Methods 59 (2013) 154–163

Contents lists available at SciVerse ScienceDirect

Methods

journal homepage: www.elsevier.com/locate/ymeth

Identifying transcriptional miRNA biomarkers by integrating high-throughput sequencing and real-time PCR data

Sven Rahmann[a,b,d,*], Marcel Martin[a,b], Johannes H. Schulte[c], Johannes Köster[b,d], Tobias Marschall[e], Alexander Schramm[b,c]

[a] *Bioinformatics, Computer Science XI, TU Dortmund, Germany*
[b] *Collaborative Research Center (SFB) 876, TU Dortmund, Germany*
[c] *Pediatric Oncology, University Hospital Essen, Germany*
[d] *Genome Informatics, Faculty of Medicine, University of Duisburg-Essen, Germany*
[e] *Centrum Wiskunde & Informatica (CWI), Amsterdam, The Netherlands*

ARTICLE INFO

Article history:
Available online 23 October 2012

Communicated by Michael W. Pfaffl

Keywords:
miRNA
Biomarker identification
High-throughput sequencing
Read mapping
RT-qPCR
Normalization

ABSTRACT

Using both high-throughput sequencing and real-time PCR, the miRNA transcriptome can be analyzed in complementary ways. We describe the necessary bioinformatics pipeline, including software tools, and key methodological steps in the process, such as adapter removal, read mapping, normalization, and multiple testing issues for biomarker identification. The methods are exemplified by the analysis of five favorable (event-free survival) vs. five unfavorable (died of disease) neuroblastoma tumor samples with a total of over 188 million reads.

1. Introduction

Current high-throughput sequencing (HTS) technologies offer the opportunity to characterize the genomic, epigenomic and transcriptomic state of a tumor. Here we focus on the bioinformatic methodology of characterizing the microRNA (miRNA) transcriptome of a sample. Functional miRNAs regulate the translation and cleavage of mRNAs by sequence-specific interaction with the 3′ UTR, reviewed in [1]. MiRNAs are involved in the regulation of many physiological processes, including differentiation, development and apoptosis [2]. In cancer, miRNAs may exert oncogenic function by inhibiting tumor suppressor genes or may act as tumor suppressors by inhibiting oncogenes [3]. The goal is to identify putative biomarkers, i.e., miRNAs that are differentially expressed between tumor and surrounding tissue, or between low-risk and high-risk subtypes of the tumor. In a previous study, we found differential miRNA expression between favorable versus unfavorable neuroblastoma subtypes [4].

In this article, we discuss the fundamental challenges involved in estimating expression values from short RNA reads and describe the computational pipeline for obtaining a ranked list of differentially expressed miRNAs from the raw sequence reads. Expression levels of these biomarker candidates should be confirmed by RT-qPCR, and we discuss how logarithmic HTS expression values correspond to negative ΔC_q values. To provide some guidance for other experiments, we illustrate each step with numerical examples from our previous neuroblastoma study [4].

2. Material and methods

2.1. Datasets

The dataset of our previous study consists of five low-risk patients (neuroblastoma stage 1, no MYCN amplification, event-free survival [EFS], labeled 552–556) and five high-risk patients (neuroblastoma stage 4 with MYCN amplification, died of disease [DoD], labeled 557–561). This dataset can be retrieved from the NCBI Sequence Read Archive [5] using Accession No. SRA009986.

In the form presented here, the pipeline expects color space [6] FASTA (`.csfasta`) and quality (`.qual`) files, which is the format output by a SOLiD run. The downloaded `.sra` files can be converted with `abi-dump` from the NCBI SRA SDK.

* Corresponding author at: Genome Informatics, Faculty of Medicine, Institute of Human Genetics, University of Duisburg-Essen, Hufelandstr. 55, 45122 Essen, Germany. Fax: +49 201 723 5826.

E-mail addresses: Sven.Rahmann@uni-due.de, Sven.Rahmann@tu-dortmund.de (S. Rahmann).

URL: http://www.rahmannlab.de/ (S. Rahmann).

1046-2023/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ymeth.2012.10.005

For the methods of sample preparation, RNA isolation, small RNA enriched library generation with the small RNA expression kit (SREK), SOLiD sequencing and miRNA RT-qPCR protocols on the example dataset, we refer to the Materials and Methods section of our previous study [4].

2.2. Required software

For the bioinformatics pipeline, the following software is required: Python[1] (version ⩾3.2) with the following additional packages (in their Python 3 versions): pysam, matplotlib (with PDF output support), numpy, and scipy. From our group, snakemake[2] [7] (version ⩾1.3), cutadapt[3] [8] (version ⩾1.1), and sqt[4] are required. Necessary external tools are BWA[5] [9] (a version <0.6 for color space data (for SOLiD reads); color space support was disabled with version 0.6), SAMtools[6] [10], BEDTools[7] [11] (version ⩾2.16).

2.3. Required resources

The following required resources are downloaded automatically when running our pipeline: the human genome reference assembly GRCh37 [12], as used by the 1000 Genomes Project,[8] the ENSEMBL genome annotation track,[9] and a FASTA file with mature miRNAs of all organisms provided by the most recent miRBase version 18 (as of June 2012; released on November 2, 2011)[10] [13], as well as the release 8.1 version of this file.[11]

3. Fundamental challenges

Before we discuss the steps of the pipeline, we highlight three fundamental difficulties that arise during the analysis of short (mi) RNA reads.

3.1. Short sequences do not map uniquely against a large genome

Mapping all reads against the human genome (see Section 4.3) serves the purpose of obtaining a global picture about which RNA types are contained in the library at which levels. For example, we may assess the success of mRNA and rRNA depletion in the sample.

However, mapping short reads against a whole genome creates a fundamental problem: sequences with a length of approximately 22 bp are not always uniquely mappable, even without error tolerance. We are not aware of detailed uniqueness statistics of the human genome in the literature, so we provide them here (Fig. 1).

If a random position is picked in the human genome (GRCh37) and the 22-mer starting at that position is examined, there is a 23.4% chance that this 22-mer occurs somewhere else in the genome, or a 76.6% chance for uniqueness. Fig. 1 plots the uniqueness fraction for k-mers as a function of k. Because of long repeats in the human genome, the fraction of unique k-mers never reaches 90% in the plotted range. It makes little sense to attempt to map a DNA fragment of length 18 or below.

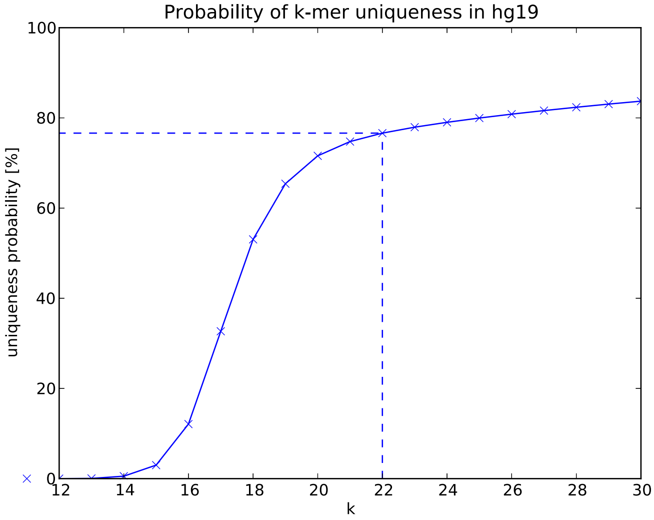

Fig. 1. Chance that a k-mer at a random strand and position in the human genome (GRCh37) is unique within the genome and its reverse complement, shown as a function of k. The 76.6% chance at $k = 22$ is highlighted by dashed lines.

These statistics have been obtained by constructing the enhanced suffix array [14] of Genome Reference Consortium Human genome build 37 (GRCh37) and its reverse complement using the algorithm described in [15] and considering the longest common prefixes (lcp table) of lexicographically adjacent suffix pairs. If at position p, the maximum of the adjacent longest common prefix lengths is ℓ, then the $(\ell + 1)$-mer at position p is unique. Only those suffix pairs were considered where the unique $(\ell + 1)$-mer consists of proper nucleotides (no IUPAC wildcards, no sequence separators).

3.2. Mature miRNAs do not map uniquely against the human genome

While the statistics of Fig. 1 are true in general for short sequences, the same cautionary statement holds for miRNAs in particular. We mapped all 1898 unique mature miRNA sequences in miRBase release 18 against GRCh37 with BWA (see also Section 4.3) and found that 370 of them cannot be mapped back uniquely and were often mapped to locations outside of annotated miRNAs since other locations in the genome share the same sequence. It has also been reported that some miRNAs cross-map to tRNAs because of sequence similarity [16]. However, there is a significant length difference between mature miRNAs (20–23 nt) and tRNAs (70–90 nt). Thus a solution is to map all reads of typical mature miRNA length not against the genome, but specifically against the mature miRNAs in miRBase.

3.3. Some miRNA share high sequence similarity

To obtain miRNA expression levels, we map specifically against miRBase. Most mature miRNA sequences are dissimilar to each other; however, there are a few distinct miRNAs with very similar sequences (edit distance of 1 or 2); see Fig. 2. While this is no *a priori* reason for concern, we must keep these relations in mind when evaluating biomarker candidates.

4. Automated miRNA expression analysis

Computing the expression profiles of all miRNAs from raw sequence reads involves several steps. In addition to a textual description, a well-documented and formalized workflow description is necessary to reproduce each single step. We use the

[1] http://www.python.org/.
[2] http://code.google.com/p/snakemake/.
[3] http://code.google.com/p/cutadapt/.
[4] http://code.google.com/p/sqt/.
[5] http://bio-bwa.sourceforge.net/.
[6] http://samtools.sourceforge.net/.
[7] http://code.google.com/p/bedtools/.
[8] ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/reference/human_g1k_v37.fasta.gz.
[9] ftp://ftp.ensembl.org/pub/release-67/gtf/homo_sapiens/Homo_sapiens.GRCh37.67.gtf.gz (version 67).
[10] ftp://mirbase.org/pub/mirbase/CURRENT/mature.fa.gz (most recent release).
[11] ftp://mirbase.org/pub/mirbase/8.1/mature.fa.gz (release 8.1).

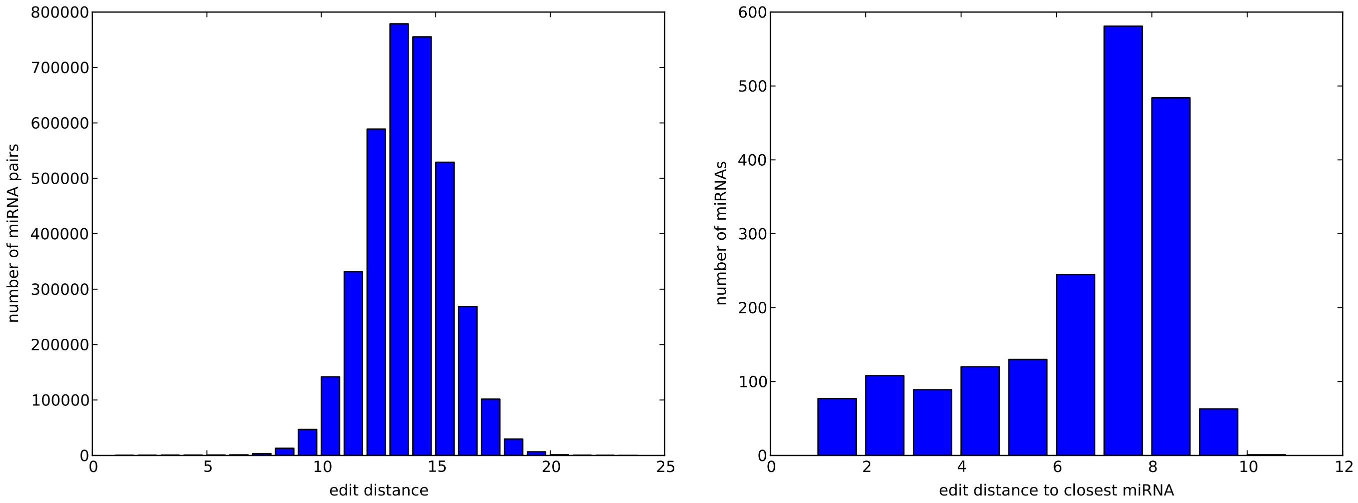

Fig. 2. Left: distribution of edit distances between all pairs of miRNAs in miRBase release 18 (November 2011). Right: distribution of edit distance to most similar miRNA among all miRNAs of miRBase release 18.

workflow system `snakemake` [7] to describe the bioinformatics pipeline in a way that is both formal and human-readable and can be visualized in graphical form (cf. Section 4.1).

We then describe each key step in more detail. For each dataset (patient), the following steps are executed: pre-processing, adapter removal and quality control of the raw sequence files obtained from the sequencing service (Section 4.2), followed by read mapping to the genome (for a global overview for RNA types in the library) and to the target miRNA transcripts (for accurate expression level estimation; Section 4.3). Once the raw miRNA counts have been obtained for each patient, several steps follow that operate on all datasets jointly, most importantly normalization between experiments (Section 4.4) and detection of differential expression between two classes (Section 4.5), which is the first step for biomarker identification. Potential biomarkers must then be verified technically with an independent method, and biologically on independent samples (i.e., other patients). We discuss how sequencing-based expression values are compared with quantification cycle (C_q) values from RT-qPCR (Section 4.6).

4.1. Workflow management with snakemake

Several interactive graphical workflow specification and execution systems exist, as well as text-based systems. Here we use `snakemake` [7], which uses a textual representation to both document and formally specify a workflow in an executable way. The workflow is described by defining rules that specify how a group of output files is created from a group of input files by executing a specific set of commands. `Snakemake` reads the rules from a `Snakefile`, combines them and executes the resulting workflow. The system automatically determines which sequences of rules must be applied to create a desired file from existing input files, and which rules can be skipped because their (intermediate) results are already present. The `snakemake` specification is independent of how the output files are created, whether by executing an external script, a shell command, calling a web service, or submitting a job to a compute cluster and collecting the results. One focus of `snakemake` is human readability; this is why we present the pipeline in this form.

4.2. Pre-processing and adapter removal with cutadapt

The initial quality control and filtering of the raw reads depends on the sequencing technology used. As most vendor software comes with integrated quality control and filter options, we do not discuss these in detail.

All technologies produce reads that are longer than the expected length of mature miRNAs (20–23 nt); e.g., SOLiD produces at least 35-mers. Thus, reads containing a mature miRNA also contain (part of) the adapter sequence at the end, and it is necessary to remove the adapters before aligning the reads to the reference transcriptome.

If the reads are obtained in dinucleotide color space (using ABI SOLiD sequencing [6]), adapter removal should also occur in color space; if the reads are obtained in nucleotide space, adapter removal should occur in nucleotide space as well. A tool that supports both scenarios is `cutadapt` [8].

The start position of the adapter sequence within the read is located using a free-end-gap (also called semiglobal or overlapping) sequence alignment with a carefully chosen error rate (see below). Computation of full alignments requires time proportional to the product of read length, error rate and adapter length, but since both sequences are short, adapter removal takes two minutes per million reads on a typical desktop computer, using a single core of an Intel Core 2 Quad (Q9400) processor at 2.66 GHz.

At the same time, cutadapt may convert different input formats (e.g., separate `.csfasta` and `.qual` files) into the common `.fastq` format usable by most read mappers. Fig. 3 shows the part of the `Snakefile` that implements adapter cutting. We set the maximum error rate to 20% for the following reason: since we use 35 bp reads, those reads that contain miRNAs of the most common lengths 20–23 nt contain adapters of lengths 15–12 nt (corresponding to 14–11 colors in color space [6]). The error rate used by cutadapt is defined as the number of errors divided by the length of the matching part of the adapter. Since $\lfloor 14 \cdot 0.2 \rfloor = 2$ and $\lfloor 11 \cdot 0.2 \rfloor = 2$, an error rate of 20% ensures that in most miRNA-containing reads two errors are allowed in the part of the read mapping to the adapter. One should be aware that an unfortunate choice of error rate could introduce a bias. Short miRNAs may preferentially be found simply because the trailing adapter is longer: since the length is multiplied with the error rate, short matches are allowed to contain n errors, while slightly longer matches allow $n + 1$ errors. For example, we find that reducing the error rate from 20% to 17% reduces the number of reads containing 23 nt miRNAs by 15% while the number of shorter trimmed reads changes by less than 3%.

When considering the distribution of read lengths after adapter removal, we expect a peak in the range of 20–23 nt, which gives

```
ADAPTER = ’330201030313112312’ # in color space
CUTADAPT_MINLENGTH = 15 # minimum length of reads after adapter removal
CUTADAPT_OVERLAP = 5 # minimum overlap of read and adapter to cut
CUTADAPT_ERROR_RATE = 0.2

rule run_cutadapt_on_dataset:
    input:
        qual=’reads/{ds}_F3_QV.qual.gz’,
        csfasta=’reads/{ds}_F3.csfasta.gz’
    output:
        fastq=’cutreads/{ds}.fastq.gz’,
        log=’cutreads/{ds}.log’
    shell:
        ’cutadapt --bwa -m {CUTADAPT_MINLENGTH} -e {CUTADAPT_ERROR_RATE}\
         -a {ADAPTER} -O {CUTADAPT_OVERLAP} -x {wildcards.ds}: -o {output.fastq}\
         {input.csfasta} {input.qual} > {output.log}’
```

Fig. 3. Part of a `Snakefile` used by `snakemake` describing the rule that cuts adapters from raw reads. There are two input files (the gzipped `.csfasta` and `.qual` files) and two output files (a gzipped `.fastq` file and a `.log` file); their name is determined automatically by inserting proper values for the dataset wildcard {`ds`} when the output files are requested as input files by other rules. When the shell command (a call of the `cutadapt` tool) is executed, the variables (contents of curly brackets) are replaced with actual values of file names and other parameters.

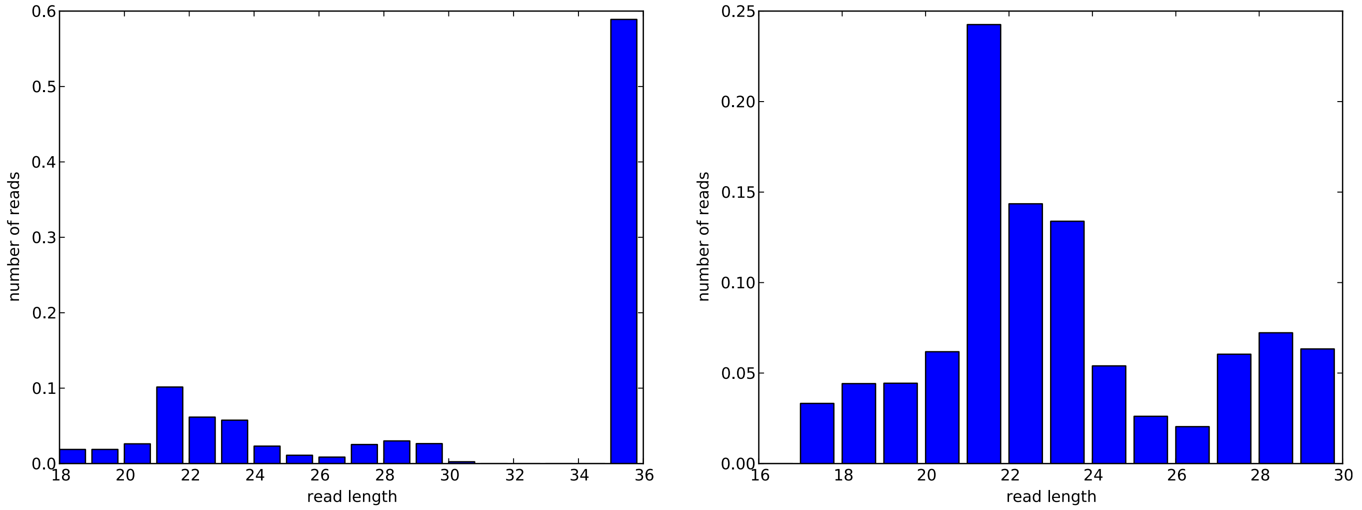

Fig. 4. Distribution of read lengths after adapter removal (*y*-axis shows relative number of reads). Left: length range for mapping against the human genome. Most sequence reads (close to 60%) do not contain the adapter and hence are most likely not mature miRNAs. Right: length range restricted to plausible miRNA lengths (here 17–29 nt).

evidence that miRNAs are present in sufficient quantity in the samples. In our datasets, we found that on average 25% of the reads have a length in this close range after adapter removal (cf. Fig. 4).

If one is exclusively interested in reads that probably contain mature miRNAs after adapter removal, the output of cutadapt can be restricted to those reads that have the desired length of 20–23 nt (19–22 colors) with the `–minimum-length` and `–maximum-length` options.

4.3. Mapping processed reads with BWA

To quantify the expression of short RNA transcripts, two different ways of mapping the processed reads are advisable. They use different references and serve different purposes.

1. For quality control and assessing whether small noncoding RNAs were enriched against mRNA and ribosomal RNA (rRNA), all reads are mapped against the entire reference genome and their locations compared against several RNA annotation tracks.
2. For computing accurate miRNA expression values, the reads are mapped specifically against miRBase.

For both mapping steps, we use BWA [9] version 0.5.9. In more recent versions (0.6.×), mapping of color space reads is disabled.

Before mapping reads, indexes of the respective reference sequences need to be created with `bwa index` or `bwa index -c` for color space. The reads without adapters, which must be stored in FASTQ files, are then mapped with `bwa aln` (again adding `-c` for color space), processed further with `bwa samse`, and converted to BAM format with `samtools` [10]. All three commands can be run at the same time by connecting them through a Unix pipe, which avoids creating unnecessary temporary files and makes better use of multiple CPU cores.

We now describe the two mapping procedures in detail.

4.3.1. Mapping against the human genome

The BWA index is created from the human genome reference assembly GRCh37 (see Section 2). All reads of a minimum length of 17 colors after adapter removal (18 nt, see Section 3.1), which includes those reads that were not trimmed and may therefore contain regular mRNA, are mapped against the reference.

Using `bedtools intersect` from BEDtools [11], each mapped read in the resulting BAM file, including those that do not map uniquely, is annotated with the features of the ENSEMBL genome annotation track. The annotations are grouped into the categories *coding* (annotations ‘protein_coding’, ‘retained_intron’), *miRNA* (annotation ‘miRNA’), and *other non-coding* (other annotations). Both `Snakefile` rules are shown in Fig. 5.

The error tolerance may be increased from the BWA default of two to three errors, but this increases the required CPU time for

```
rule annotate_dataset:
input: track=HGTRACK, bam='maphg/{ds}.bam'
output: 'annotations/{ds}.txt'
shell:
  'bedtools intersect -wa -wb -bed -abam {input.bam} -b {input.track} | \
   cut -f 1-6,14-15 > {output}'

rule rnatypes_of_dataset:
  input: 'annotations/{ds}.txt'
  output: 'annotations/{ds}.rnatypes.txt'
  run:
    reads = defaultdict(set)
    with open(input[0], 'rt') as infile:
      for record in csv.reader(infile, delimiter='\t'):
        read_name = record[3]
        feature_type = record[6]
        reads[read_name].add(feature_type)
    counter = Counter()
    for read_name, feature_types in reads.items():
      if feature_types.intersection(['protein_coding', 'retained_intron']):
        counter['coding'] += 1
      elif feature_types.intersection(['miRNA']):
        counter['mirna'] += 1
      else:
        counter['non-coding'] += 1
    with open(output[0], 'wt') as outfile:
      print(*sorted(counter), sep='\t', file=outfile)
      print(*[counter[key] for key in sorted(counter)], sep='\t', file=outfile)
```

Fig. 5. The first rule calls `bedtools intersect` using the BAM file of a single dataset and the ENSEMBL annotation track to annotate each mapped read with its features. The second rule's **r**un-section is written in Python and classifies each genome-mapped read into one of three types (miRNA, non-coding or coding) and counts their abundance.

mapping tenfold. We observed that although the absolute number of reads mapped to one of the three categories increased by 25%, the ratios did not change. Since we map against the whole human genome for quality control only, the resulting increase in sensitivity may therefore not be required.

We now have two indicators on the amount of miRNAs present in each sample: (1) the fraction of reads of appropriate length after adapter removal (in our experiments, this varied between 22% and 37% of all reads), which is an over-estimation of the reads that will map to miRNAs, and (2) the fraction of reads that map to a 'miRNA'-annotated genomic region (in our experiments, this mainly varied between 1% and 9%), which is an under-estimation. The under-estimation is due in part to the ENSEMBL track version 67 being incomplete regarding miRNAs compared to the most recent miRBase database: it contains 1826 annotated miRNAs, while miRBase contains 1921. Furthermore, miRNA sequences are not unique in the human genome (Section 3.2).

Further possible quality control measures (not used here) may include mapping of trimmed reads against pre-miRNA sequences ('hairpin' sequences) as obtained from miRBase, and mapping against fRNAdb sequences [17]. We expect that experiments based on more recent versions of the small RNA expression kit (SREK) would yield higher results.

4.3.2. Mapping against miRBase

After verifying that miRNAs in the sample are sufficiently enriched, we can proceed by mapping trimmed reads against the human mature miRNA sequences (see Section 2 for the reference sequence file, which is filtered for human sequences). Of the 1921 sequences, 38 sequences are identical to one or more other mature miRNAs in the database. Since these miRNAs cannot be distinguished, we merge them into unique entries, named such that the original miRNA names can be recovered. After merging, there are 1898 entries.

One further change of the reference sequences is recommended when using BWA. BWA does not align reads that extend beyond the reference sequence even when the match is otherwise perfect and the number of additional bases (counted as insertions) would be within the number of allowed errors. Since we want to count such reads, as they probably contain miRNAs with terminal additions, we append five 'N' characters to each mature miRNA reference. BWA replaces each 'N' character with a random base and therefore the overhanging bases will be counted as errors. The number of 'N' characters appended should not be too large in order to limit the bias introduced by random matches into the 'N' region. We found that the number of mapped reads improved up to a number of five 'N' characters, while more than five characters merely increased the amount of random matches. Prepending 'N' characters in addition to appending them did not improve mapping results.

From the adapter-trimmed reads, we use only those that fall within the length range of mature miRNAs in miRBase, which is 16–26 nt. Due to the small size of the reference sequence, the mapping process is very fast and the number of allowed mismatches can be increased to three (parameters `-n 3 -k 3`) with negligible increase in runtime.

In contrast to mapping against the full genome, we recommend a conservative approach of discarding non-unique mappings for further analysis of the resulting BAM files. A read is mapped non-uniquely if it maps to two distinct locations in the target sequences with the same minimum error number. While this systematically underestimates the read counts for those miRNAs that are very similar to other miRNAs, a precise estimate is simply not possible in such cases.

4.3.3. Color space considerations

Reads obtained in dinucleotide color space should be mapped to a color space reference [6]. Mapping in color space bears the advantage of removing technical sequencing errors during conversion from color space to nucleotide space, since two colors need to be changed if one nucleotide is altered. BWA's options take care of the technical details, but with some attention to detail, the mapping sensitivity can be increased, as we describe here.

Each color is the result of interrogating two adjacent bases (a dinucleotide) of the sequenced fragment. The entire fragment contains a 5′ adapter, a sequence of interest (miRNA) and a 3′ adapter. Sequencing starts from the dinucleotide that covers the last base of the 5′ adapter (usually a 'T') and the first base of the miRNA.

The first color therefore contains information about a base that is not part of the actual read. During mapping, this leads to spurious mismatches in the first color. Some read mappers such as BWA therefore require the first color of each read to be removed before mapping.

A similar problem becomes apparent when considering the dinucleotide that covers the transition from the miRNA into the adapter sequence. The corresponding color will also lead to spurious mismatches. During adapter removal, cutadapt takes care of this by removing the adapter and also the color preceding it from the read.

The procedure is illustrated below; colors are represented by numbers between 0 and 3. The 'T' at the start of the read is the last base of the 5′ adapter and not part of the read (but conventionally included in the FASTQ file to allow decoding the read). The colors that need to be removed are underlined.

adapter sequence:			330201030313112312
original read:	T$\underline{3}$	00023210010122222$\underline{3}$	330201030313112
trimmed read:		00023210010122222	

BWA outputs an alignment in nucleotide space by decoding color space reads of length n (which contain information about $n+1$ nucleotides) to the most likely nucleotide sequence, guided by the reference sequence to which the read was mapped. It retains only those nucleotides for which two colors are available, that is, the decoded read has a length of $n-1$.

As a result, the alignments start one base too late and end one base too early. For example, for a miRNA of 20 nt, the alignment would only contain 18 nt. We use the following method in order to get full alignments, and also to avoid discarding colors of the read: we prepend a 'T' (last base of the primer) and append a 'C' (first base of the 3′ adapter) to each reference sequence; as described above, we also append five 'N's. Then we run the adapter removal step again, but we retain the first color and also the color preceding the adapter (the initial 'T' in the read still needs to be removed). This avoids the issue of spurious mismatches and the alignment will cover the entire miRNA since all nucleotides are supported by two colors. The only thing to keep in mind is that the start position of the alignment is off by one in the resulting BAM file.

4.3.4. Computing raw miRNA expressions

Raw absolute expression counts for each dataset are obtained from the BAM files by counting the number of reads mapping to a specific miRNA. Reads that map non-uniquely are discarded. We also discard a read if its reverse complement was mapped instead of the forward read. Reads starting at an offset greater than two within the reference are also discarded.

Each read that is mapped to a specific mature miRNA and not discarded increases the expression count for that miRNA by one. We thus obtain a raw count for each miRNA in each experiment, resulting in a table in which each row represents a miRNA and each column is an experiment or patient.

4.4. Normalization

The raw counts are not comparable across patients, as the absolute number of reads obtained from each experiment varies. Therefore, the raw counts have to be normalized and brought to a common scale. There exist many normalization approaches, e.g., for microarray expression values, or for RNA-seq data. Quantile normalization [18] ensures that the sequence of sorted values (or distribution or histogram) of each experiment agrees with a reference (either one of the datasets or a consensus); it thus entirely modifies each experiment's expression value distribution. Quantile normalization is appropriate when there are many transcripts to consider and the majority of them does not change their expression across experiments for biological reasons, i.e., when most observed variations are due to technical causes. Since there are only a few hundred distinct miRNAs, the underlying assumptions of quantile normalization may be hard to justify.

The least invasive way of normalizing expression data is to rescale each experiment to a common point of reference [19], which could be the mean of all expression values (or, equivalently, the sum), i.e., we could compute a factor such that the sum of expression values is 1,000,000 in each experiment. The disadvantage of mean- or sum-based normalizations is that they are unrobust with respect to a few high expressed miRNAs that dominate the raw counts.

Here we propose two non-invasive yet robust variants. Both variants are part of the pipeline and yield very similar results (cf. Figs. 6 and 7). For each normalization method, we obtain a table of normalized expressions such that each row represents a miRNA and each column represents an experiment or patient. We reduce this table to those miRNAs whose raw (unnormalized) expression values reach at least 5 counts in at least half of the experiments; we simply call them the (somewhere) "expressed" miRNAs. Below this level, no reliable statements are possible from statistical analysis, and such miRNAs are simply referred to as "not expressed". In our experiments, 548 of the 1898 miRNAs (28.9%) were expressed according to this definition.

4.4.1. Quantile-based scaling normalization

The idea is to use a simple scaling normalization but to compute the scaling factor in a robust way. We first pick one experiment as a reference, whose typical counts of expressed miRNAs should be high. We propose to pick the experiment with the highest third quartile (0.75-quantile). When scaling another experiment to the reference, we compute the scaling factor as follows. We sort the expression values of both experiments individually to obtain all quantiles. We now consider those quantiles which reach or exceed 20 raw counts in both experiments and compute the ratios between them. We pick the median of the ratios as the scaling factor.

4.4.2. Capped quantile normalization

In principle, we perform a standard quantile normalization, but we leave out extreme values, which are normalized by an

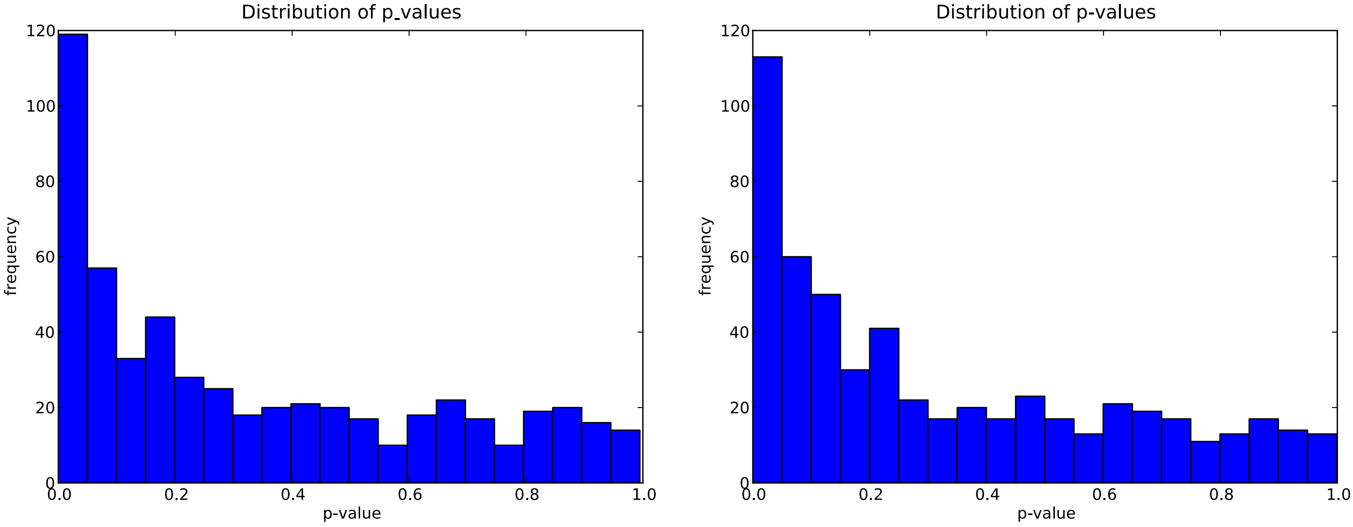

Fig. 6. Distribution of *t*-test *p*-values using quantile-based scaling normalization (left) and capped quantile normalization (right). The leftmost bar represents uncorrected *p*-values $\leqslant 0.05$. In a completely randomized experiment, we expect a uniform distribution (27.4 miRNAs on average).

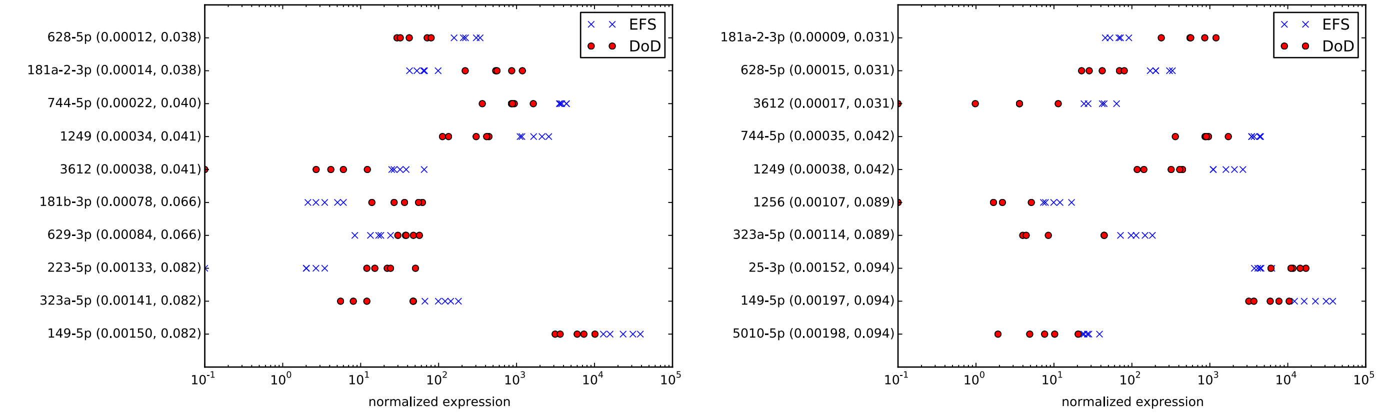

Fig. 7. The ten most significantly differentially expressed miRNAs after quantile-based scaling normalization (left) and capped quantile normalization (right). Blue crosses: expression in favorable neuroblastoma samples (event-free survival, EFS); red circles: dito in unfavorable neuroblastoma (died of disease; DoD). Next to the miRNA name, the uncorrected *p*-value and the FDR are shown.

appropriate scaling factor. In more detail, we first perform the quantile-based scaling normalization as described above to bring all expression values to a common scale and work on this pre-normalized set. Then we define the *i*-th reference quantile as the mean of the *i*-th quantiles of all experiments. However, if the standard deviation of the *i*-th quantiles exceeds a given value (here 25), we consider this quantile and all higher quantiles to be “extreme”. For the extreme quantiles, a new scaling factor is computed as the median of the ratios between high-expressed (but not extreme) quantiles. The non-extreme quantiles are set to the reference quantiles, as for standard quantile normalization.

4.5. Analysis of differential miRNA expression

The expressed miRNAs are tested for differential expression among the two classes (here, favorable vs. unfavorable neuroblastoma, i.e., event-free survival vs. died of disease). Several statistics are computed for each miRNA: (1) a regularized log fold-change value, (2) a *p*-value from a two-sided *t*-test for different mean of regularized log-expression values, and (3) a false discovery rate (FDR).

The main difficulties when testing for differential expression stem from low expressed miRNAs. We partially alleviate this problem by considering only miRNAs that are expressed with a raw count of 5 in at least 5 experiments. Nevertheless, consider the following hypothetical case of a miRNA that is expressed 0–5 times in one class with an average of 3, and 5–10 times in the other class with an average of 6. Numerically, this may well test as significant differential expression with a fold-change factor of 2. However, the absolute counts are so low that this statement is unreliable; small differences in read mapping may lead to very different values and results. Therefore we consistently add 20 pseudocounts to each expression value before any test or computation. We choose 20 pseudocounts with the following rationale: if zero counts are observed but subsequently change to a single one for any reason (e.g., due to a slightly different quality filter), we would like to limit the influence on the tests. With a baseline of 20 pseudocounts, the effect of the additional read is 5%, whereas it would be 10% with 10 pseudocounts or even 100% without pseudocounts. The pseudocounts have the effect of equalizing the perceived expression of low expressed miRNAs in both classes; e.g., in the above example the ratio would be 26/23 instead of 6/3. High expressed miRNAs are barely affected.

The regularized mean log expression of miRNA i in class $k \in \{1,2\}$ is

$$\mu_{i,k} := \frac{1}{|\mathrm{Class}k|} \cdot \sum_{j \in \mathrm{Class}k} \log(x_{i,j} + \rho),$$

where $\rho = 20$ is the number of pseudocounts and $x_{i,j}$ is the (normalized) expression value of miRNA i in experiment j. The regularized log fold-change estimate between the classes is

$$\mathrm{lfc}_i := \mu_{i,1} - \mu_{i,2}.$$

The *t*-test for differential expression uses the $\log(x_{i,j} + \rho)$ values of the two classes and hence tests whether $\mu_{i,1} = \mu_{i,2}$.

When performing many tests, a fraction of p features will have a p-value $\leqslant p$ just by chance, i.e., here we expect about 27 miRNAs with a p-value $\leqslant 0.05$ by chance. When plotting a histogram of all resulting p-values, a uniform distribution on the interval $[0, 1]$ is expected if no signal is present. Here, however, we observe a strong bias towards small *p*-values (Fig. 6), independently of the normalization method, which is a global indicator for differential expression without nominating specific miRNAs yet.

To single out significantly differentially expressed miRNAs, the *p*-values have to be corrected for multiple testing. One option is to compute the false discovery rate (FDR) according to Benjamini–Hochberg [20,21]. When considering all miRNAs with an FDR below 0.05, we may expect 5% of the reported miRNAs are falsely discovered as differentially expressed. In our experiments, the reported set of miRNAs is independent of the normalization method and consists of the following hsa-miRs: 181a-2-3p, 628-5p, 3612, 744-5p, 1249. All of these are good *biomarker candidates* for differentiating between the two classes. The ten most significant miRNAs, together with their uncorrected *p*-values and their FDRs, are shown in Fig. 7.

4.6. Validation by RT-qPCR

While the sequencing-based approach provides biomarker candidates, they have to be validated in two different ways to be confirmed as biomarkers.

The first validation is technical and consists of measuring miRNA expression on the *same samples* with an independent technique, for which RT-qPCR is the obvious choice. A multiplex stem–loop RT-qPCR procotol suitable for miRNA expression measurement [22] and corresponding normalization techniques [23] have been described previously. Reported are normalized quantification cycle numbers (ΔC_q), which is the (fractional) PCR cycle number in which a certain quantity of the desired target is first reached in relation to a baseline, which is used for normalization. Since PCR amplifies the molecules exponentially, we expect that $-\Delta C_q$ values correlate linearly with logarithmic normalized expression values from high-throughput sequencing (HTS) experiments.

For the comparison, the following technical challenge arises: depending on the RT-qPCR kit, its miRNA names may differ from those in the current miRBase version. We identified miRBase release 8.1 as the relevant one for the used PCR kit in our experiments, whereas the current version is release 18. In the meantime, miRBase has dropped miR/miR* naming and now consistently uses −5p/−3p naming. To compare sequencing-based expression levels with RT-qPCR $-\Delta C_q$ values, corresponding names have to be mapped to each other. For this task, we advocate a sequence-based approach instead of a name-based approach, as it is the sequence that defines the miRNA. However, several sequences have also changed between miRBase releases (they gained, lost, or were shifted by one or a few nucleotides) without being assigned a new name.

We thus implemented the following comparison method. For both miRBase versions, a bi-directional mapping between names and sequences is created; however, each name is not only associated with the exact sequence, but also with the sequence without its first or last character. Given a previous miRNA name, the three associated sequences are retrieved; for each sequence, we check if any new names exist and collect these. Either the unique or most common new name is chosen. This procedure may be unsuccessful for two reasons: the sequence has diverged too much to be found in the newer release, or there may be two good candidate names without a clear preference. In those cases, we deem it appropriate that no explicit connection is made. This procedure uniquely identifies the current names for 404 of the 455 miRNAs of miRBase 8.1; for 48 miRNAs, the sequence has changed considerably, and for three miRNAs, the name assignment would be ambiguous.

Logarithmic expression and $-\Delta C_q$ values are compared by scatterplots for each experiment and summarized by Pearson correlation coefficients (Fig. 8 left). On our datasets, we found experiment-wise correlation coefficients between 0.655 and 0.732, independently of the normalization method. Additionally, we compute a correlation coefficient for each expressed miRNA across all experiments. This makes sense only for a sufficient number of experiments; $5 + 5 = 10$ in our dataset is certainly the lower limit. Looking at the resulting histogram of miRNA correlation coefficients (Fig. 8 right), we observe a good agreement between the two methods for most miRNAs, with some notable exceptions: for the capped quantile normalization, the 11 anti-correlated miRNAs (correlation $\leqslant 0.0$) are 148b-3p, 151a-3p, 186–5p, 188–5p, 23a-3p, 30b-5p, 361–5p, 378a-5p, 517[ab]-3p, 617, 627. For quantile-based scaling normalization, 589–3p is additionally anti-correlated. Specific reasons for non-correlation or even anti-correlation between HTS and PCR are unknown to us. Contrary to our expectations, we did not find low correlation coefficients for those miRNAs for with highly similar other miRNAs exist (cf. Section 3.3). The correlation coefficients also appear to be mostly independent of a miRNA's overall expression level or differential expression (data not shown).

The second validation of biomarker candidates involves establishing their biological predictiveness on *independent samples*. Ideally, their expression is measured by specific RT-qPCR reactions on a large collective of patients. We re-analyzed an independent cohort of 69 primary neuroblastomas [24], which were profiled using the Megaplex RT stem–loop primer pool (Applied Biosystems, Foster City, CA, USA) with a two-step amplification protocol [22] and a corresponding normalization technique [23].

The assay is limited to the detection of miRNAs known and specified at design time. Therefore, we were not able to assess each biomarker candidate identified by sequencing with this existing assay. The results are shown in Table 1, where the false discovery rate (FDR) is limited by 0.10, i.e., 10% of the candidates can be expected to be falsely included. All assessable candidates (hsa-miR-149-5p, hsa-miR-331-3p, hsa-miR-181a-5p, hsa-miR-654-5p and hsa-miR-25-3p) were confirmed by the RT-qPCR approach on the independent test set. Additionally, for these same candidates the correlation between sequencing-based logarithmic HTS expression and PCR-based $-\Delta C_q$ values on the 10 profiled datasets was high. This finding underlines the robustness of our sequencing results and suggests that most of the remaining candidates warrant further study.

We additionally investigated the differential expression of miRNAs previously described as associated with neuroblastoma outcome and for which RT-qPCR data on the test cohort was available. The findings are summarized in Table 2. It should be noted that the shown RT-qPCR false discovery rate (PCR-FDR) is based on 69 samples, whereas the sequencing-based FDR (HTS-FDR) is derived from only 10 samples, and thus has less power to

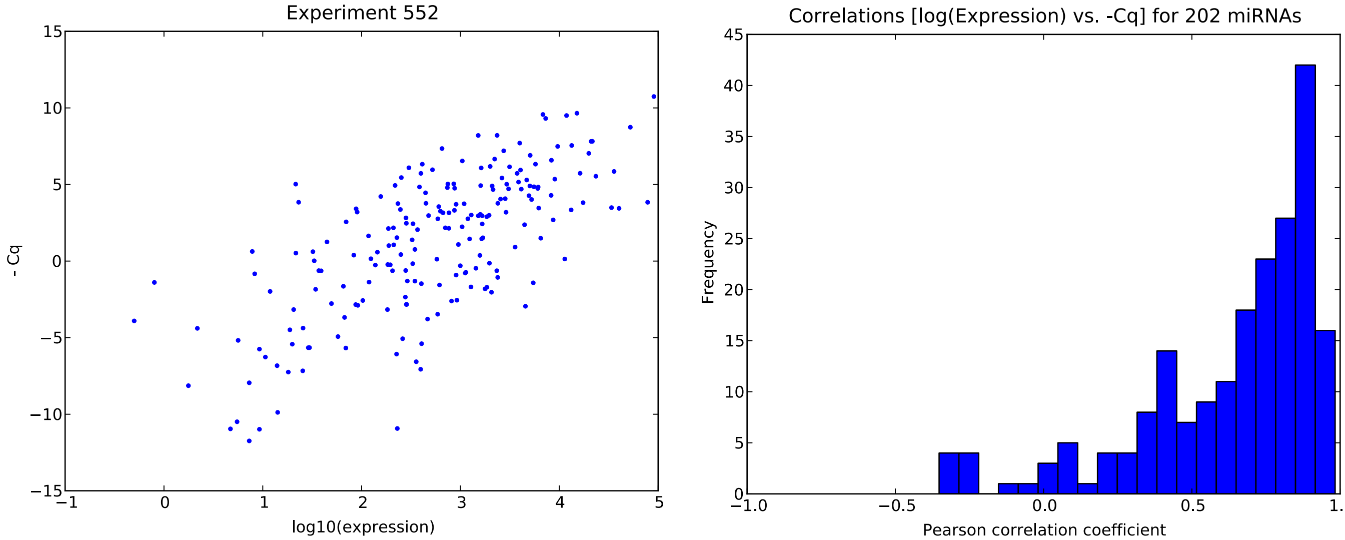

Fig. 8. Left: scatter plot of logarithmic HTS miRNA expression values (x-axis) against negative ΔC_q values (RT-qPCR, y-axis) on dataset 552 after capped quantile normalization with a Pearson correlation coefficient of 0.695. Each dot represents one of 202 expressed miRNAs. The other datasets look similar with correlation coefficients between 0.655 and 0.732, independently of the normalization method. Right: histogram of correlation coefficients for each of 202 miRNAs across 10 experiments after capped quantile normalization.

Table 1
Potential biomarker candidates for discriminating between prognostically favorable vs. unfavorable neuroblastomas, discovered with an FDR $\leqslant 0.10$. HTS-FDR: false discovery rate from 5 + 5 HTS datasets, using the worse of the two FDRs from the two normalization methods; PCR-FDR: false discovery rate from an independent cohort of 69 patients using RT-qPCR, visualized using $***$ for FDR $\leqslant$0.001, $**$ for FDR $\leqslant$0.01, $*$ for FDR $\leqslant$0.05, and ? if FDR not available (N/A); Corr.: miRNA-specific correlation coefficient between log (expression) and $-\Delta C_q$ values on the same 5 + 5 patient datasets.

miRNA	HTS-FDR	PCR-FDR		Corr.
hsa-miR-181a-2-3p	0.038	N/A	?	N/A
hsa-miR-628-5p	0.038	N/A	?	N/A
hsa-miR-3612	0.041	N/A	?	N/A
hsa-miR-1249	0.042	N/A	?	N/A
hsa-miR-744-5p	0.042	N/A	?	N/A
hsa-miR-323a-5p	0.089	N/A	?	N/A
hsa-miR-149-5p	0.094	0.001	***	0.904
hsa-miR-331-3p	0.094	0.001	***	0.916
hsa-miR-181a-5p	0.094	0.005	**	0.972
hsa-miR-654-5p	0.094	0.005	**	0.775
hsa-miR-25-3p	0.094	0.036	*	0.678
hsa-miR-431-3p	0.094	N/A	?	N/A
hsa-miR-5010-5p	0.094	N/A	?	N/A
hsa-miR-3605-3p	0.096	N/A	?	N/A

Table 2
FDRs of miRNAs previously reported as potentially discriminating between prognostically favorable vs. unfavorable neuroblastoma courses. HTS-FDR: false discovery rate from 5 + 5 sequencing datasets, using the worse of the two FDRs from the two normalization methods, visualized using $***$ for FDR $\leqslant$0.001, $**$ for FDR $\leqslant$0.01, $*$ for FDR $\leqslant$0.05, o for FDR $\leqslant 1/3$, empty otherwise; PCR-FDR: false discovery rate from independent cohort of 69 patients using RT-qPCR, visualized similarly; Corr.: miRNA-specific correlation coefficient between log (expression) values and $-\Delta C_q$ values on the same 5 + 5 patients.

miRNA	HTS-FDR		PCR-FDR		Corr.
hsa-miR-17-5p	0.193	o	0.004	**	0.883
hsa-miR-18a-5p	0.732		0.023	*	0.537
hsa-miR-19a-3p	0.390		0.017	*	0.852
hsa-miR-19b-3p	0.314	o	0.204	o	0.946
hsa-miR-20a-5p	0.412		0.008	**	0.731
hsa-miR-25-3p	0.094	o	0.036	*	0.678
hsa-miR-34a-5p	0.271	o	0.532		0.925
hsa-miR-34c-5p	0.507		0.230	o	0.734
hsa-miR-92a-3p	0.315	o	0.002	**	0.946
hsa-miR-125a-5p	0.823		0.364		0.755
hsa-miR-125b-5p	0.603		0.501		0.773
hsa-miR-149-5p	0.094	o	0.001	***	0.904
hsa-miR-181a-5p	0.094	o	0.005	**	0.972
hsa-miR-181b-5p	0.232	o	0.676		0.790
hsa-miR-181c-5p	0.944		0.559		0.398
hsa-miR-190a	0.236	o	0.049	*	0.764
hsa-miR-199a-5p	0.152	o	0.293	o	0.843
hsa-miR-199b-5p	0.197	o	0.129	o	0.888
hsa-miR-323a-3p	0.530		0.040	*	0.844
hsa-miR-324-5p	0.133	o	0.008	**	0.918
hsa-miR-542-5p	0.152	o	0.005	**	0.802
hsa-miR-628-3p	0.271	o	0.006	**	0.518
hsa-miR-654-5p	0.094	o	0.005	**	0.775
hsa-let-7a-5p	0.458		0.137	o	0.903
hsa-let-7b-5p	0.304	o	0.312	o	0.887
hsa-let-7c	0.358		0.046	*	0.852
hsa-let-7d-5p	0.782		0.890		0.525
hsa-let-7e-5p	0.986		0.847		0.419
hsa-let-7f-5p	0.707		0.194	o	0.633
#Samples	10		69		10

detect differential expression. Most miRNAs detected by RT-qPCR, marked ‘$*$’ or ‘$**$’, are considered as weak candidates by the sequencing approach, marked ‘o’. One could speculate that sequencing more patients could yield results comparable to PCR.

5. Discussion and conclusion

As discussed in Section 3, there are several challenges present in the typical short reads of miRNA datasets that distinguish their analysis from that of typical RNA-seq experiments. Attention to detail in every analysis step, such as embedding the mature miRNA reference sequences into their adapter context when mapping in color space, or robust non-invasive normalization methods are crucial for obtaining accurate expression level estimates.

We provide an optimized automated pipeline, as described in Section 4 for miRNA expression estimation and differential expression analysis in order to discover putative biomarkers for distinguishing favorable from non-favorable neuroblastoma tumors. The pipeline is complex, but consists of modules of rules that can be modified, left out or expanded with additional rules to customize the workflow for different environments or slightly different tasks. Visualization as a directed acyclic graph highlights the dependencies between computational steps (see electronic supplement). The pipeline itself is available as a `Snakefile` in the electronic supplement. Sample names and corresponding file names and paths have to be reconfigured for different datasets and on different systems.

The pipeline is optimized for the analysis of miRNA reads in dinucleotide color space; this appears to be one of the main applications of the SOLiD sequencing system. If the sequencing data is directly obtained in nucleotide space (e.g., in FASTQ format), the complications related to color-space mapping disappear, but those related to short read length remain. We have not found the similarity of single miRNA pairs (Section 3.3) to be a problem.

We believe that it is reassuring (and worth verifying) that different normalization methods (one pure scaling method and a more invasive quantile-based method) give similar results. If this were not the case, the resulting biomarker candidates should probably be discarded, as they would be unrobust with respect to the normalization method. The named wildcards in the `Snakefile` make it easy to execute the same pipeline in parallel for any number of normalization methods and to compare the results.

Comparing the respective strengths of both high-throughput sequencing (HTS) and RT-qPCR, we would like to point out that HTS analyses can be re-applied even to older datasets whenever the objects of interests have been re-defined. As miRBase has been updated several times since previous studies, we were able to propose novel biomarker candidates. However, due to high costs, miRNA analysis by HTS is limited to few datasets, sacrificing power to detect differential expression. RT-qPCR is limited to existing assays (here, we could not obtain ΔC_q values for many of the newer miRBase entries), but it can be applied in a targeted way cost-effectively to more samples than the HTS strategy, yielding higher detection power.

It is further advisable to combine the respective strengths of HTS for unbiased candidate generation and potentially discovery of yet unknown miRNAs ([4]; not discussed here) and of RT-qPCR to examine generated candidates in a larger number of biological samples for verification or rejection.

Acknowledgments

S.R., J.K., A.S. gratefully acknowledge funding by the DFG Collaborative Research Center (Sonderforschungsbereich, SFB) 876 “Providing Information by Resource-Constrained Data Analysis” within projects TB1 and C1 (http://sfb876.tu-dortmund.de).

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.ymeth.2012.10.005.

References

[1] J. Winter, S. Jung, S. Keller, R.I. Gregory, S. Diederichs, Nat. Cell Biol. 11 (3) (2009) 228–234.

[2] D.P. Bartel, Cell 116 (2) (2004) 281–297.

[3] R. Garzon, G.A. Calin, C.M. Croce, Annu. Rev. Med. 60 (2009) 167–179.

[4] J.H. Schulte, T. Marschall, M. Martin, P. Rosenstiel, P. Mestdagh, S. Schlierf, T. Thor, J. Vandesompele, A. Eggert, S. Schreiber, S. Rahmann, A. Schramm, Nucleic Acids Res. 38 (17) (2010) 5919–5928.

[5] R. Leinonen, H. Sugawara, M. Shumway, Nucleic Acids Res. 39 (database issue) (2011) D19–D21.

[6] H. Breu, A Theoretical Understanding of 2 Base Color Codes and Its Application to Annotation, Error Detection, and Error Correction, Tech. Rep., Applied Biosystems by Life Technologies Corporation, White Paper, 2010.

[7] J. Köster, S. Rahmann, Bioinformatics 28 (19) (2012) 2520–2522.

[8] M. Martin, EMBnet.journal 17 (1) (2011) 10–12.

[9] H. Li, R. Durbin, Bioinformatics 25 (14) (2009) 1754–1760.

[10] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, Bioinformatics 25 (16) (2009) 2078–2079.

[11] A.R. Quinlan, I.M. Hall, Bioinformatics 26 (6) (2010) 841–842.

[12] D.M. Church, V.A. Schneider, T. Graves, K. Auger, F. Cunningham, N. Bouk, H.-C. Chen, R. Agarwala, W.M. McLaren, G.R.S. Ritchie, D. Albracht, M. Kremitzki, S. Rock, H. Kotkiewicz, C. Kremitzki, A. Wollam, L. Trani, L. Fulton, R. Fulton, L. Matthews, S. Whitehead, W. Chow, J. Torrance, M. Dunn, G. Harden, G. Threadgold, J. Wood, J. Collins, P. Heath, G. Griffiths, S. Pelan, D. Grafham, E.E. Eichler, G. Weinstock, E.R. Mardis, R.K. Wilson, K. Howe, P. Flicek, T. Hubbard, PLoS Biol. 9 (7) (2011) e1001091.

[13] A. Kozomara, S. Griffiths-Jones, Nucleic Acids Res. 39 (database issue) (2011) D152–D157.

[14] M.I. Abouelhoda, S. Kurtz, E. Ohlebusch, J. Discrete Algorithms 2 (1) (2004) 53–86.

[15] T. Marschall, M. Martin, S. Rahmann, A BWT-based suffix array construction, in: Biological Sequence Analysis Using the SeqAn C++ Library, CRC Press, 2009, pp. 261–282 (Chapter 16).

[16] M.J.L. de Hoon, R.J. Taft, T. Hashimoto, M. Kanamori-Katayama, H. Kawaji, M. Kawano, M. Kishima, T. Lassmann, G.J. Faulkner, J.S. Mattick, C.O. Daub, P. Carninci, J. Kawai, H. Suzuki, Y. Hayashizaki, Genome Res. 20 (2) (2010) 257–264.

[17] T. Mituyama, K. Yamada, E. Hattori, H. Okida, Y. Ono, G. Terai, A. Yoshizawa, T. Komori, K. Asai, Nucleic Acids Res. 37 (database issue) (2009) D89–D92.

[18] B.M. Bolstad, R.A. Irizarry, M. Astrand, T.P. Speed, Bioinformatics 19 (2) (2003) 185–193.

[19] M.D. Robinson, A. Oshlack, Genome Biol. 11 (3) (2010) R25.

[20] Y. Benjamini, Y. Hochberg, J. R. Stat. Soc. B 57 (1) (1995) 289–300.

[21] J.D. Storey, J. R. Stat. Soc. B 64 (3) (2002) 479–498.

[22] P. Mestdagh, T. Feys, N. Bernard, S. Guenther, C. Chen, F. Speleman, J. Vandesompele, Nucleic Acids Res. 36 (21) (2008) e143.

[23] P. Mestdagh, P.V. Vlierberghe, A.D. Weer, D. Muth, F. Westermann, F. Speleman, J. Vandesompele, Genome Biol. 10 (6) (2009) R64.

[24] J.H. Schulte, B. Schowe, P. Mestdagh, L. Kaderali, P. Kalaghatgi, S. Schlierf, J. Vermeulen, B. Brockmeyer, K. Pajtler, T. Thor, K. de Preter, F. Speleman, K. Morik, A. Eggert, J. Vandesompele, A. Schramm, Int. J. Cancer 127 (10) (2010) 2374–2385.