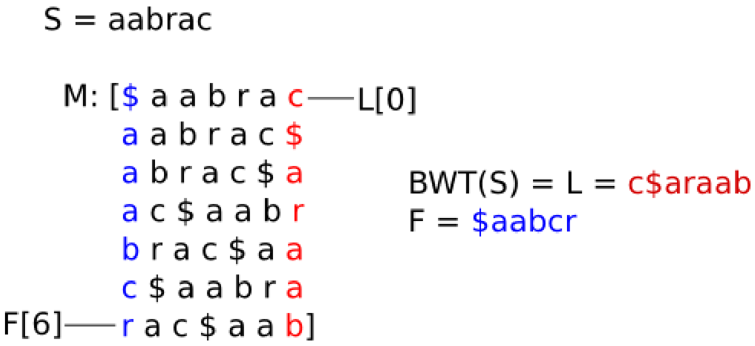


Figure 2.10: Burrows Wheeler Transform

Note that because the rows correspond to cyclic shifts, all columns of M

correspond to a permutation of S. From this it also follows that for every row the last character of the row will precede the first character of the same row in the original input. And because the rows of M are sorted lexicographically, F can be reconstructed from L simply by sorting L (or any column).

A key property of the BWT, known as Last-First (LF) mapping, is that for any character X, the i'th occurrence of X in L corresponds to the same character in the original input as the i'th occurrence of X in F. *Proof:* Consider two equal characters L(i) and L(j) in L, and the characters that follow them in their respective rows, F(i) and F(j). Let k and l denote the two rows in M obtained by shifting rows i and j one position to the right, respectively, so that F(k) = L(i) and F(l)=L(j). We see that the second character in rows k and l will be F(i) and F(j), respectively. And because M is sorted lexicographically, it follows that the relative order between rows k and l will be the same as for rows i and j.

An example LF-mapping, where each character in L is mapped to the corresponding character in F, is shown in Figure 2.11.

Recovering the original input

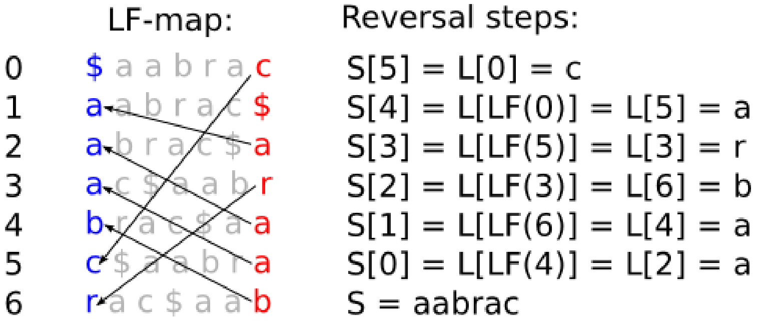
Recovering the input S can be done by following the links of the LF-mapping. To construct the LF-map it is useful to first define two precomputed tables: Let $C(c)$ be the number of characters in S from the set $\{\$, 1, 2..., c-1\}$, i.e. the number of characters in S lexicographically smaller than c. It follows that if c appears at least once in S its first occurrence in F will be at position $C(c)$. Also, let $r(c, i)$ be the number of occurrences of c among the first i characters of L.

The LF-mapping can then be computed as $LF(i) = C(L[i]) + r(L[i], i)$. The position of the i'th character in L is mapped to the position of the corresponding character in F. Algorithm 2.1 shows how S now can be recovered from L, starting from the known last character and following the LF-map. This is also illustrated with an example in Figure 2.11

Algorithm 2.1 Recover original input S from L =BWT(S), adapted from [Ferragina and Manzini, 2000]

Precondition: Assume LF(i) precomputed.

```
function ReverseBWT(L)
    rowIndex ← 0
    S[0] ← L[0]
    for $i \leftarrow len(L) - 1$ to 1 do
        rowIndex ← LF[rowIndex]
        S[i] ← L[rowIndex]
    return S
```



Reversal steps:
S[5] = L[0] = c
S[4] = L[LF(0)] = L[5] = a
S[3] = L[LF(5)] = L[3] = r
S[2] = L[LF(3)] = L[6] = b
S[1] = L[LF(6)] = L[4] = a
S[0] = L[LF(4)] = L[2] = a
S = aabrac

Figure 2.11: Example LF-map and recovery of the input string S from L =BWT(S). S =aabrac, L =c$araab. An arrow from row i to j indicates that $LF(i) = j$. The reversal steps show how Algorithm 2.1 uses the LF-map to recover S.

BWT exact search

Searching can be done by building an index consisting of L, $C(c)$ and $r(c, i)$, known as a full-text minute-space (FM) index. The procedure is shown in Algorithm 2.2. Searching proceeds one character at a time, starting from the end of the search string q. An interval of rows with a prefix matching the suffix of q currently being considered is kept at each stage. If q is found, the interval of rows starting with q is returned. If a suffix of q is not found the interval becomes empty. A simple example search is also shown in Figure 2.12.

It is clear from the discussion above that the search runs in $O(|q|)$ time for a query q, and that storing L, $C(c)$ and $r(c, i)$ takes $O(n)$ space. In [Ferragina and Manzini, 2000] a procedure is given for computing $r(c, i)$ in constant time (assuming the machine word size is at least $\log_2(|S|)$ bits) for a given c, so that this precomputation step is also $O(q)$. Building the index is thus dominated by the BWT and computation of C, both of which are $O(|S|)$.

Allowing for mismatches

When mapping reads to a reference genome one typically wants to allow for some degree of mismatches. To achieve this Bowtie uses a modified variant of the exact search algorithm. When the interval becomes empty, i.e. the query string is not found, the search may backtrack to an already matched position and substitute in a different base, thus introducing a mismatch. The search then proceeds as normal from the substituted position. The position substituted is chosen randomly from the positions having the lowest read quality.

Algorithm 2.2 Exact search in FM-index, adapted from [Ferragina and Manzini, 2000]. An interval (start, end) of rows matching a suffix of the query q is kept as the suffix is increased one character at a time.

Precondition: Assume C(c), r(c, i) precomputed.

```
function FM-ExactSearch(q)
    n ← len(q)
    c ← q[n-1]
    (start, end) ← (C(c), C(c+1))
    i ← n-1
    while start < end and i ≥ 0 do
        c ← q[i]
        start ← C(c) + r(c, start)
        end ← C(c) + r(c, end)
        i ← i - 1
    return (start, end)
```

To avoid excessive backtracking an upper limit on the number of backtracks is used. This means it is possible to miss the best alignment, giving a tradeoff in runtime vs increasing probability of missing the best alignment. For the case when only one mismatch is allowed, a special “double index” technique is used, where an additional BWT of the reverse genome (i.e. the character string as read 3’ to 5’) is used. The starting point of the technique is the obvious fact that the mismatch must be in either the right or left half of the query, and that most excessive backtracking happens for bases close the 3’ end. A search is first done on the forward index, but with the constraint that a substitution cannot be done in the right half of the query. A second search is then done using the reverse index on the reversed query, with the same constraint that no mismatches are allowed in the right half. Together these two searches cover the two possible cases of a mismatch in the left and right half, and reduces excessive backtracking close to the 3’ end.

2.5 Identifying differentially expressed RNA

In this section I will look at identifying differentially expressed sequences from an RNA-seq experiment. I will focus on a simple design setup where we are interested in measuring differential expression between two different sample groups, S_a and S_b, although the described procedure can also be applied to more complex designs. If the samples have been sequenced to give m RNA-seq profiles (some from S_a and some from S_b) and we are considering the expression of n sequences, the

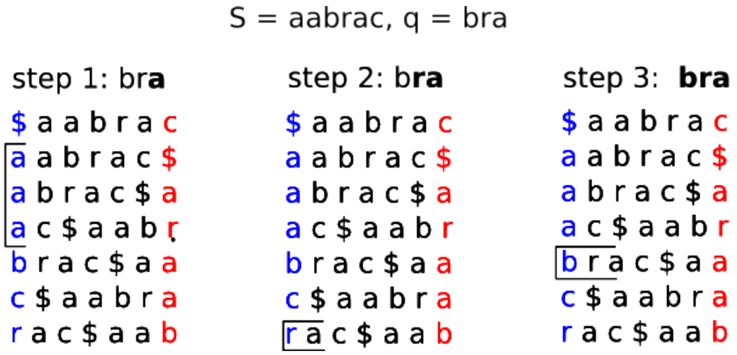


Figure 2.12: An example exact search. The borders show the interval of rows matching the prefix for each step.

starting point for the analysis can be represented by a $n * m$ matrix Y, where the i'th row gives the raw expression counts for sequence i over the m profiles. I will denote a row of Y as $\boldsymbol{y}_i$ and a single read by $y_{i,j}$, $i \in [n]$, $j \in [m]$.

The methods I will describe here are implemented in the limma R package [Smyth et al., 2004], which I have used for my work, and are detailed in [Law et al., 2013] and Smyth [2005]. The following material is based on the presentation given in these papers. The key idea is to fit a weighted linear model to the counts for each sequence (i.e. to each row of Y), and to use an empirical Bayes approach where variance information from all sequences is incorporated as a prior when testing each sequence individually.

I will first discuss the idea of fitting a linear model to the data, then look at normalization and handling heteroscedasticity, before giving a test for differential expression.

2.5.1 Sequence-wise linear modeling

A linear model for each sequence can be represented as

$$E(\boldsymbol{y}_i) = X\boldsymbol{\alpha}_i \tag{2.1}$$

Here X is the design matrix of the experiment, and $\boldsymbol{\alpha}_i$ are the coefficients to be fitted. For the setup considered here, a natural choice for X is (assuming two wild-type profiles and two knockout profiles):

$$X = \begin{array}{c} \begin{array}{cc} WT & KO \end{array} \\ \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \end{array}$$

so that the coefficients $\alpha_{i,1}$ $\alpha_{i,2}$ of $\boldsymbol{\alpha}_i$ describe wild-type and knockout contributions, respectively. By adjusting X more complex designs can be tested using the same methods described here.

Differences (or contrasts) in certain coefficients of α_i are then tested to identify differential expression. This can be written as

$$B_i = C\boldsymbol{\alpha}_i \tag{2.2}$$

where C is the contrast matrix and B_i are the contrasts of interest. Again considering the WT-KO design from above, we are interested in testing for a difference in the two coefficients, so the contrast matrix would simply be $C = \begin{pmatrix} -1 & 1 \end{pmatrix}$, giving a single contrast value to be tested in B_i.

2.5.2 Normalization and read weighting

Because there can be significant differences in the sequencing depth between the runs, i.e. some columns of Y have overall higher counts than others, it is necessary to normalize the data. The standard way to do this is to express the counts in counts-per-million (CPM), so that the counts for each profile are normalized by the total read count of the profile. Counts can then more naturally be compared between profiles.

One problem with now using these normalized CPM for fitting linear models is the heteroscedastic nature of such count data. Intuitively, the higher the expected value for a read $Y_{i,j}$ the higher the variance. If assuming sequencing of a sample follows a Poisson process, as argued in [McCarthy et al., 2012], the mean-variance relationship would be linear. It can then be shown that by taking the logarithm of the counts the mean-variance becomes approximately constant, except for a higher variance for sequences with few reads (see Figure 2.13). To avoid taking the log of zero, the log-CPM for a read is computed as

$$\text{log-CPM}(y_{i,j}) = \gamma_{i,j} = \log_2\left(\frac{y_{i,j} + 0.5}{R_i + 1.0} * 10^6\right) \tag{2.3}$$

Where R_i is is the sum of the reads for profile i, i.e. the sum of column i of Y.

The idea is to use the mean-variance relationship to assign a weight to each individual read $y_{i,j}$, and use the weights to fit a weighted linear model to each $\boldsymbol{y}_i$ to test for differential expression, as explained below. Because the variance depends on the raw read count, the weights should be computed for this count and not a normalized count. To derive the weights, a least square linear model is first fitted for the log-CPM values $\boldsymbol{\gamma_i}$ for each sequence, giving fitted values $\hat{\boldsymbol{u}}_i = X\hat{\boldsymbol{\alpha}}_i$ and corresponding residual standard deviations s_i. The average log-cpm $\bar{\gamma}_i$ is computed and converted back to an average log-count, which is plotted against s_i to obtain a mean-variance plot. A curve w is fitted to this data to obtain an estimated mean-variance relationship. The predicted standard deviation for a read $y_{i,j}$ is calculated by first converting the fitted log-cpm values $\hat{u}_{i,j}$ to fitted log-counts (by reversing equation 2.3), and then looking up the value of w for this log-count. The weight given to $y_{i,j}$ is then calculated as a function of the estimated standard deviation. See [Law et al., 2013] for more details. Figure 2.13 shows an example mean-variance plot for log-counts, computed using one of the datasets I have worked with.

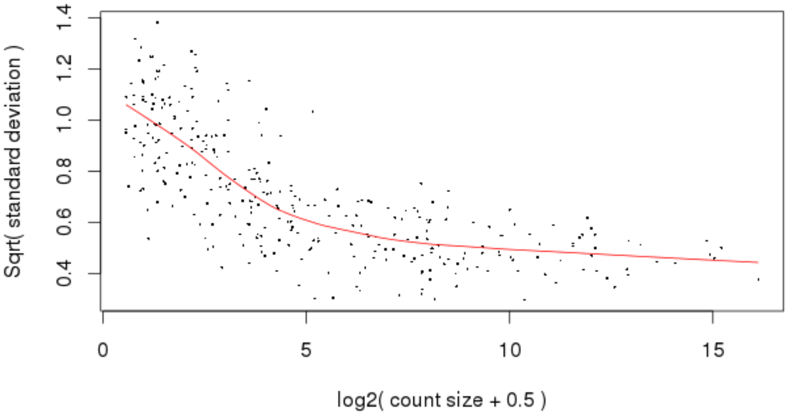


Figure 2.13: An example mean-variance plot. The variance can be seen to approach a constant value for $\log_2$ counts $\gtrsim 6$.

2.5.3 Testing for differential expression

Using the individual read weights and log-CPM values, a linear model is fitted as described in 2.5.1, using a e.g. weighted least squares, to obtain estimates $\hat{\boldsymbol{\alpha}}$, s_i^2

and covariance matrices V_i; $var(\hat{\alpha}) = V_i s_i^2$. If we assume the estimated contrast values $\hat{B}_{i,j}$ are approximately normally distributed as follows:

$$\hat{B}_{i,j} | B_{i,j}, \sigma_i^2 \sim N(B_{i,j}, v_{i,j}\sigma_i^2)$$

then we can construct a t-statistic with $d_g = m - 1$ degrees of freedom (the number of terms in s_i minus one) for testing whether $B_{i,j}$ is different from 0:

$$t_{i,j} = \frac{\hat{B}_{i,j}}{s_i \sqrt{v_{i,j}}} \tag{2.4}$$

This statistic is improved upon by utilizing information from all sequences to put a prior on σ_i^2. The conjugate prior for the variance of a normal distribution is a scaled inverse chi-squared distribution, so that given a prior estimate s_0^2 of d_0 degrees of freedom: $\sigma_i \sim$ Scale-Inv-$\chi^2(d_0, s_0^2)$. The posterior estimate $\tilde{s}_i^2$ is then (see e.g. [Wikipedia, 2014] for a derivation):

$$\tilde{s}_i^2 = \frac{d_0 s_0^2 + d_i s_i^2}{d_0 + d_g}$$

The *moderated* t-statistic, of $d_0 + d_g$ degrees of freedom, is formed by using $\tilde{s}_i$ for s_i:

$$t_{i,j} = \frac{\hat{B}_{i,j}}{\tilde{s}_i \sqrt{v_{i,j}}} \tag{2.5}$$

See [Smyth et al., 2004] for more details, including derivation of the hyperparameters s_0 and d_0, as well as a comparison between using the moderated t-statistic from Equation 2.5, and other simpler approaches including the t-statistic from Equation 2.4.

2.6 Machine learning and classification

In this section I describe the theory of the core methods employed in section 5.1, where I try to perform classification of up- and down regulated isomiRs. These methods are part of the broader field of machine learning, which in general can be defined as the study of computer systems that learn from data; the performance of the system is improved by learning from observed data, instead of being explicitly programmed [Mitchell, 1997]. I first give an introduction to classification with support-vector machines, before discussing how the performance of a classifer can be estimated.

2.6.1 Classification with Support Vector Machines

The support vector machine (SVM) introduction given here is a modified and somewhat shortened version of the one I gave in [Mossin, 2013], and is based on material from [Ben-Hur and Weston, 2010] and [Barber, 2012].

In classification, the goal is to correctly predict the label of new observations, given that a training set of labeled data has been observed. SVMs perform classification by constructing a separating hyperplane in the feature space of the observations. Given such a hyperplane, the label of a new observation is simply determined by which side of the plane it belongs.

In the following I will denote the p-dimensional feature space of all possible observations by $\mathbb{X} \in \mathbb{R}^p$, a single observation by $\mathbf{x}_i$, the components of an observation by $x_i \in \mathbb{R}$, and the label of observation $\mathbf{x}_i$ by $y_i \in \{-1, 1\}$.

Linearly separable data

The notion of a separating hyperplane and its usage in classification is best introduced in the case of data that is linearly separable in $\mathbb{X}$. This is illustrated in Figure 2.14 for a 2-dimensional example.

If we represent the hyperplane by its normal vector $\mathbf{w}$ and an offset constant b, as follows:

$$\{ \ \mathbf{a} : f(\mathbf{a}) = \mathbf{a} \cdot \mathbf{w} + b = 0 \ \}$$

we can classify an observation $\mathbf{x}$ by taking the sign of $f(\mathbf{x})$ as the label.

Now let $\mathbf{x}_-$ and $\mathbf{x}_+$ denote the vectors from the negative and positive groups, respectively, that are closest to the hyperplane, and let the hyperplane be defined such that the distance is the same for $\mathbf{x}_-$ and $\mathbf{x}_+$: $f(\mathbf{x}_+) = -f(\mathbf{x}_-) = k$. The constant k can be set arbitrarily by scaling $\mathbf{w}$ and b, and it is convenient to set it to $k = 1$. Then we can define the margin of the hyperplane, i.e. the distance from $\mathbf{x}_+$ to $\mathbf{x}_-$, as follows:

$$m_f = \frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot (\mathbf{x}_+ - \mathbf{x}_-) \tag{2.6}$$

$$m_f = \frac{2}{\|\mathbf{w}\|} \tag{2.7}$$

The main idea of SVMs is to use the maximum margin hyperplane, and from Equation 2.7 we that the maximum margin is found by minimizing $\|\mathbf{w}\| = \sqrt{\mathbf{w}^\mathbf{T}\mathbf{w}}$. This can be formulated and solved as a quadratic programming (QP) problem:

$$\begin{aligned} \underset{(\mathbf{w},b)}{\text{minimize}} \quad & \mathbf{w}^\mathbf{T}\mathbf{w} \\ \text{subject to} \quad & y_i(\mathbf{x_i} \cdot \mathbf{w} + b) \geq 1, \ i = 1, \ldots, m. \end{aligned} \tag{2.8}$$

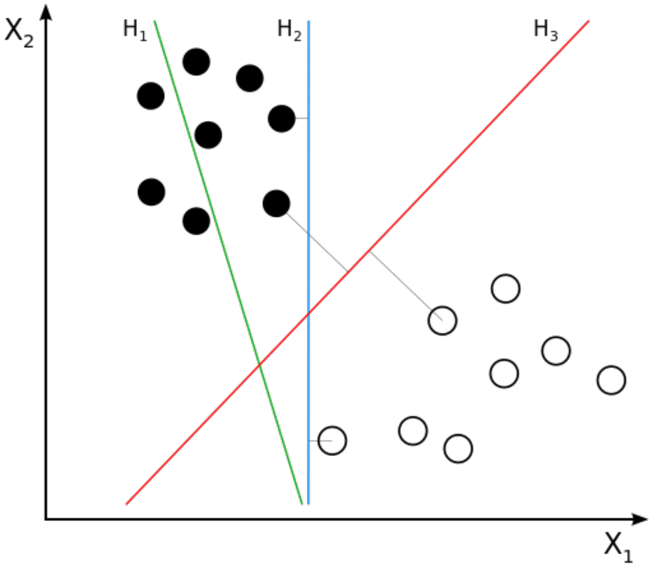


Figure 2.14: An example of data that is linearly separable in 2D. The planes H3 and H2 separates the data, while H1 does not. For H2 and H3 the grey lines show the closest data points from each group, and we see that H3 has a larger margin. Image courtesy of Zack Weinber, under CC BY-SA 3.0 license.

Soft-margin SVMs and kernel functions

If the training data is not linearly separable then the QP problem 2.8 has no solution. To account for this we need to relax the constraints to allow for points to be on the wrong side of or inside the margin. As illustrated in Figure 2.16, this may also help in constructing a better classifier for the case of linearly separable training data. Equation 2.8 is modified by adding “slack“ variables to the constraints as follows:

$$\begin{aligned} &\underset{(\mathbf{w},b)}{\text{minimize}} && \mathbf{w}^{\mathbf{T}}\mathbf{w} + C\sum_i \xi_i \\ &\text{subject to} && y_i(\mathbf{x_i}\cdot\mathbf{w}+b) \geq 1-\xi_i,\ i=1,\ldots,m. \end{aligned} \tag{2.9}$$

Data points with non-zero ξ are those that are inside or on the wrong side of the margin, and the parameter C determines how hard to penalize such points.

To further deal with data that is hard to separate in a good way with a linear plane, SVMs can be made to work with what is known as kernel functions. To

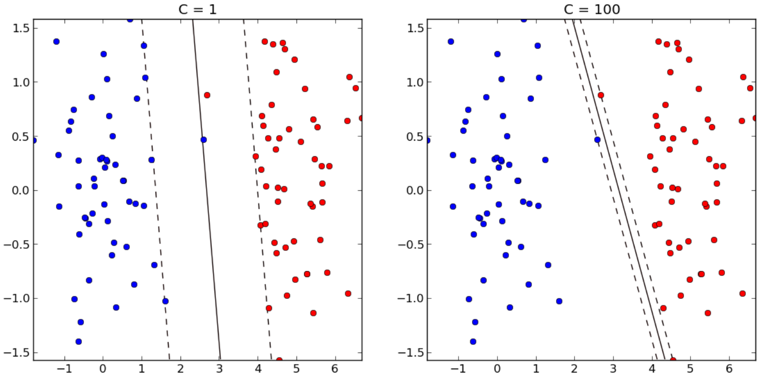


Figure 2.15: Soft margin SVM example. A higher C (on the right,) gives a higher penalty to points with non-zero ξ. Even though the data is linearly separable, the left hyperplane (C=1), which has two datapoints inside the margin area, seems like it would generalize better to new data.

obtain a non-linear decision plane we can map the observations $\mathbf{x}$ in a non-linear way to a new feature space with a mapping $\phi(\mathbf{x}) : \mathbb{X} \rightarrow \mathbb{F}$. $\mathbb{F}$ will typically be of a much higher dimension (e.g. exponential in the size of the original feature space), making computing the mappings unpractical. I will skip the details (see e.g. [Ben-Hur and Weston, 2010]), but the main point is that it is possible to define useful mappings ϕ for which dot products can be efficiently computed without explicitly computing the mappings, and that the SVM equations can be stated in terms of dot products only. In fact, $\mathbb{F}$ can be an infinite-dimensional space, as long as it has well defined finite dot products. The kernel I have used for my work, called the radial basis function (RBF), defines a mapping to such an infinite space.

2.6.2 Classifier generalization and cross-validation

The goal of a classifier is of course to perform well on new observations. With this goal in mind, there are two central and related questions: How well can the classifier be expected to perform on new data, and what setting of the parameter values can be expected to give the best performance?

An obvious way to get an estimate of expected performance is to reserve part of the available training data for evaluation (e.g. 25%), so that the classifier is

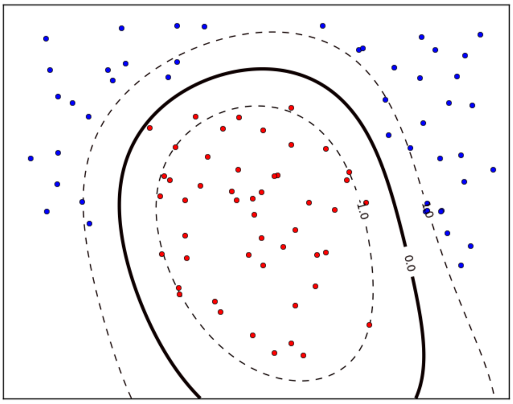


Figure 2.16: An example of non-linearly separable data, and a hyperplane and margin obtained using an RBF kernel.

trained on one part of the data and then tested on the remaining. If only a limited amount of training data is available though, such a complete hold-out of part of the data may not be desirable. Cross-validation (CV) is a performance estimation technique that deals with this problem. In k-fold CV the training data is first split into k folds of equal size, followed by k-rounds of training and evaluation. In the i'th round, the i'th fold is reserved for testing and the other folds are used for training. The basic principle is illustrated in Figure 2.17. A score is computed in each round, and the final score is computed from the individual scores (e.g. the average).

One way to select the parameters of a model is then to perform CV for each parameter setting and use the setting that performs the best. Though as an estimation of a classifiers generalization to new data, the best score obtained from such a grid search over different parameter settings is prone to overfitting, as we are fitting the parameters to all available training data.

A technique for obtaining an unbiased performance estimate is known as nested CV, where two nested rounds of CV is performed. The training data is split into k-folds for the outer CV loop, as explained earlier and shown in Figure 2.17. The training fold is then used to select the model parameters by performing

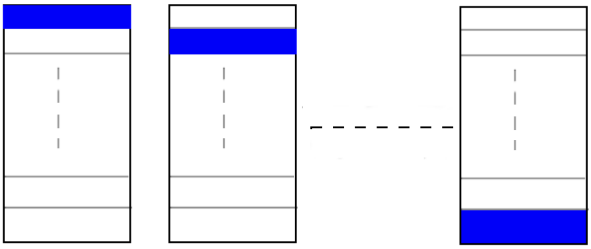


Figure 2.17: Illustration of k-fold cross validation. In each round, the classifier is tested on the fold marked in blue, and trained on the others.

CV for each setting of the parameters. Having obtained the parameters, the actual performance is estimated on the testing fold, which took no part in the parameter selection search.

CHAPTER 3

Related Work

In this chapter I will briefly go through some previous work and results relevant for my project.

A review of the evidence suggesting isomiRs have distinct functional roles is given in [Neilsen et al., 2012]. The main conclusion is that, while many isomiRs seem to be functionally redundant, there is growing evidence for at least some isomiRs being biologically significant, and the authors argue this should be considered when analysing miRNA data. In [Cloonan et al., 2011] isomiR variants are found to associate with RISC and take part in mRNA silencing. They also report that isomiR expression levels are highly correlated, and use this and other findings to conclude that isomiRs generally target common biological pathways.

A related problem to finding isomiRs showing different knockout expression changes has previously been studied for the case of differential expression of exon transcripts. It is known that a single gene can give rise to multiple transcripts, and that such variations in some cases result in proteins with functional differences [Nilsen and Graveley, 2010]. This observation has helped explain how highly complex organisms like mammals contain only a few times the number of genes found in e.g. budding yeast. [Anders et al., 2012] presents a statistical framework for determining whether the relative expression between the different exon transcripts of a gene is preserved between samples of different experimental conditions, i.e. whether the transcript variants show differing fold changes. The starting point for such an analysis is similar to the problem I study for isomiRs in section 5.2: a table containing the observed count for each exon over the different samples. The presented method is based on modeling the counts for each gene using generalized linear models (GLMs), with the random variable representing the observed count of a specific exon in a specific sample modeled by a negative binomial distribution. Two types of models are fitted for each gene: one model treating all exons of a gene together, representing the null hypothesis that the relative fractions of exon expressions should equal between the studied conditions, and one model which includes a coefficient for representing an exon specific expression change. The former model is fitted once per gene, while the latter is fitted for each exon. An analysis of deviance test is then performed to test the significance of the exon-specific term.

In the introduction chapter I briefly described my previous work in [Mossin, 2013], where I tried to classify miRNAs based on expected expression changes in Ago2-knockout cells. This study was the starting point for the work presented in section 5.1. Several other previous studies have also applied support vector machines for performing classification of miRNA data, e.g. in [Xue et al., 2005] where SVMs are used to separate real- and pseudo pre-miRNAs. Using sequence features for classification has also been studied for determining miRNA and siRNA efficacy and target prediction, see e.g. [Sætrom and Snøve Jr, 2004] and [Saito and Sætrom, 2010].

A review of mechanisms affecting the degradation of mature miRNAs is given in [Kai and Pasquinelli, 2010], including a discussion of 3' A- and U-tailing. An extensive study of 3' additions is presented in [Wyman et al., 2011], and they report that 3' additions are commonly found in species ranging from humans to C.elegans, and that the additions most often consist of either A- or U-tails. They speculate A/U-tailing either serve as a tag for degradation, or represent a consequence of miRNAs already being targeted for degradation, but conclude future studies are required for determining the functional scale of such 3' additions.

CHAPTER 4

Materials and Methods

4.1 Project datasets

I have used two different datasets for my work, both originating from small RNA sequencing projects conducted at NTNU's Department of Cancer Research and Molecular Medicine.

The first dataset comes from an experiment conducted by Marie Lundbæk, Robin Mjelle and Pål Sætrom, measuring miRNA expression levels in mouse Ago2-knockout cells and corresponding wild-type cells. The data is made up of samples from two separate sets of cell lines, with three wildtype and three knockout samples from each. The cell lines are described in [O'Carroll et al., 2007] and [Liu et al., 2004], and when working with data from a specific cell type I will refer to these as the DOC and GH cells, respectively. I will refer to the complete dataset (the DOC and GH data together) as the GCF data.

The DOC wildtype and knockout samples are derived from cells that should be otherwise equal. For the GH data the wildtype and knockout samples were obtained from mice of different gender. Although unfortunate, the cells are still similar enough for studying the effect of Ago2-knockout.

For work not related to differential expression in Ago2-knockout cells, I have also used data from mouse CH12 cell lines [Nakamura et al., 1996], obtained from a project by Hans Krokan and Robin Mjelle. I will refer to these samples as the CH12 data. The data contains a total of 20 samples, and compared to the GCF data the CH12 samples also generally contain more reads. Except for this detail, the specifics of the CH12 data will not be important for the work in this report; it is included as an additional dataset to get stronger evidence for results first discovered in the GCF data.

4.2 Reference sequence data

I have used miRBase (version 20) [Kozomara and Griffiths-Jones, 2011] to obtain both reference mature and precursor sequences. MirBase is the primary public repository for all published miRNA sequence data and corresponding annotation information. I used the database dump files provided by miRBase to set up a local MySQL version of the database. As of version 20, miRBase contains a total of 1186 mouse pre-miRNA sequences, of which 370 are labeled as “high-confidence” miRNAs, i.e. hairpins that are believed to be real pre-miRNA with a high degree of confidence. To reduce the possibility of noise due to pseudo-hairpins I have worked only with the high-confidence miRNAs.

Predicted secondary structures are not included in the database files, but can be separately downloaded from miRBase. The structure information is only given in a format more suitable for visual inspection though, making it necessary to parse the data to a more machine friendly format. I therefore converted all structure information to a dot-bracket representation, shown in the following example:

```
Example miRBase secondary structure representation:
c    u  u      uu   g   u            uagaguuac   aa
 ugca gu cccagg  gag uag agguuguauaguu         auc  g
 |||| || ||||||  ||| ||| |||||||||||||         |||
 acgu ca ggguuc  uuc auc uccgacaugucaa         uag  g
c    u  -      cu   g   c            ---------   ag

Corresponding parsed representation:

.((((.((.((((((..(((.(((.(((((((((((((.........(((......)))))))))))))))).))).)))..)))))))).)))).
```

4.3 Preprocessing pipeline

In this section I will describe the initial preprocessing steps I performed when starting the analysis of a new dataset. I have followed the main steps in the analysis pipeline described in [Farazi et al., 2012]. The analysis starting point is a set of FASTQ files containing the raw RNA-seq read data, each file representing the reads from one sample. Each sample file is processed independently to produce

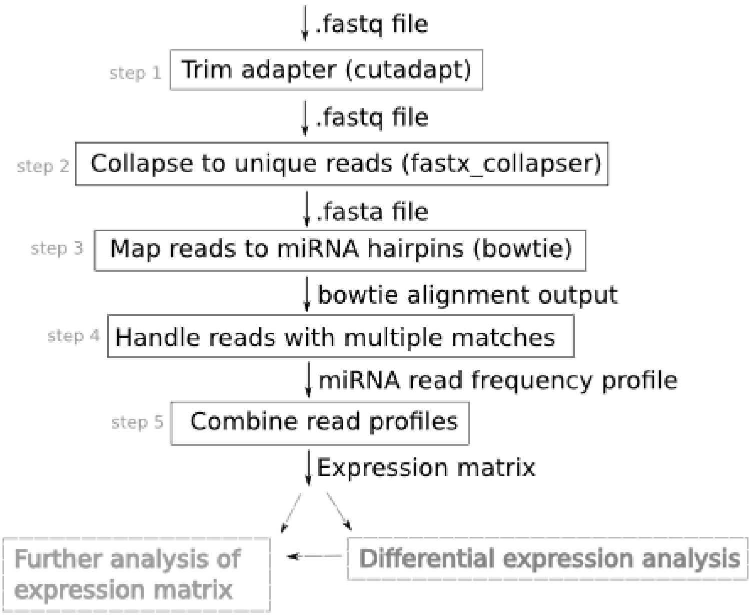

.fastq file
step 1
Trim adapter (cutadapt)
.fastq file
step 2
Collapse to unique reads (fastx_collapser)
.fasta file
step 3
Map reads to miRNA hairpins (bowtie)
bowtie alignment output
step 4
Handle reads with multiple matches
miRNA read frequency profile
step 5
Combine read profiles
Expression matrix
Further analysis of expression matrix
Differential expression analysis

Figure 4.1: High level overview of the preprocessing pipeline. The arrows indicate input/output data, and the rectangles represent the work performed. Specific tools used are indicated in parentheses. A FASTQ file is processed to produce miRNA read frequencies for the sample. After all samples have been processed the profiles can be combined to produce an expression matrix, which becomes the starting point for further analysis.

a profile of miRNA read frequencies. An overview of the pipeline steps is shown in Figure 4.1.

The first step of the pipeline is adapter removal. For the datasets I have worked with the reads were already grouped by barcode, so that partitioning of reads during adapter removal was not necessary. I used *cutadapt* to remove the adapters, giving additional minimum and maximum read length parameters. When working with mature miRNAs the minimum and maximum lengths were set to 16 and 26 nucleotides, respectively. The next step is to obtain read counts per individual sequence. For this I have used *fastx_collapser* [FASTX-Toolkit, 2014], which counts the occurrences of each sequence to produce a .fasta file with one entry per sequence. Collapsing the reads naturally removes read quality information, as there is no good way to combine quality scores for multiple reads.

This means quality scores are not available when aligning the reads, but as described below I have required a perfect match when aligning, so that it would not have made a difference.

The relevant reads for this project are those that correspond to mature miRNA sequences (including isomiR variants). The identification of these reads is represented by step 3 in Figure 4.1. As a first step in extracting mature miRNA-reads I used *bowtie* to align the reads against mouse pre-miRNA sequences. The hairpin sequences were downloaded as a FASTA file from miRBase (version 20), and used to build a *bowtie* index. Using the miRBase reference mature miRNAs, I then kept the reads where the alignment position was no more than 3 positions up- or downstream from a reference mature sequence. Figure 4.2 gives some examples of included and excluded reads.

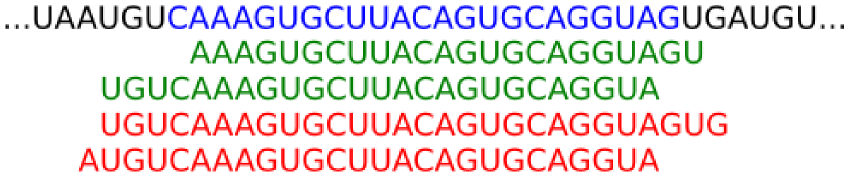

...UAAUGUCAAAGUGCUUACAGUGCAGGUAGUGAUGU...
AAAGUGCUUACAGUGCAGGUAGU
UGUCAAAGUGCUUACAGUGCAGGUA
UGUCAAAGUGCUUACAGUGCAGGUAGUG
AUGUCAAAGUGCUUACAGUGCAGGUA

Figure 4.2: The first line gives a substring of the mmu-mir-17 hairpin sequence, with mmu-miR-17-5p shown in blue. Then follow four example reads. The two first reads will be kept as valid isomiRs, while the latter two will be discarded. The third read, of length 28, will be removed during adapter trimming for being too long. The fourth read aligns against the hairpin at an offset of 4 relative to the canonical miRNA, and so will be removed when filtering for valid isomiRs.

The fourth pipeline step deals with the problem that some sequences will map to multiple hairpins. This is especially true if mismatches are allowed in the alignment, and I therefore decided to require a perfect alignment when running *bowtie*. As the probability for RNA-seq read errors are higher the closer a base is to the 3' end of the read, small RNAs are less likely to have read errors than e.g. mRNAs. But even requiring a perfect alignment, some sequences still have multiple matches. By default *bowtie* reports only one alignment per sequence, choosing randomly between the alignments to avoid a bias in the reporting [Bowtie, 2014]. This does not work well for my purposes, as a read could be attributed to different pre-miRNA for different samples, making e.g. differential expression analysis difficult. I therefore instructed *bowtie* to report all found alignments, and chose one in a deterministic way: I first compared the alignment offsets with the positions of the mature miRNA for the respective hairpins, choosing the hairpin where the alignment was closest to a reference mature miRNA. For further ties I chose the alphabetically smallest pre-miRNA.

An alternative approach could be to simply keep all found alignments and keep track of which isomiRs map to multiple hairpins, but I decided against this approach as it would make the later analysis more complicated.

An important practical detail is that the steps 1-4 described above can be run in parallel per sample, i.e. the processing of several .fastq files can be done concurrently. As I only aligned the reads against mouse pre-miRNAs and not a genome, adapter removal was by far the most time consuming operation. Adapter removal can itself be parallelized for reads within one sample, since each read can be handled independently, but this is not currently supported by *cutadapt*. Processing multiple samples concurrently thus greatly reduced the running time.

The read profile output from step 4 can be viewed as a vector $\mathbf{v_j}$, where the i'th element $v_{j,i}$ is the read count observed in sample j for isomiR i. In the last pipeline step, the read vectors $\mathbf{v_j}$ are combined to form an expression matrix M, such that $M_{i,j} = v_{j,i}$.

4.3.1 Read normalization and differentially expressed sequences

To obtain counts comparable between samples, the raw read counts were normalized to log-CPM values using the R limma package, see section 2.5.2. For the datasets where I have looked at differential expression, the limma package was also used to find differentially expressed sequences. I used a p-value (adjusted by limma for multiple testing) of 0.05 as the cutoff for labeling a sequence as differentially expressed.

4.4 Finding short reads

In this section I will describe the identification of shorter 9-13 nt reads that align to observed isomiRs, which I deal with in section 5.3. I will simply refer to such reads as “short-reads”. I started with the same pipeline process as described above in section 4.3, but instead now keeping only reads of length 9-13 during adapter removal, to obtain a profile of such reads for each sample. For all datasets the resulting reads turned out to contain a large number of low complexity G/C sequences that generally will have many matches in the genome. To avoid having to further process these reads, and to prevent possible random precursor alignments, I skipped all reads containing only 2 or fewer A/U bases. Although this could lead to discarding short-reads that would align to a miRNA, the impact should not be very significant. The average short-read length is around 11 nt, and of the total 416 different miRNAs observed in the GCF and CH12 data, only

36 contain an 11 nt substring of 2 or fewer A/U bases anywhere in the miRBase reference mature sequence.

I then built a k-mer index out of the observed miRNA reads, mapping each seen k-length substring back to a list of reads containing the substring. To find miRNA reads fully containing a short-read as a substring, I used a k-length substring of the short-read to look up potential miRNA reads, and checked each of these for containing the complete short read sequence. In some parts of section 5.3, I have relied on associating short-reads with a specific isomiR. This causes some problems when there are multiple matches, as there really is no way of knowing which isomiR gave rise to a short-read. I have taken the following approach: the miRNA reads were scored based on how close the short read substring was to either the 5'- or 3' end of the miRNA read. This is best explained with an illustration, see Figure 4.3 and the accompanying description. For further ties, the miRNA read with the highest average expression was chosen.

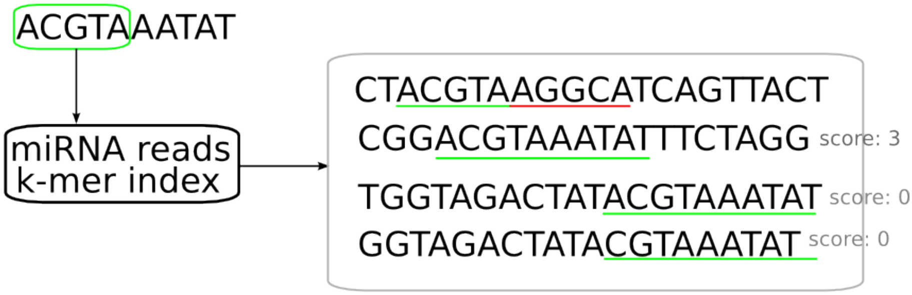


Figure 4.3: K-mer index search example for $k = 5$. There are 4 miRNA reads containing ACGTA, and three of these contain the full short read sequence (ACGTAAATAT). A score is computed as the smaller of the distance from the miRNA 5' end to the 5' end of the short sequence, and the distance from the miRNA 3' end to the 3' end of the short sequence. For the first matching miRNA sequence above, the 5' distance is the shortest. For the two following isomiRs the 3' distance is the smaller (0). Because of a tie between the two last isomiRs the one with the highest observed expression would be chosen.

4.5 Finding non-templated 3' additions

As mentioned in section 2.2.3, a non-templated isomiR refers to an isomiR whose sequence does not correspond to a sequence in the original gene. The work and results relating to non-templated isomiRs is presented is section 5.4. An example non-templated isomiR was given in Figure 2.6, where the last sequence has an AA tail at the 3' end not matching the precursor sequence. In this section I describe finding isomiRs with such non-templated 3' tails, when an isomiR not having the tail has been observed. The tail should consist of an l-length single-base sequence (e.g. AA or UUU) . For convenience I will below assume this base to be U, but the same method of course applies to other nucleotides. The process is also described with an example in Figure 4.4.

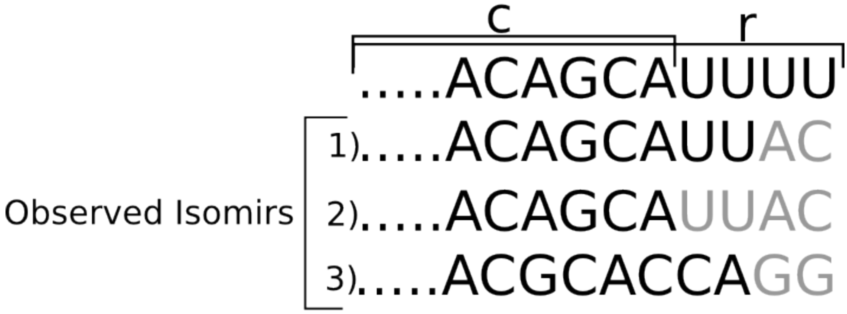


Figure 4.4: The read r contains a length 4 U-tail. After checking that no isomiR containing r has been observed, the tail is removed to give c. MicroRNA reads (marked 1, 2 and 3) containing c are found. The grey colored tails indicate the hairpin sequence, and are not part of the reads. Only 1) and 2) are substrings of r, and are considered further since r's tail does not correspond to the hairpin sequences. The longer read, 1), is compared against r, and r is found to have length 2 non-templated U-tail.

A read r is first checked for having a U-tail suffix of length at least l. The miRNA k-mer index described above in section 4.4 is reused to check that the read is not a substring of an observed isomiR. The tail is removed to obtain a cut sequence c, and the k-mer index is used to find miRNA reads containing the sequence c. Of these miRNA-reads the ones that also are substrings of r, and where r does not a have a match in the corresponding pre-miRNA sequence, are kept. These reads will have c as a prefix and a possible 3' U-tail (shorter than $r's$ U-tail). The longest of any remaining miRNA reads is compared against r, and r is kept if its U-tail is at least l nt longer than that of the miRNA read.

4.6 Sequence statistics

This section describes the ways in which I have analyzed and compared sequence data.

4.6.1 Sequence content and base-pairing

The two main statistics I have looked at and used to compare groups of miRNAs are position specific nucleotide content and base-pairing counts. With base-pairing counts for mature miRNAs I mean whether the pre-miRNA hairpin structure has base-pairs at the positions corresponding to the mature sequence.

Features of both the 5' and 3' end of sequences are known to be of importance for miRNAs. Because of the variable length of miRNAs I have therefore looked at sequence content and base-pairing counting both upstream from the 3' end and downstream from the 5' end. In addition, when looking at isomiRs and their respective end-reads I have used the end-read alignment offset as the starting position. These three different ways of counting are illustrated in Figure 4.5.

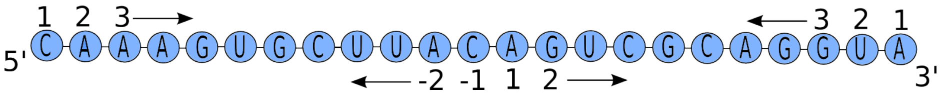


Figure 4.5: Three different ways of counting for either sequence content or base pairing; downstream from the 5' end, upstream from the 3' end, and both down- and upstream from a position corresponding to the alignment offset of an end-read. For the latter case, the figure illustrates the scenario where the end-read "AGUCGCAGGUA" has been observed.

4.6.2 Statistical testing

Having computed counts of either nucleotide content or base-pairing for groups of isomiRs, I have used statistical tests to find features that differ significantly between the groups. To do this I have mostly used *Fisher's exact test* (see for example [Weisstein, 2013]), by constructing a $2 * 2$ contingency table for each tested feature. For example, for testing whether the proportion of isomiRs having an A nucleotide at position 1 differs significantly between two groups, I construct and test the following table:

	1=A	**1≠ A**
group 1	a	b
group 2	c	d

where *a* is the number of isomiRs in group 1 having an A at position 1, and similarly for b, c and d.

An exception to this way of comparing features is the statistical feature selection in section 5.1, where I use a built in chi-squared based feature selection procedure from *scikit-learn* (see section 4.7).

4.7 Tools and languages used

For implementing the work of the project, I have mainly used the *Python* programming language, together with the *SciPy*[1] third party scientific computing library. When working with support vector machines I have used the excellent *scikit-learn*[2] machine learning library, which in addition to providing an easy to use SVM interface has a number of useful utilities, including feature selection and cross validation.

A small amount of work was also done using the R language with the *limma* package, specifically the normalization of read counts and identification of differentially expressed sequences, as mentioned in section 4.3.1.

[1] scipy.org

[2] scikit-learn.org

CHAPTER 5

Results and Discussion

In this chapter I present and discuss the work and results of the project. I start by looking at classification of up/down-regulated isomiRs in section 5.1, and continue in section 5.2 with looking at isomiRs of the same miRNA that experience different Ago2-knockout expression changes. The central problem addressed in these two first sections is to identify sequence features that determine the observed expression changes. In sections 5.3 and 5.4 I describe the work relating to short reads that align to the 3' end of miRNAs, and non-templated sequences, respectively.

5.1 Classification of up- and down regulated sequences

As mentioned in the introduction chapter, my main goal in [Mossin, 2013] was the classification of miRNAs by whether a miRNA would experience a higher, lower, or unchanged expression level in Ago2-knockout cells. The work was based on data containing fold change and average expression information aggregated per miRNA, thus ignoring possible differences between isomiRs. In the report I identified such isomiR-differences as a possible cause of noise, and discussed the possibility of improving the results obtained by taking individual isomiRs into consideration.

In this section I describe my attempt at creating a support-vector machine

#	**Feature description**
1-60	Nucleotide content for 15 first 5’ bases
61-120	Nucleotide content for 15 first 3’ bases
121-135	Base-pairing for 15 first 5’ bases
136-150	Base-pairing for 15 first 3’ bases
150-155	Overall G/C, A, C, G, U content

Table 5.1: The features used for classification.

classifier for predicting up-/down regulation of isomiRs in Ago2-knockout cells, utilizing all available sequence data for training and testing. A description of the classifier construction is first given in section 5.1.1, before the results are given in section 5.1.2. In essence, the classifier is not successful, and performs no better than random. A valuable lesson from the classifier results will be the importance of proper performance estimation, including handling highly similar sequences.

5.1.1 SVM construction

When mapping isomiRs to feature vectors, I have taken the approach of first automatically generating a large number of features, and then incrementally reducing the number of features used while observing classifier performance.

The underlying goal of the classifier is not simply successful classification in itself, but to gain more knowledge of the underlying biological processes involved in the miRNA pathway. Given a classifier achieving good performance, the biological interest will then center on which features are important for the classifier, and whether the features support or can be explained from previous knowledge. By measuring performance over different numbers of features, the goal was to find the minimum set of features of importance for the classifier.

The initial set of features (i.e. all considered features) was constructed from position specific nucleotide features, position specific pre-miRNA base-pairing features, and features based on overall nucleotide content. The details are given in table 5.1. All features are normalized to be in $[0, 1]$. For position specific nucleotide content this is done by each position contributing 4 features, one binary feature per nucleotide, where only one of the four will be “positive”. Base-pairing features are binary by nature, and overall nucleotide content features are included as percentages.

For an unbiased estimation of performance I have used a nested cross validation approach, shown in pseudocode in Algorithm 5.1. To avoid any overfitting, both feature selection and parameter setting is done by considering only the training data (X_{train}, Y_{train}). The most significant features are determined by a chi-squared test.

Algorithm 5.1 Nested cross validation. The inner CV-loop is represented by the call to *gridSearchCV*, which performs cross validation to select a parameter setting from the given parameter grid. A subset of the features are selected by calculating the most significant features from the training set.

```
function nestedCV(svm, paramGrid, numFeatures)
    scores ← List()
    outerCVFolds ← StratifiedCV(folds=3)
    for trainIndices, testIndices in outerCVFolds do
        $X_{train}, X_{test}$ ← trainTestSplit(X, trainIndices, testIndices)
        $Y_{train}, Y_{test}$ ← trainTestSplit(Y, trainIndices, testIndices)
        features ← kMostSignificantFeatures($X_{train}, Y_{train}$, k=numFeatures)
        svm.params ← gridSearchCV(svm, paramGrid)
        train(svm, $X_{train}, Y_{train}$)
        scores.append(score(predicted, $Y_{test}$))
    return scores
```

There are more miRNAs that are down-regulated in KO than up-regulated, around $\frac{2}{3}$ vs $\frac{1}{3}$, so the cross-validation is performed such that the relative group frequencies are retained within the folds (stratified cross-validation). Another important element that should be handled during cross-validation is the high similarity between many isomiRs. If such similar isomiRs are handled independently when creating cross-validation folds, this could heavily bias the estimated performance, as many isomiRs in the test fold can be expected to have one or more similar isomiRs in the training folds. I have therefore taken care to always put isomiRs of the same miRNA in the same fold.

5.1.2 The classifier has random performance

I have mainly used receiver operating characteristic (ROC) curves for evaluating performance. Figures 5.1 a) and c) shows the number of features used plotted against ROC-curve area, and a ROC-curve produced using 30 features, respectively. These results were produced using all the GCF data, and the results are equally weak when considering only the DOC or GH data. It is quite clear from these figures alone that the performance is no better than random.

This classification failure then suggest that up/down-regulation in Ago2 knockout cells is not determined by properties of the sequence structure. This conclusion contradicts the findings I reported in [Mossin, 2013], where the two-class classifiers achieved ROC-area scores of 0.65. An unfortunate but likely explanation for this contradiction is that my previous results were a result of overfitting. The approach I used in [Mossin, 2013] in summary consisted of first using all

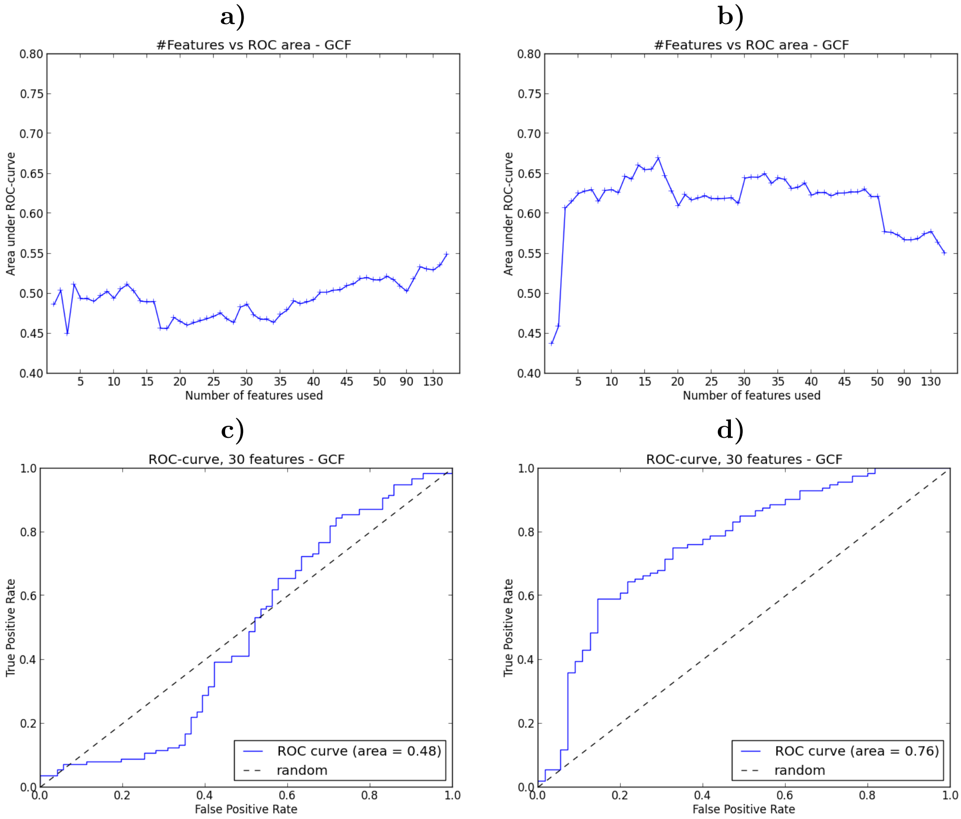


Figure 5.1: Performance estimation curves. In **a)** and **b)**: The number of features used for classification plotted against ROC-curve area. For each feature count K, the K most significant features are used in a 3-fold nested cross-validation procedure to produce estimates of the ROC-curve area. The score for each K is computed as the average of the 3 scores of the outer cross-validation loop. The curve in **a**) is produced using the steps of Algorithm 5.1 for each feature count. In **b**) the most significant features are computed before CV from all available data, giving a biased result. In **c)** and **d)**: ROC-curves obtained using 30 features. In **c)**) care is taken to put isomiRs of the same miRNA in the same fold when partitioning the data into training and testing folds The curve in **d)** illustrates the biased result obtained if this is not done.

data to find significant features, and then use these features when estimating the performance using CV. Because the data in the CV testing folds will have been part of the feature selection process, the reported results are susceptible to an optimistic bias.

If the procedure used to generate Figure 5.1 a) is changed so that the significant features are first determined using all available data, the stronger results from [Mossin, 2013] can be reproduced. Figure 5.1 b) shows the result of modifying Algorithm 5.1 so that the feature selection is done before the outer CV loop.

Another result of a biased performance estimation is shown in Figure 5.1 d). It illustrates the strong performance obtained if no care is taken to group similar isomiRs when creating training and testing folds.

5.2 Variation in differential expression between isomiRs

In addition to the classification of individual sequences discussed above in section 5.1, I have also looked at identifying pairs of isomiRs of the same miRNA that experience significantly different changes in expression between KO and WT samples, with the goal of finding common features that result in a higher or lower expression change for otherwise very similar sequences.

I first describe the modeling framework I have used to identify the isomiRs of interest in section 5.2.1, before discussing the significance of the obtained results in section 5.2.2. In section 5.2.3 I compare the isomiRs with high/low fold change to try to find differing features. Finding no significant features, I conclude that the differences seen between isomiR-pairs may simply be the result of a natural variation.

5.2.1 Identifying Significant isomiR pairs

The core of this problem can be described as follows: Given sample counts for each observed sequence, and under the null hypothesis that isomiRs of the same miRNA should show the same change in expression between WT and KO cells, then identify isomiR pairs where the expression changes observed for the two isomiRs are significantly different. And for such pairs, see if there are common features that differ between the isomiRs with a higher expression change and those with a lower expression change.

“Higher” and “lower” expression change are here meant relative to a pair of two isomiRs. If for one miRNA there are two isomiRs with KO fold changes of 5 and 1, and another miRNA with two isomiRs with KO fold changes of -1 and

-5, then the isomiRs with "higher" KO fold change are the isomiRs with fold change 5 and -1.

The problem does not lend itself well to standard tests of count data, such as e.g. a chi-square test on a contingency table of counts. A read "count" c cannot really be viewed as a count variable in the sense that it represents an outcome that has been observed c times. Rather, it represents *one* observation from an underlying distribution.

Below I will describe the modeling framework I have used. The idea is to create a point of (average expression, Δ logFC) for all pairs of isomiRs, estimate the variance in $\Delta logFC$ for different levels of average expression, and then find outliers in this data. A plot showing all such pairs is shown for the GCF data in Figure 5.2.

Using average expression to determine the expected $\Delta logFC$ distribution is likely to be too simplistic, as it can e.g. equal a pair consisting of one highly and one lowly expressed isomiR with a pair consisting of two more equally expressed isomiRs. Intuitively, pairs of one highly and lowly expressed isomiR should be expected to show a higher variance, due to more variance in the fold change of the lowly expressed isomiR. In an attempt to deal with this, the average expression is computed as the geometric mean, which for the log-counts here will tend to give a lower value to pairs with one lowly expressed isomiR, compared to using the arithmetic mean (for example: $\sqrt{3 * 10} = 5.48 < \frac{3+10}{2} = 6.5$). Additionally, I consider only isomiRs with average $\log_2$ expression (as reported by limma) larger than 2.0.

The variance is estimated by partitioning all points into buckets by rounding the average expression to the nearest integer, and calculating the variance for each bucket. A second degree polynomial is then fitted to the points to obtain a smooth estimator curve. To avoid estimating the variance with only a few points, all points with an average expression higher than a cut-off value are put in the same bucket. See Figure 5.2.

The normal distribution is used to model the data and find outliers. I have used a p-value of 0.05 as the significance cut-off. Figure 5.3 shows a Quartile-Quartile plot of the distribution of points within each bucket against the quartiles of normal distributions. The normal distributions have been plotted with variances equal to the variances of the respective buckets. The data can be seen to be approximately normally distributed, except for most buckets having heavier tails than the respective normal distributions.

5.2.2 Model evaluation

As can be seen in Figure 5.2, there are many isomiR pairs that experience quite different expression level changes. But an important question is to what degree

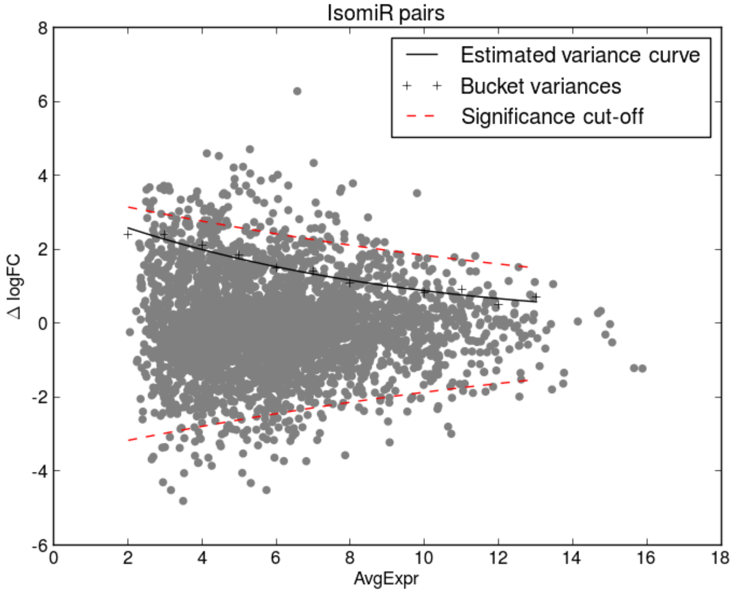


Figure 5.2: For each miRNA, all pairs of isomiRs are generated and converted to a point of (average expression, delta log fold change). A variance curve is estimated by grouping points by rounding average expression to the nearest integer. The dashed red line indicates the 0.05 significance cut-off for a normal distribution under the estimated variance. For the data shown here all points with average expression higher than 13 are represented by the last bucket.

the differences in fold change represent any real underlying biological difference, or if they rather are a result of natural variation and multiple testing.

From the plotted curves in Figure 5.2 the significance cut-off can be seen to vary approximately from a log fold-change between 3 and 2, depending on the average expression. As an argument for the soundness of the model, this level agrees well with the log fold-change required by *limma* to report an isomiR as differentially expressed for the same sequence data. And if such a fold change is significant for individual sequences, it seems sensible that a difference in fold-change of around the same magnitude should also constitute a significant difference.

A weakness of the above argument of comparing the fold change levels with those required in the limma analysis, is that there of course are many more pairs of isomiRs than individual isomiRs, and because of this more outliers can be expected. Looking more deeply at the data seems indeed to suggest that

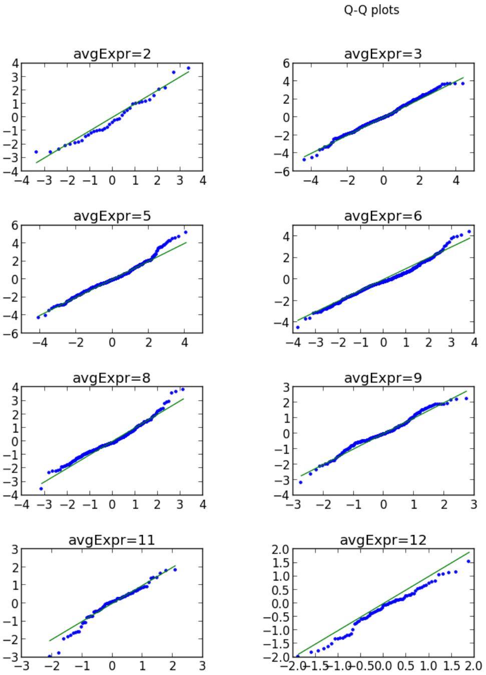


Figure 5.3: Q-Q plot of the distribution of points within each bucket vs respective normal distributions with equal variance.

the larger fold change differences are a result of multiple testing. As a first sign, if comparing the total number of isomiRs between those miRNAs for which there is at least one significant pair (significant as defined by the cut-off line in Figure 5.2) and other miRNAs, the first group have $\sim$ 9 observed isomiRs on average compared with only $\sim$ 3 isomiRs for the latter group (p-value $\simeq 1e-30$). This large difference may indicate that the significant pairs arise not due to biological differences, but simply because there are many observed isomiRs for some miRNA. But it can also be argued that many isomiRs gives more diversity in sequence structure, thus creating a higher probability of the isomiRs exhibiting different sequence features affecting the Ago2-knockout expression.

Figures A.1 - A.3 in the appendix show fold-changes and average expression for the isomiRs of the significant pairs. With some exceptions, the expression data shown in the figures show that the pairs are often made up of one lowly expressed isomiR and one more highly expressed isomiR. This seems to further strengthen the argument that the observed fold-change differences between isomiR pairs are a result of natural variation and multiple testing, not of a systematically different reaction to Ago2 removal.

5.2.3 Comparing the isomiRs

In an attempt to identify common features that differ between the isomiRs with a higher expression change and those with a lower expression change, I compared all "high" fold change isomiRs against all the "low" fold change isomiRs. Again, high and low are here meant relative to an isomiR pair, so that for each pair one isomiR is put in the "high" group and one in the "low" group. To avoid a bias from miRNAs where there are several significant pairs, only one pair per miRNA is considered (the pair with the strongest p-value). Working with the GCF, DOC and GH data, this then results in 42, 38 and 44 pairs, respectively. Fold-change and expression data for these isomiRs is given in Figures A.1 - A.3 (only for GCF).

Because 3' end differences are more common between isomiRs than differences at the 5' end (for the 42 GCF pairs all differ at the 3' end, with about half also differing at the 5' end), any significant features will likely only show up when counting upstream from the 3' end. Alternatively, those isomiR pairs that do differ at the 5' end can be analysed separately, but I have not done this due to the few such pairs in the data. Because the features I found in [Mossin, 2013] were based on counting downstream 5' to 3', this makes a comparison with the features found here difficult. Features for each of the GH, DOC, and GCF datasets with p-values stronger than 0.05 are listed in appendix B. The p-values have not been adjusted for multiple testing.

In summary, for the GCF and DOC data I found no features differing between the groups with p-values stronger than around 0.01, and generally few features with p-values better than 0.05. Considering the multiple testing performed, a few such p-values can be expected. For the GH data, a C base at position 12, counting 3' to 5', stands out with a p-value of 0.0001 (see Table B.3 for the count details). In [Mossin, 2013], I found the 5' downstream positions 10 and 11 to be among the most significant. Since Ago2 is known to cleave its targets at the bond between the 10th and 11th nucleotide, this was a biologically interesting result. With the average mature miRNA length being $\sim$ 22 nt, the 11th upstream position will often correspond to the 10th or 11th position counted downstream. But comparing the GCF, DOC and GH data I found no features, including the GH position 11 feature, to be consistently significant between the datasets. Though given the above discussion of the nature of the fold change variance, finding few features is not surprising, and in itself further supports the argument that the fold change differences are not the result of sequence differences.

5.3 Short reads aligning to miRNAs

I now move away from the central topic of the last two sections of finding features affecting differential expression in Ago-2-knockout cells. The raw read data contains a significant number shorter reads of around 10 nucleotides that align to mature miRNAs (here referred to as "short-reads"). The exact definition and the method used to find the short-reads was given in section 4.4. In section 5.3.1, I show that the majority of the short-reads that align to a miRNA align perfectly to the 3' end of an observed isomiR, i.e. the short read sequence is equal to the 3' suffix of the isomiR sequence. Such short-reads that align to an isomiR 3' end will just be denoted "end-reads". Section 5.3.2 presents the interesting find that end-reads are also found Ago2-knockout samples, and section 5.3.3 further shows a correlation between end-reads in the GCF and CH12 data. There are significant differences in which isomiRs have corresponding end-reads, and in section 5.3.4 I try to find sequence features that differ between the isomiRs that have and do not have end-reads.

With the exception of section 5.3.2, the work presented from here on is not specific to Ago2-knockout experiments, and I will in addition to the GCF data now also present results for the CH12 dataset.

5.3.1 The majority of short-reads are end-reads

Having first observed that many short-reads align to mature miRNAs, the further analysis was set off from the finding that the majority of these alignments coincide with the miRNA 3' end, as illustrated and described in Figure 5.4.

The numbers in Figure 5.4 are obtained from assigning each short-read uniquely to an isomiR, and therefore of course depend on how the mapping from short-read to isomiR is done, as there often will be several candidate isomiRs that align perfectly with a short-read. As described in more detail in section 4.4, I have prioritized isomiRs by how close the alignment is to the 5' or 3' end of the isomiR, which can be expected to give a bias for the two "0" positions in Figure 5.4. An alternative approach could for example be to use the matching isomiR with the highest observed expression. If one assumes no prior knowledge of the biological processes causing short-reads to be observed, this would arguably give the maximum-likelihood isomiR. As expected, if Figure 5.4 is recreated using this method, the 3' bars over "-2", "-1" and "0" are more equally sized, with $\sim 25\%$ end-reads, as shown in Figure 5.5. This maximum expression approach

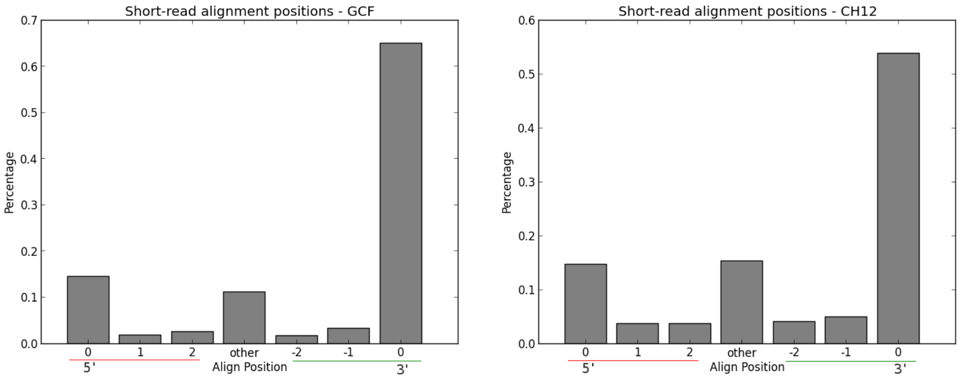


Figure 5.4: Alignment of short-reads against isomiRs for the GCF (left) and CH12 (right) data. Each short-read is matched to an observed isomiR by the method described in section 4.4, and contributes to one of the bars depending on its alignment to the isomiR. The first three bars, underlined in red and marked as 5', represent those isomiRs where the short-read aligns with an offset of 0, 1 or 2, counting from the 5' end of the isomiR. For the bars underlined in green and marked as 3', the offset is counted from the 3' end (i.e. the distance from the isomiR 3' to the short-read 3' in the alignment). The end-reads are represented by the rightmost bar: they have an align offset of 0 when counting from the 3' end. The middle bar counts all other short-reads that are not captured by the six 5' and 3' bars.

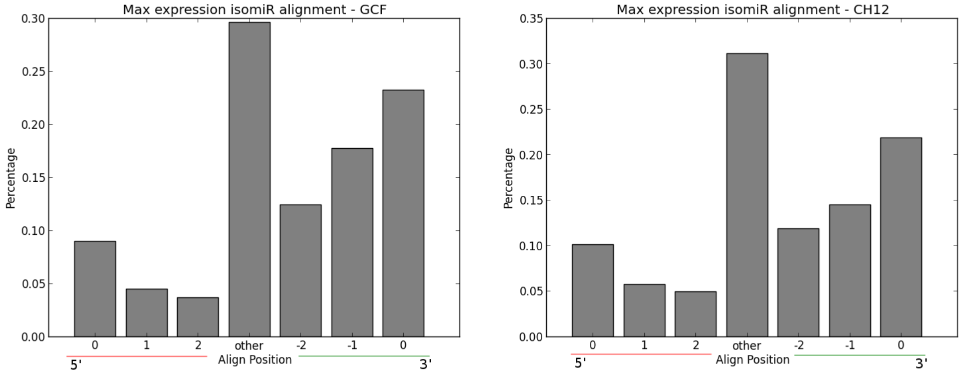


Figure 5.5: Alignment of short-reads against isomiRs, using the matching isomiR with highest expression. Results for GCF and CH12 datasets are given in the left and right plots, respecivly. See Figure 5.4 for how to interpret the numbers.

can be seen as a simplification of a more robust probabilistic analysis, where each isomiR is attributed for some fraction of the observed short-reads that align to it, depending on its own expression level.

But although my approach likely has some degree of bias, no matter the method used to map short-reads to isomiRs it is clear that a non-random fraction of short-reads do align to the 3' end. I believe this observation invalidates the "maximum-likelihood" argument above, and justifies the method I have used.

5.3.2 End-reads are found in both WT and Ago2-KO samples

As Ago2 is known to cleave the passenger strand during RISC activation, a natural first guess to the cause of the observation of end-reads would be that they are produced from Ago2 cleavage. But under this assumption, one would of course expect that the level of end-reads observed would be significantly lower in Ago2 knockout cells. I found this not to be the case. Figure 5.6 shows end-read counts for WT vs KO.

Some differences should be expected between WT and KO, due to many sequences being differentially expressed. Specifically, more miRNAs are down-regulated than up-regulated in knockout. And except for a few exceptions, most end-reads have low read counts, which naturally leads to some random differences between the samples. But the overall trend is clear: end-reads are found both in WT and Ago2-knockout samples, and there is a clear correlation between KO

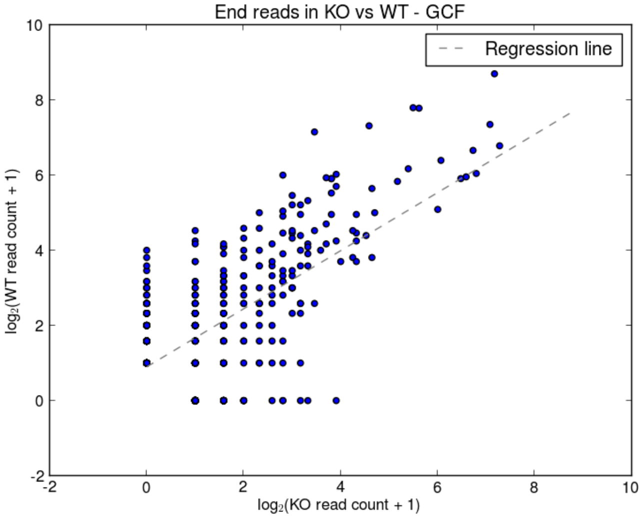


Figure 5.6: End-reads in WT vs KO. The figure shows both that end-reads occur in Ago2-KO samples, and that there is a clear correlation between read counts in WT and KO.

and WT read counts.

5.3.3 End-read counts correlate between GCF and CH12

In the previous section, the end-read count correlation was between WT and KO samples from the same dataset. I have also looked at correlation between datasets, specifically between the GCF and CH12 samples. Figure 5.7 plots read counts for end-reads in GCF against CH12, and shows a clear correlation. Because isomiR expression levels vary considerably between the datasets, with some isomiRs more highly expressed in CH12 and vice versa, differences are expected. The rather large number of points along the lines $x = 0$ and $y = 0$ can likely be explained by isomiRs with large expression differences between the datasets.

This correlation between datasets signify end-reads are produced as the result of a regulated degradation process. As I discuss in the summary chapter, it would be interesting to analyse more datasets and see how well the correlation holds. As all data I have worked with originate from the same lab, analysing new datasets with a different origin should give valuable information on the generality of the observed correlation.

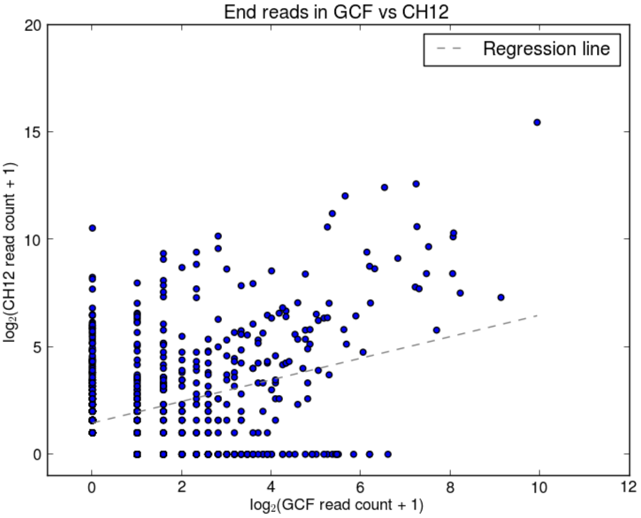


Figure 5.7: End-read counts in GCF vs CH12. Of the higher expressed end-reads in GCF almost all are also highly expressed in CH12.

5.3.4 Differences in end-read counts between miRNAs

There is a big difference in the number of end-reads observed for different miRNAs, and also between isomiRs of the same miRNA. Of course, highly expressed isomiRs are expected to have more matching end-reads, but although I have found an overall correlation between isomiR expression and end-read expression, there are large deviations from the general trend. This is illustrated by the plots in in Figure 5.8.

My focus has been on finding the cause of the differences seen: Why do some miRNAs give rise to more end-reads than others? To try to determine this, I have looked at finding features that differ significantly between the sequences that have many corresponding end-reads and those that have few or none. A complicating factor in the analysis is that an end-read can align to the 3' end of several isomiRs. Even though isomiR 3' modifications are more common than differences at the 5' end, some ismomiRs still have identical 3' suffixes. A point in Figure 5.8 therefore represent all isomiRs with equal suffix. The question is then what to do when comparing the groups for differing sequence features. I have taken the approach of using for each point the isomiR with the highest expression (of the isomiRs that are represented by the point).

I have used two different approaches when testing for significant features. In

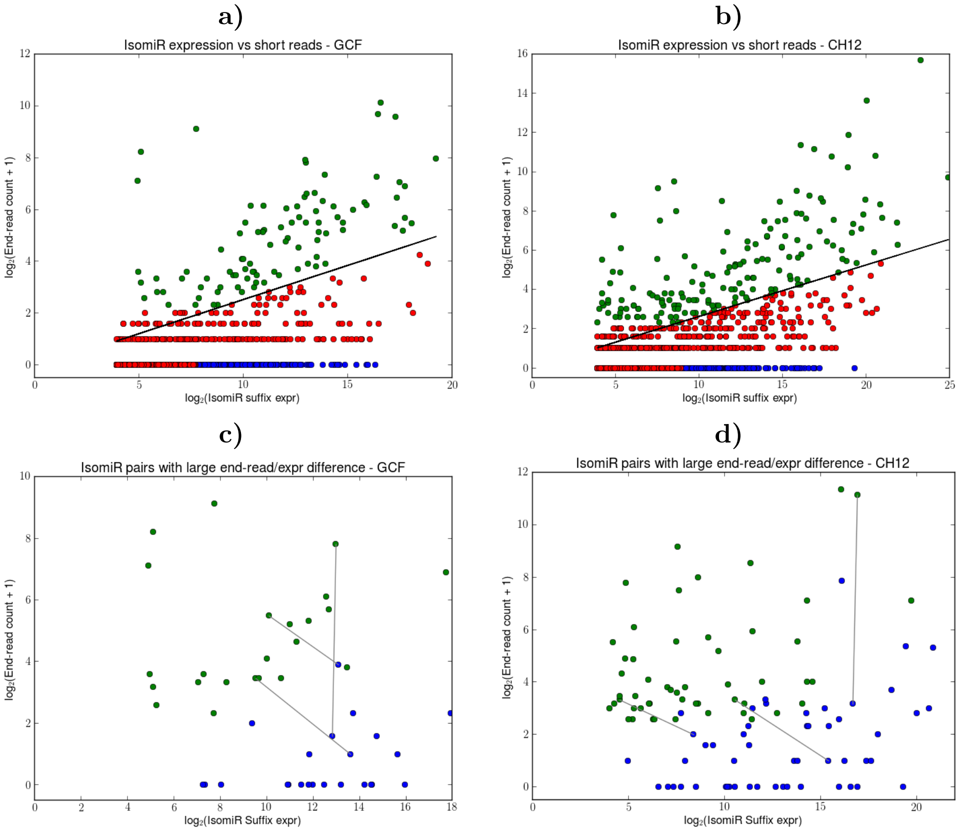


Figure 5.8: In **a)** (GCF) and **b)** (CH12): IsomiR expression summed by suffix plotted against the number of end-reads aligning to the suffix. Because an end-read can map perfectly to the 3' end of several isomiRs, isomiR expression values are summed for isomiRs having equal suffixes, counted from the last 9 bases. This suffix count is then plotted against the total number of short-reads having the same suffix. A clustering of the points is done by first finding the overall trend of suffix expression vs short-read count by fitting a regression line. The points marked in green represent the cluster of suffixes having many short-reads given the isomiR suffix expression. Conversely, the suffixes with few short-reads in relation to the isomiR suffix expression are marked in blue. All remaining points are marked in red. In **c)** (GCF) and **d)** (CH12): The plots show a subset of the points from the respective plots in in a) and b). The distance along the $(1, -1)$ vector is measured for all pairs of points representing isomiR suffixes from the same miRNA. I have then used a distance of 3 or more along this vector to define points that show large differences in the ratio of isomiR expression vs end-read expression. Each miRNA contributes at most one pair of points. The clustering is performed by labeling each point as the "high" or "low" point of its pair. Three pairs of points are illustrated in the plots by a connecting line.

the first approach, illustrated and described in Figure 5.8 a) and b), each suffix is clustered independently depending its isomiR- and end-read expression values. The second approach is similar to the analysis in section 5.2; I find pairs of isomiR suffixes that have large differences in isomiR/end-read expression, and construct one group of the isomiR suffixes with many end-reads and one group of those with few end-reads (in relation to isomiR expression). See Figure 5.8 c) and d) and the accompanying description. The first technique thus looks more generally at differences between sequences that have end-reads and those that do not, while the latter technique looks for common differences between pairs of isomiRs with many/few end-reads. I will denote the two approaches as the "global" and the "pair-based" clusterings, respectively. For the pair-based analysis, only differences counted upstream from the 3' can be expected, since for most pairs the two isomiRs will have equal 5' ends. Though for both the GCF and CH12 data the isomiRs of all significant pairs have different 3' ends.

The features found for both types of comparisons are given by the tables in Appendix C, which lists all features with a p-value less than 0.05. I have found no very strong results, with most of the listed features having p-values in the range $0.01 - 0.05$. This can not be considered significant under the multiple testing performed, and as expected given the p-values, few features are consistent between the datasets. The two strongest features both have p-values of 0.004. The first is for the global CH12 analysis, where the isomiRs with many end-reads are found to have a relative preference for a G base at 3' position 3. The second is for the pair-based analysis of the GCF data, where no isomiRs from the "low"-group have a C base at the 3' end. But neither of these two features are found to be strong in both datasets. Perhaps the most interesting observation is that three of the four base-pairing features are for positions that will be close to the center of the sequences (9, 11, and 13, counted upstream). This could suggest the stability of the precursor in this region is of importance. But the argument is weakened by observing that the three features differ on which group has a preference for base-pairing. To look more into this, I summed the base-pairs in positions 9-13 for each isomiR, and then compared the values for the two groups with a t-test. I ran the test twice, once counting the positions downstream and once counting upstream, but did not get any significant results.

Thus the analysis result is that I have found no clear features that differ between miRNAs with many/few end-reads. Some features are found with p-values ~ 0.01, but there is little consistency between the datasets. But even so, it should still be interesting to look at additional datasets to get a stronger indication of whether any of the features are of any real biological interest.

5.4 Non-templated A- and U tails

In the previous section, I discussed the possibility that the observed end-reads could be a result of a miRNA degradation process. As the stability and degradation of some miRNAs has been reported to be regulated by the addition of non-templated A- and U tails at the 3' end, I have also analyzed the sequence data for miRNA reads with such non-templated additions. I will refer to these reads that except for a non-templated single-nucleotide 3' tail align to a mature miRNA as "tail-reads". The method and criteria used to identify the tail-reads was covered in section 4.5.

In section 5.4.1, I first show that I have found many A/U tail-reads, and that these reads correlate between the GCF and CH12 data, while G/C additions are hardly found. Section 5.4.2 then presents an analysis similar to that performed in the previous section, where the goal is to find features differing between miRNAs with many/few corresponding tail-reads. Finally, section 5.4.3 studies my main reason for including this section on tail-reads: If both non-templated 3' additions and the end-reads discussed in the previous section are connected to miRNA degradation processes, it could be expected that they target different miRNAs. I have found no proof of this, though.

5.4.1 A/U tails are common, G/C tails are rare

The first observation that is clear when analysing both the GCF and CH12 data for non-templated 3' tails, is that there are a considerable number of reads with A- and U tails, while G- and C tails are much more uncommon. Additionally, although some isomiRs seem to experience only A- or U-tailing, there is in general a clear correlation between the number of U-tails and the number of A-tails observed for an isomiR. See Figure 5.9. Because of this correlation, I have chosen to treat A/U-tailing together in the coming sections; summing the number of reads of A- and U tails observed for an isomiR.

I have also found that there is a correlation between which isomiRs have tail-reads in the GCF and CH12 data, shown in Figure 5.10. The figure is similar to the corresponding result for end-reads shown in Figure 5.7. And as I discussed for the end-read result, considerable deviations from the general correlation is expected, due to sequences that are highly expressed in one dataset and lowly expressed in the other. The correlation between the GCF and CH12 data, as well as the observation that A/U-tailing is much more common than G/C-tailing, agrees with A/U-tailing playing a role in miRNA degradation.

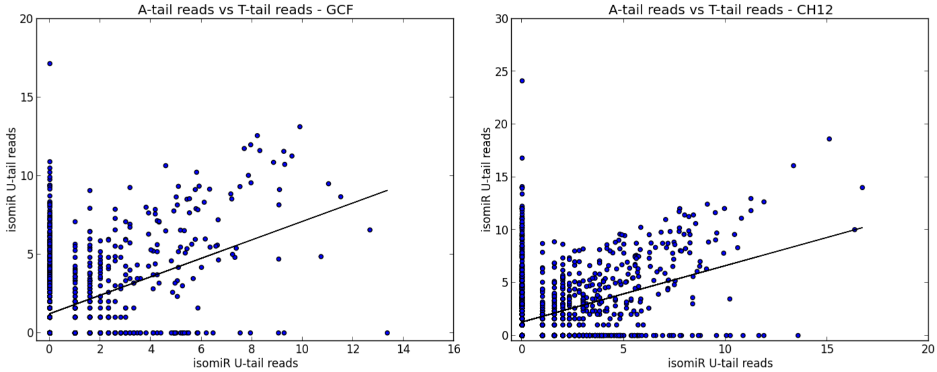


Figure 5.9: The number of A-tail reads vs U-tail reads, with fitted regression lines, for the GCF (left) and CH12 (right) data.

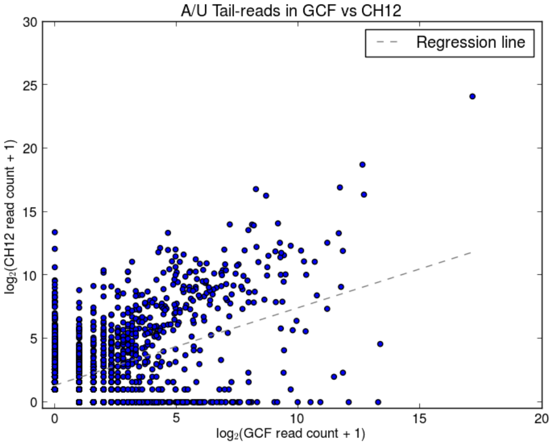


Figure 5.10: Tail-read counts in GCF vs CH12.

5.4.2 Differences in tail-read counts between miRNAs and isomiRs

Similar to what I found for end-reads, the ratio of sequence expression to the number of corresponding tail-reads differ significantly both at the miRNA and isomiR level. While there is a strong general correlation between sequence expression and corresponding tail-reads, there are large variations.

I have performed an analysis analogous to that performed for end-reads in section 5.3.4, with the goal of investigating whether sequence features can explain why some miRNAs experience 3' tailing while others do not. Like I did for end-reads, I perform the analysis using both a "global" comparison approach, where each isomiR is clustered independently based on its expression level and corresponding tail-read count, and a "pair-based" approach, where I find pairs of isomiRs of the same miRNA that have different tail-read/expression ratios. This is illustrated and further explained in Figure 5.11.

The features found for both clustering approaches and both the GCF and CH12 datasets are listed in Appendix D in the appendix. As can be read from the tables, there are a few very significant features; the most significant feature has a p-value of 5e-10. Adjusting for multiple testing, the significance cutoff will be around 0.0005[1], so that the stronger features are still significant. These features are all for nucleotide-preferences at either position 1 or 3 at the 3' end, and come up as significant in both the global and pair-based analyses of both datasets. The preferences can be summarized as follows: At 3' position 1, the isomiRs with few tail-reads have a bias towards a C base and rarely have an A base, while the isomiRs with many tail-reads have a bias for an A base and rarely have a C base. At 3' position 3, the preferences are the opposite of those for position 1 (i.e. the isomiRs with many tail-reads have a bias for a C base, etc). In addition to the strong results obtained for these features, there are several less significant (not significant when adjusting for multiple testing), including some other weaker preferences for positions close to the 3' end, though these are not as consistent between the datasets and clustering approaches.

As the isomiRs with corresponding (A/U) tail-reads are found to have a preference for an A at the 3' end, it should be noted that such a bias is not created by the method used to find tail-reads and map them to corresponding isomiRs. In fact, a U base at position 1 actually comes up as a preference for isomiRs with few tail-reads in the GCF data.

The nucleotides of the 3' end thus appear to affect which miRNAs experience non-templated A/U additions at the 3' end. And sequence variations at the 3' end then also seem to explain the differences seen between otherwise equal isomiRs, which suggest these variations are biologically relevant and can affect the relative

[1] Roughly 100 tests are performed, a simple correction gives: 0.05 / 100 = 0.0005

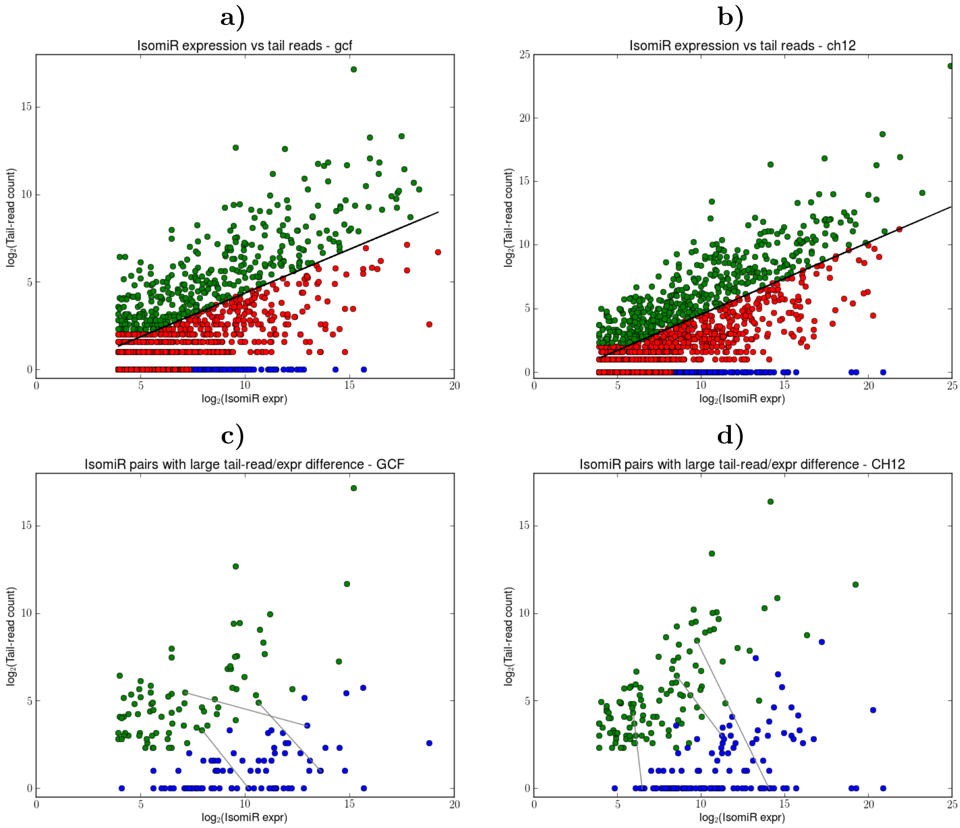


Figure 5.11: In **a)** (GCF) and **b)** (CH12): IsomiR expression plotted against the read count for corresponding tail-reads. A clustering is performed by first finding the overall trend of isomiR expression vs tail-reads by fitting a linear regression line. The points marked in green represent the cluster of isomiRs having a high degree of corresponding tail-reads in relation to the isomiR expression, while the blue points represent the cluster of isomiRs with high expression and no tail-reads. All remaining points are marked in red.
Plots **c)** (GCF) and **d)** (CH12) show a subset of the points from a) and b), respectively. The difference along the vector (1, -1) is measured for all pairs of isomiRs of the same miRNA. I have used a distance of 3 or more along this vector to define isomiRs that show large differences. Each miRNA contributes at most one pair to avoid any bias in the analysis. For each pair, each isomiR is labeled by whether it has the higher or lower ratio of tail-reads to isomiR expression.

expression of isomiR variants.

5.4.3 Tail- and end-read correlation

My main reason for first starting to analyse the project datasets for tail-reads was to try to find out whether miRNAs targeted by A/U-tailing are less likely to give rise to end-reads, and vice versa. If end-reads and tail-reads represent products connected to two different degradation mechanisms, it could perhaps be expected that most miRNAs are predominantly targeted by only one of the two respective processes.

To investigate this, I have looked at the relationship between how many corresponding end-reads and tail-reads are observed for each isomiR. As I discussed in section 5.3.4, end-reads generally can not be mapped uniquely to isomiRs, only to the set of isomiRs containing the end-read sequence as a 3' suffix. For a correct comparison of the number of tail-reads against the number of end-reads observed for isomiRs, tail-read counts must then also be summed for isomiRs with equal suffixes. The resulting GCF and CH12 plots are shown in Figure 5.12.

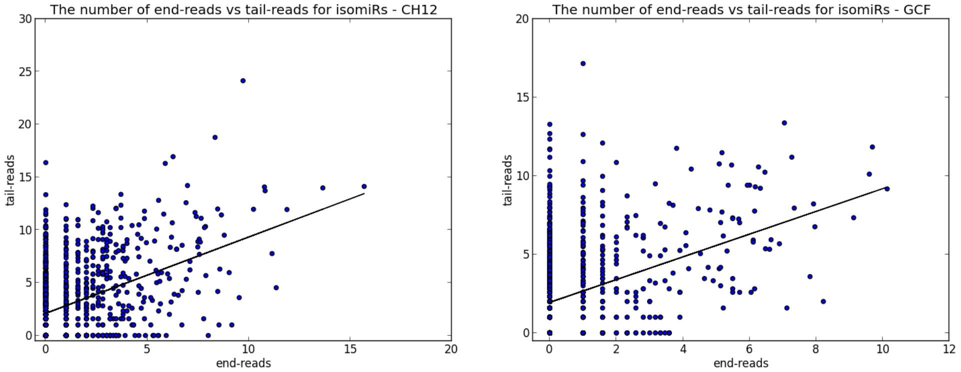


Figure 5.12: End-reads vs tail-reads

From Figure 5.12 alone it is quite clear that there at least is no very strong exclusiveness in targeted miRNAs, as clearly many isomiRs (or more precisely, groups of isomiRs with equal suffixes) have both high end-read and tail-read counts. For a statistically stronger test of this result, I have used the observed data of isomiR expression versus end-read counts (the data shown in Figure 5.8) and isomiR expression versus tail-read counts (shown in Figure 5.11) to create distributions over observed end-read and tail-read counts, given the expression of an isomiR: $Pr(\textit{end-read count} \mid \textit{isomiR expression})$, and $Pr(\textit{tail-read count} \mid$

isomiR expression). The distributions are created by rounding the isomiR expression values of Figures 5.8 and 5.11 to the nearest integer, so that for each integer expression value there are empirical distributions over end-read and tail-read counts. To test the hypothesis that miRNAs tend to have either corresponding end-reads or tail-reads, but not both, to a larger degree than can be expected by chance, I run a simulation by sampling random end-read and tail-read counts for each isomiR based on its expression. Some example simulation results are shown in Figure 5.13 a). To compare the data produced by the simulations to the result presented in Figure 5.12 a linear regression line is fitted to to each simulation outcome. Figure 5.13 b) shows the distribution of fitted linear regression slopes for 1000 simulations. The slope from 5.12 is marked in red, and can be seen to be consistent with the simulation results.

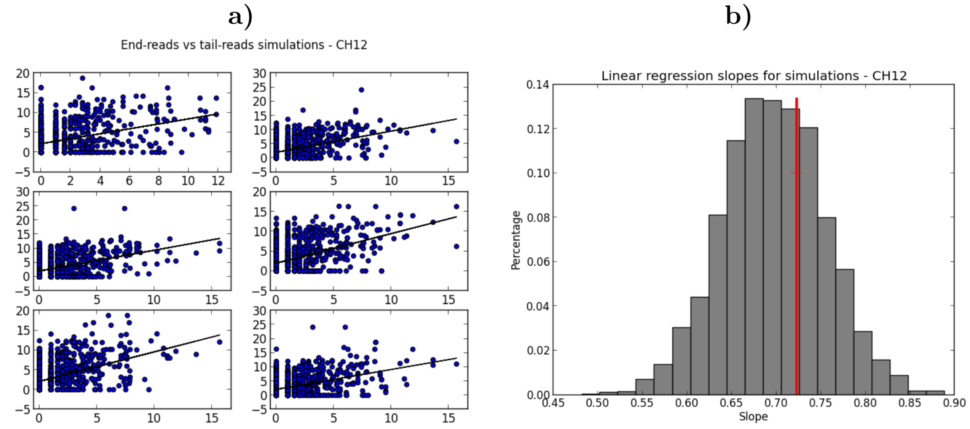


Figure 5.13: End-reads vs tail-reads simulations. In **a)**: The outcome of six example simulations. In **b)**: The distribution of fitted linear regression slopes for 1000 simulations. The red line shows the slope of the regression line from Figure 5.12.

The conclusion from this comparison of end-reads and tail-reads, is then that there is no significant lack of overlap between which isomiRs have matching end-reads and which isomiRs have matching tail-reads.

CHAPTER 6

Summary and Future Work

This chapter first gives a summary of the presented work, before looking at possible future work.

6.1 Summary and conclusion

This project was started from the goal of improving upon the results obtained in [Mossin, 2013], by taking observed isomiR variations into consideration when predicting Ago2-knockout expression changes. From working with the raw sequencing output from an Ago2-knockout experiment I obtained sample read counts for individual isomiR sequences, and analysed each sequence for differential expression. From this analysis, I constructed an SVM classifier for determining whether an isomiR would be up- or down regulated in Ago2-knockout cells. Given that proper care is taken in handling bias from highly similar isomiRs, the classifier is not successful. As the classifier is constructed from features derived from sequence content and pre-miRNA base-pairing, this suggests expression changes in Ago2-knockout is primarily determined by other factors.

I have also presented an analysis on whether there are miRNAs where isomiR variants experience significantly different Ago2-knockout expression changes. To this end, I developed a modeling framework based on fitting normal distributions to the differences seen between pairs of isomiRs. I concluded that although some pairs of isomiR show quite different expression changes, overall the differences appear to be due to natural variation and multiple testing.

Working directly with the sequencing data also led me to discover a group of short $\sim$ 10 nt reads that align to mature miRNAs. For the majority of these reads, the alignment was found to coincide with the 3' end of the mature sequence (termed "end-reads"). Interestingly, this result is found both in wildtype and Ago2-knockout samples. This implies end-reads are not products of Ago2-cleavage, and the likely explanation is that they are produced from a different degradation process. End-reads are observed for a large number miRNAs, though the numbers differ significantly both between miRNAs and between isomiRs of the same miRNA. I attempted to find sequence features differing between sequences having many and few/none corresponding end-reads, but found no significant results.

Working under the hypothesis that end-reads are products of a miRNA degradation process, I further analysed the project data for isomiRs with 3' non-templated A/U-tails, which previously have been suggested to affect miRNA stability and degradation. I found that whether a miRNA experiences 3' tailing is affected by position specific nucleotide preferences at the 3' end, and that these preferences also cause differences between isomiRs of the same miRNA. This could potentially represent a biologically relevant difference between isomiRs. Finally, I looked at whether miRNAs with corresponding end-reads are less likely to be targeted by 3' tailing, and vice versa, but found no evidence of this.

6.2 Future work

In this section I will discuss areas where the work of the report can be either improved or extended.

Other features affecting Ago2-knockout expression

It can be argued that the problem of predicting differential expression is more naturally modeled as a regression problem, since the fold change observed for each sequence is a continous variable. But because the classification results I have obtained are so weak, it seems likely sequence features are of little importance in affecting the expression change. Simply substituting regression for classification should therefore not be expected to perform well. But of course, the extreme differences in expression change must have some cause. Future work should therefore be focused on investigating other possible features, e.g. are there strong correlations within miRNA families?

Modeling of differential expression differences between isomiRs

In section 5.2 I concluded that isomiRs of the same miRNA generally seem to experience the same expression changes in Ago2-knockout, and that the differences seen between some isomiRs were likely a result of multiple testing. But the modeling framework I used is obviously rather simple, and as described in chapter 3 on related work, similar problems have previously been treated using statistically more sophistacted techniques. It would be interesting to apply the GLM-based method described in chapter 3 for a more robust analysis.

End-read/isomiR mapping

The fact that an end-read can not be mapped uniquely to an isomiR poses a problem when looking at which isomiRs have corresonding end-reads and which do not. When comparing sequence features between these two groups of isomiRs, I took the straightforward approach of first finding all isomoRs such that the alignment coincides perfectly with the isomiR 3' end, and then used the most highly expressed isomiR of these to compute sequence features. An alternative approach could be to model the problem in a probabililistic way, where each isomiR aligning with a short-read is given some probability of being the source of the short-read. This probability could depend on a number of factors, including isomiR expression and the offset of the alignment,

Generality of end- and tail-read results

Perhaps the most natural first step for continuing the work of this report, is to extend the work on end- and tail-reads for more datasets. Especially more evidence for the universality of end-reads would be very interesting. Sequencing data from published miRNA experiments can be obtained from public databases such as *ArrayExpress*[1]. I have tried to write the project code in such a way that new datasets can be processed with little manual work, making it possible to automatically perform the same analysis on a large number of datasets.

[1] www.ebi.ac.uk/arrayexpress/

Bibliography

Anders, S., Reyes, A., and Huber, W. (2012). Detecting differential usage of exons from rna-seq data. *Genome Research*, 22(10):2008–2017.

Barber, D. (2012). *Bayesian reasoning and machine learning.* Cambridge University Press.

Bartel, D. P. (2009). MicroRNAs: Target recognition and regulatory functions. *Cell*, 136:215–233.

Ben-Hur, A. and Weston, J. (2010). A user's guide to support vector machines. *Methods in Molecular Biology*, 609:223–239.

Bowtie (2014). Bowtie: Manual. [Online; accessed 18-April-2014].

Burrows, M., Wheeler, D. J., Burrows, M., and Wheeler, D. J. (1994). A block-sorting lossless data compression algorithm. Technical report, test.

Cenik, E. S. and Zamore, P. D. (2011). Argonaute proteins. *Cell*, 21:R446–R449.

Cloonan, N., Wani, S., Xu, Q., Gu, J., Lea, K., Heater, S., Barbacioru, C., Steptoe, A. L., Martin, H. C., Nourbakhsh, E., et al. (2011). MicroRNAs and their isomiRs function cooperatively to target common biological pathways. *Genome Biol*, 12(12):R126.

Cock, P. J., Fields, C. J., Goto, N., Heuer, M. L., and Rice, P. M. (2010). The sanger fastq file format for sequences with quality scores, and the solexa/illumina fastq variants. *Nucleic acids research*, 38(6):1767–1771.

Durbin, R. (1998). *Biological sequence analysis: probabilistic models of proteins and nucleic acids.* Cambridge university press.

Farazi, T. A., Brown, M., Morozov, P., ten Hoeve, J. J., Ben-Dov, I. Z., Hovestadt, V., Hafner, M., Renwick, N., Mihailović, A., Wessels, L. F., et al. (2012). Bioinformatic analysis of barcoded cdna libraries for small RNA profiling by next-generation sequencing. *Methods*, 58(2):171–187.

FASTX-Toolkit (2014). fastx_collapser [fix]. [Online; accessed 20-Jan-2014].

Ferragina, P. and Manzini, G. (2000). Opportunistic data structures with applications. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 390–398. IEEE.

Gu, S. and Kay, M. A. (2010). How do miRNAs mediate translational repression? *Silence*, 1(1):1–5.

Kai, Z. S. and Pasquinelli, A. E. (2010). Microrna assassins: factors that regulate the disappearance of mirnas. *Nature structural & molecular biology*, 17(1):5–10.

Kawamata, T., Seitz, H., and Tomari, Y. (2009). Structural determinants of miRNAs for RISC loading and slicer-independent unwinding. *Nature structural & molecular biology*, 16(9):953–960.

Kozomara, A. and Griffiths-Jones, S. (2011). mirbase: integrating microrna annotation and deep-sequencing data. *Nucleic acids research*, 39(suppl 1):D152–D157.

Langmead, B., Trapnell, C., Pop, M., Salzberg, S. L., et al. (2009). Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biol*, 10(3):R25.

Law, C. W., Chen, Y., Shi, W., and Smyth, G. K. (2013). Voom! precision weights unlock linear model analysis tools for rna-seq read counts. *Preprint 2013.*

Lee, R., Feinbaum, R., and Ambros, V. (1993). The c. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14. *Cell*, 75:843–854.

Lee, R. C. and Ambros, V. (2001). An extensive class of small RNAs in caenorhabditis elegans. *Science*, 294(5543):862–864.

Liu, J., Carmell, M. A., Rivas, F. V., Marsden, C. G., Thomson, J. M., Song, J.-J., Hammond, S. M., Joshua-Tor, L., and Hannon, G. J. (2004). Argonaute2 is the catalytic engine of mammalian RNAi. *Science*, 305(5689):1437–1441.

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, 17(1):10–12.

Mattick, J. S. and Makunin, I. V. (2006). Non-coding rna. *Human molecular genetics*, 15(suppl 1):R17–R29.

McCarthy, D. J., Chen, Y., and Smyth, G. K. (2012). Differential expression analysis of multifactor rna-seq experiments with respect to biological variation. *Nucleic acids research*, 40(10):4288–4297.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Higher Education.

Mossin, J. (2013). Predicting the effect of ago2-knockout on microRNA expression levels. CS specialisation project, TDT4501, NTNU.

Nakamura, M., Kondo, S., Sugai, M., Nazarea, M., Imamura, S., and Honjo, T. (1996). High frequency class switching of an lgm+ b lymphoma clone ch12f3 to lga+ cells. *International immunology*, 8(2):193–201.

Neilsen, C. T., Goodall, G. J., and Bracken, C. P. (2012). Isomirs–the overlooked repertoire in the dynamic micrornaome. *Trends in Genetics*, 28(11):544–549.

Nilsen, T. W. and Graveley, B. R. (2010). Expansion of the eukaryotic proteome by alternative splicing. *Nature*, 463(7280):457–463.

Okamura, K., Hagen, J. W., Duan, H., Tyler, D. M., and Lai, E. C. (2007). The mirtron pathway generates microrna-class regulatory RNAs in¡ i¿ drosophila¡/i¿. *Cell*, 130(1):89–100.

O'Carroll, D., Mecklenbrauker, I., Das, P. P., Santana, A., Koenig, U., Enright, A. J., Miska, E. A., and Tarakhovsky, A. (2007). A slicer-independent role for Argonaute 2 in hematopoiesis and the microRNA pathway. *Genes and development*, 21(16):1999–2004.

Sætrom, P. and Snøve Jr, O. (2004). A comparison of sirna efficacy predictors. *Biochemical and biophysical research communications*, 321(1):247–253.

Saito, T. and Sætrom, P. (2010). Micrornas–targeting and target prediction. *New biotechnology*, 27(3):243–249.

Smyth, G. K. (2005). Limma: linear models for microarray data. In *Bioinformatics and computational biology solutions using R and Bioconductor*, pages 397–420. Springer.

Smyth, G. K. et al. (2004). Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Stat Appl Genet Mol Biol*, 3(1):3.

Soifer, H., Rossi1, J. J., and Sætrom, P. (2007). MicroRNAs in disease and potential therapeutic applications. *Mol. Ther.*, 15:2070–2079.

Sung, W.-K. (2011). *Algorithms in bioinformatics: A practical introduction.* CRC Press.

Wang, D., Zhang, Z., O'Loughlin, E., Lee, T., Houel, S., O'Carroll, D., Tarakhovsky, A., Ahn, N. G., and Yi, R. (2012). Quantitative functions of argonaute proteins in mammalian development. *Genes & development*, 26(7):693–704.

Wang, Z., Gerstein, M., and Snyder, M. (2009). Rna-seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63.

Weisstein, E. W. (2013). Fisher's exact test. `http://mathworld.wolfram.com/FishersExactTest.html` - A Wolfram Web Resource. Accessed: 2013-21-05.

Wikipedia (2014). Scaled inverse chi-squared distribution. [Online; accessed 16-March-2014].

Wyman, S. K., Knouf, E. C., Parkin, R. K., Fritz, B. R., Lin, D. W., Dennis, L. M., Krouse, M. A., Webster, P. J., and Tewari, M. (2011). Post-transcriptional generation of mirna variants by multiple nucleotidyl transferases contributes to mirna transcriptome complexity. *Genome research*, 21(9):1450–1461.

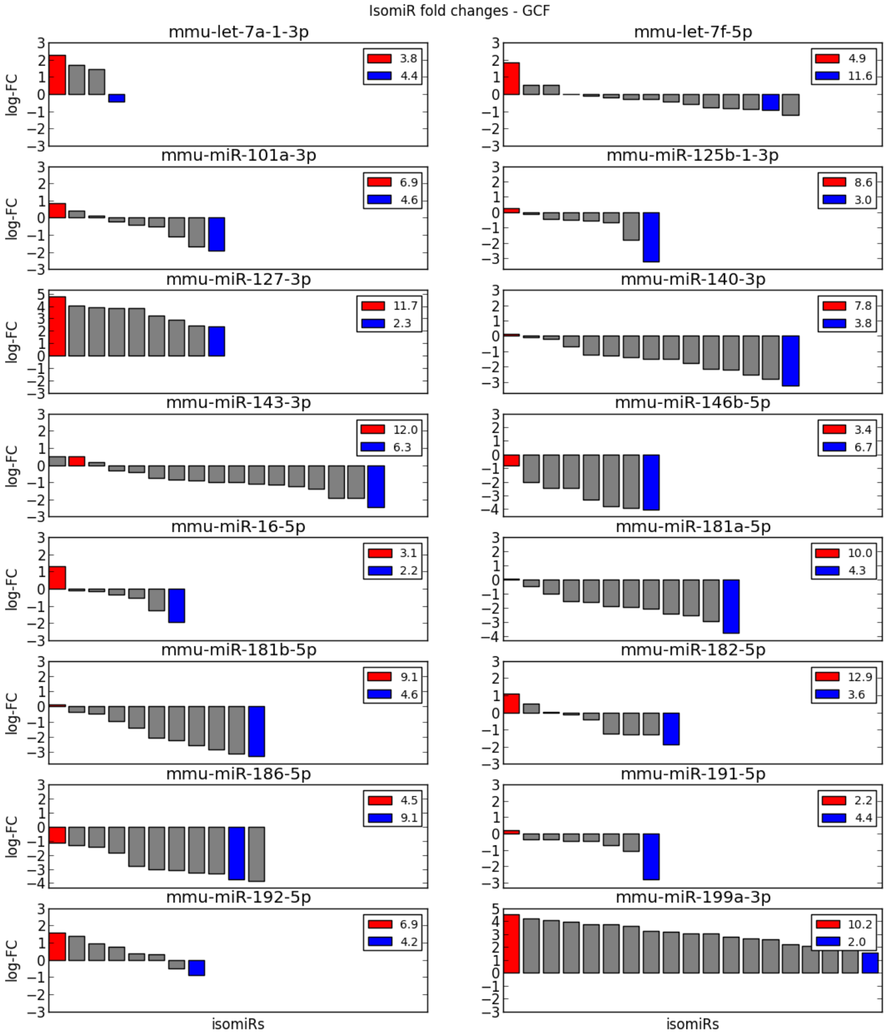
Xue, C., Li, F., He, T., Liu, G.-P., Li, Y., and Zhang, X. (2005). Classification of real and pseudo microrna precursors using local structure-sequence features and support vector machine. *BMC bioinformatics*, 6(1):310.

Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic acids research*, 31(13):3406–3415.

Appendix

A Isomir fold-change figures

Figure A.1: IsomiR Ago2-KO fold changes. See the description in Figure A.3.



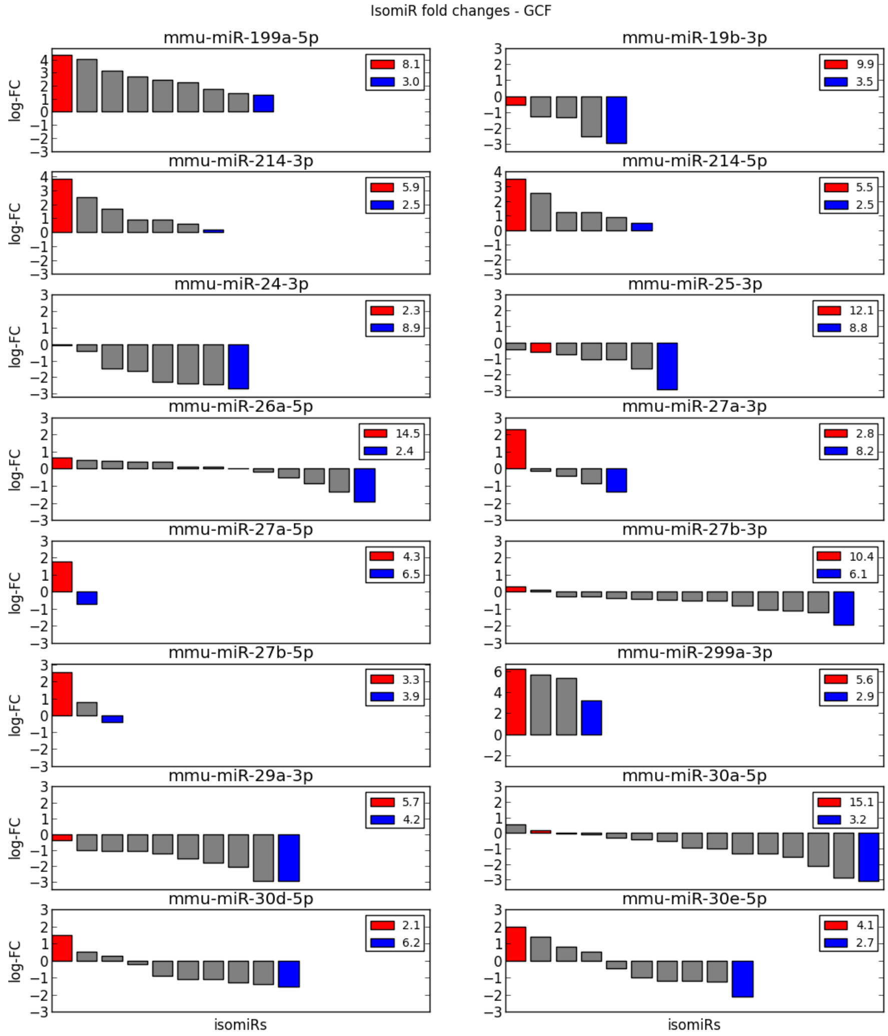


Figure A.2: IsomiR Ago2-KO fold changes. See the description in Figure A.3.

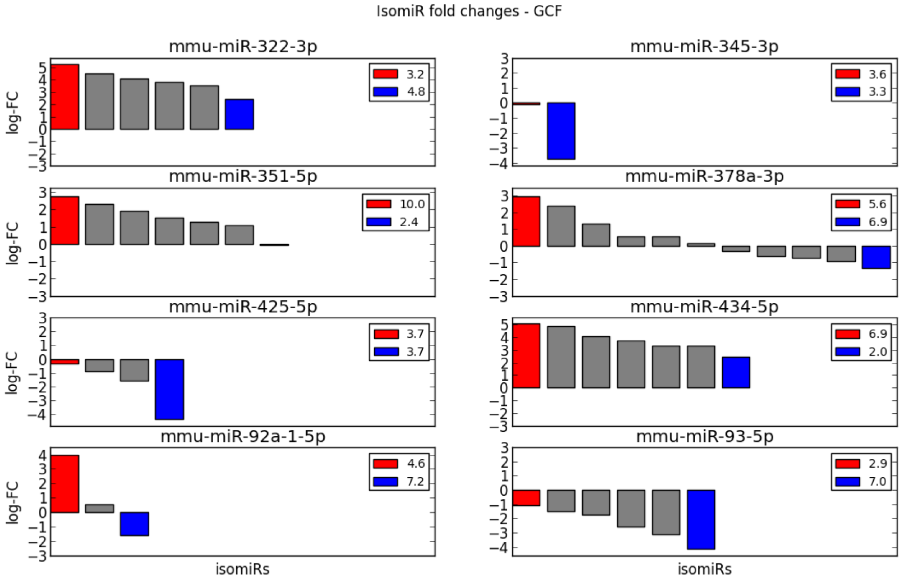


Figure A.3: Ago2-knockout $\log_2$ fold-changes for isomiRs of miRNAs where there is atleast one isomiR pair showing a significant difference in fold-change, as illustrated and defined in Figure 5.2. The "high" and "low" isomiRs that together forms the most significant pair for each miRNA are shown in red and blue, respectivly. The average expression for these two isomiRs are also shown in the legend boxes in the upper right corner of each plot. The data for mmu-miR-10a-5p, which has 30 different observed isomiRs, is not included due to the limited space available for each plot.

B Features derived from IsomiR-pairs comparison

Feature	p-value	counts
C at -12	0.029	(7, 35), (17, 25)
C at -6	0.049	(12, 30), (4, 38)
C at -10	0.049	(16, 26), (7, 35)
Base-pair at -13	0.049	(38, 4), (30, 12)

Table B.1: Features with p-values < 0.05 from the GCF isomiR-pairs comparison. Negative positions indicate the position is counted 3' to 5'. Positions are 1 index, i.e. 1 is the first 5' position and -1 is the first 3' position. The numbers in the *counts* column indicate for the number of isomiRs from the "high" (i.e. high Ago2-KO fold-change) and "low" groups, respectivly, that have and do not have a given feature: The first row the table table shows that there are 7 isomiRs from the high group with a C at position -12, and 35 isomiRs from the same group with some other base at position -12. Similarly for the "low" group, the numbers are 17 and 25.

Feature	p-value	counts
G at -6	0.006	(5, 39), (17, 27)

Table B.2: Features with p-values < 0.05 from the DOC isomiR-pairs comparison

Feature	p-value	counts
C at -12	0.0001	(2, 36), (17, 21)
A at -12	0.010	(13, 25), (3, 35)
A at -14	0.012	(11, 27), (2, 36)
C at -10	0.025	(17, 21), (7, 31)
Base-pair at -13	0.027	(36, 4), (27, 13)
G at -2	0.045	(16, 22), (7, 31)
A at -10	0.047	(4, 34), (12, 26)

Table B.3: Features with p-values < 0.05 from the GH isomiR-pairs comparison

C End-read analysis features

Feature	**p-value**	**counts**
G at -3	0.004	(20, 85), (34, 55)
C at -5	0.010	(32, 73), (13, 76)
G at -8	0.015	(20, 85), (31, 58)
G at 16	0.022	(20, 85), (30, 59)
G at 1	0.023	(3, 102), (10, 79)
C at -6	0.025	(25, 80), (10, 79)
Base-pair at -11	0.027	(61, 28), (87, 18)
C at -11	0.029	(14, 91), (23, 66)
T at -8	0.030	(32, 73), (15, 74)
A at 13	0.036	(22, 83), (31, 58)
T at 10	0.038	(23, 82), (32, 57)
T at 8	0.039	(30, 75), (14, 75)
G at 2	0.047	(27, 78), (12, 77)
A at 9	0.047	(15, 90), (23, 66)
Base-pair at -9	0.048	(83, 6), (88, 17)
A at -1	0.049	(22, 83), (9, 80)

Table C.1: Features with p-values < 0.05 from the CH12 “global” clustering analysis of end-reads. Positions are 1 indexed, i.e. 1 is the first 5’ position and -1 is the first 3’ position. The numbers in the *counts* column indicate for the number of isomiRs from the “high” (i.e. many end-reads) and “low” groups, respectivly, that have and do not have a given feature: The first row of the table shows that there are 20 isomiRs from the high group with a G at position -3, and 85 isomiRs from the same group with some other base at position -3. Similarly for the “low” group, the numbers are 34 and 55.

Feature	**p-value**	**counts**
G at -11	0.009	(22, 30), (9, 43)
Base-pair at -13	0.021	(45, 7), (34, 18)
G at -2	0.024	(25, 27), (13, 39)
G at -3	0.037	(12, 40), (23, 29)
T at -3	0.040	(18, 34), (8, 44)
A at -2	0.041	(3, 49), (11, 41)

Table C.2: Features with p-values < 0.05 from the CH12 “pair-based” clustering analysis of end-reads.

Feature	**p-value**	**counts**
T at 4	0.014	(28, 95), (3, 44)
Base-pair at -1	0.014	(44, 3), (95, 28)
G at 6	0.036	(29, 94), (19, 28)
C at -7	0.037	(24, 99), (3, 44)

Table C.3: Featuxres with p-values < 0.05 from the GCF “global” clustering analysis of end-reads.

Feature	**p-value**	**counts**
C at -1	0.004	(0, 23), (8, 15)

Table C.4: Features with p-values < 0.05 from the GCF “pair-based” clustering analysis of end-reads.

D Tail-read analysis features

Feature	p-value	counts
A at -3	8e-05	(30, 55), (18, 126)
A at -1	0.0001	(9, 76), (48, 96)
C at -1	0.001	(27, 58), (19, 125)
C at -13	0.009	(11, 74), (41, 103)
G at 1	0.018	(9, 76), (4, 140)
G at -4	0.022	(31, 54), (32, 112)
G at -10	0.027	(14, 71), (43, 101)
G at 12	0.032	(16, 69), (46, 98)
T at -13	0.034	(32, 53), (34, 110)
T at -1	0.042	(36, 49), (41, 103)
C at 1	0.045	(16, 69), (45, 99)
A at -6	0.049	(21, 64), (20, 124)

Table D.1: Features with p-values < 0.05 from the CH12 “global” clustering analysis of tail-reads. Positions are 1 indexed, i.e. 1 is the first 5’ position and -1 is the first 3’ position. The numbers in the *counts* column indicate for the number of isomiRs from the “high” (i.e. many tail-reads) and “low” groups, respectivly, that have and do not have a given feature: The first row of the table shows that there are 30 isomiRs from the high group with an A at position -3, and 55 isomiRs from the same group with some other base at position -3. Similarly for the “low” group, the numbers are 18 and 126.

Feature	p-value	counts
C at -1	5e-11	(45, 67), (5, 107)
A at -1	2e-10	(8, 104), (49, 63)
A at -3	0.0004	(37, 75), (14, 98)
G at -4	0.0008	(41, 71), (18, 94)
C at -13	0.017	(23, 89), (40, 72)
G at -1	0.019	(15, 97), (30, 82)
T at -1	0.032	(44, 68), (28, 84)
T at -8	0.0336	(31, 81), (17, 95)
G at -3	0.047	(22, 90), (36, 76)

Table D.2: Features with p-values < 0.05 from the CH12 “pair-based” clustering analysis of tail-reads.

Feature	**p-value**	**counts**
C at -1	3e-05	(24, 50), (12, 127)
A at -11	0.0004	(26, 48), (19, 120)
A at -1	0.002	(6, 68), (36, 103)
A at -3	0.004	(21, 53), (16, 123)
A at -6	0.016	(21, 53), (19, 120)
C at -3	0.0187	(8, 66), (34, 105)
C at -6	0.025	(10, 64), (38, 101)
A at -14	0.028	(24, 50), (26, 113)
C at -12	0.029	(21, 53), (21, 118)
G at 7	0.032	(9, 65), (35, 104)
T at -2	0.034	(18, 56), (54, 85)
G at -9	0.038	(14, 60), (45, 94)
T at -4	0.041	(15, 59), (47, 92)
T at -14	0.043	(16, 58), (49, 9)

Table D.3: Features with p-values < 0.05 from the GCF “global” clustering analysis of tail-reads.

Feature	**p-value**	**counts**
C at -1	8e-08	(32, 47), (4, 75)
A at -1	0.0001	(5, 74), (24, 55)
A at -3	0.049	(22, 57), (11, 68)

Table D.4: Features with p-values < 0.05 from the GCF “pair-based” clustering analysis of tail-reads.