Bioinformatics, 36(6), 2020, 1933–1936
doi: 10.1093/bioinformatics/btz806
Advance Access Publication Date: 4 November 2019
Applications Note

Sequence analysis

Jasmine: a Java pipeline for isomiR characterization in miRNA-Seq data

Xiangfu Zhong [1,2], Albert Pla[1,2] and Simon Rayner[1,2,3,*]

[1]Department of Medical Genetics, University of Oslo, Oslo, [2]Department of Medical Genetics, Institute of Clinical Medicine, Oslo University Hospital, Oslo and [3]Hybrid Technology Hub – Centre of Excellence, Institute of Basic Medical Sciences, University of Oslo, Oslo, Norway

*To whom correspondence should be addressed.

Associate Editor: Yann Ponty

Received on March 7, 2019; revised on September 13, 2019; editorial decision on October 26, 2019; accepted on October 30, 2019

Abstract

Motivation: The existence of complex subpopulations of miRNA isoforms, or isomiRs, is well established. While many tools exist for investigating isomiR populations, they differ in how they characterize an isomiR, making it difficult to compare results across different tools. Thus, there is a need for a more comprehensive and systematic standard for defining isomiRs. Such a standard would allow investigation of isomiR population structure in progressively more refined sub-populations, permitting the identification of more subtle changes between conditions and leading to an improved understanding of the processes that generate these differences.

Results: We developed Jasmine, a software tool that incorporates a hierarchal framework for characterizing isomiR populations. Jasmine is a Java application that can process raw read data in fastq/fasta format, or mapped reads in SAM format to produce a detailed characterization of isomiR populations. Thus, Jasmine can reveal structure not apparent in a standard miRNA-Seq analysis pipeline.

Availability and implementation: Jasmine is implemented in Java and R and freely available at bitbucket https://bitbucket.org/bipous/jasmine/src/master/.

Contact: simon.rayner@medisin.uio.no

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

MicroRNAs (miRNA) are a class of small non-coding RNAs, which have significant function in gene regulation (Bartel, 2018). Nowadays, high-throughput sequencing (HTS) is the most common platform for investigating changes in miRNA populations between conditions (in the form of read data). However, HTS has revealed the presence of 'miRNA-like' reads, which differ from known miRNAs by one or more nucleotides. These miRNA-like isoforms (or isomiRs), were first proposed by Morin *et al.* (2008) and their functional significance is well established (see Neilsen *et al.*, 2012). Many software tools have been developed for isomiRs profiling such as isomiR-SEA (Urgese *et al.*, 2016) and Chimira (Vitsios and Enright, 2015)—a complete list of available tools is listed in Supplementary Table S2—and a recent study compared the performance of seven popular isomiR identification tools (Amsel *et al.*, 2017). However, most tools only report isomiR reads, rather than classifying them in terms of the type of modification. Consequently, there is no easy way to profile the population either within, or across datasets, and it is not possible to relate studies that have used different analysis tools as many of them have their own embedded qualification criteria.

A further consideration is that miRBase (Kozomara *et al.*, 2018), the standard miRNA reference database, catalogs miRNAs as well defined entities with a distinct start and stop positions producing a single sequence. Also, many entries differ from other entries by a single nucleotide. Finally, the authors caution that they provide 'minimal gate-keeping' for annotation and entries may be removed in subsequent releases. All of these points can confound analysis of miRNA datasets and need to be considered when characterizing isomiRs in NGS datasets. Thus, to identify miRNAs or isomiRs, a curated miRNA reference annotation and well-defined standard is needed for systematic isomiR annotation.

Here, we propose a comprehensive isomiR nomenclature and implement it in Jasmine, a Java based pipeline that can incorporate curated miRBase annotation (Zhong *et al.*, 2019) and perform isomiR analysis on miRNA-Seq data.

2 Usage notes

Jasmine accepts unmapped (fastq/fasta format) or mapped (SAM format) reads as input. *Mapping*: (i) As a starting point it is assumed

that all short miRNA-like reads are potential products of miRNA biogenesis. (ii) Jasmine uses curated miRBase entries as mapping references, rather than genome or original miRBase entries, to reduce multiple and ambiguous mappings, so it can be applied to any species annotated in miRBase. (iii) Reads are collapsed into unique sequences and pre-miRNAs are used as mapping reference. Thus, Jasmine can run on PC with limited CPUs and memory. *IsomiR analysis*: Jasmine incorporates a hierarchal isomiR nomenclature (Fig. 1), allowing progressively more refined characterization of isomiR populations in HTS studies.

3 Materials and methods

Jasmine is written in Java, implementing commonly used software tools to aid pre-analysis and processing (e.g. *bowtie* for mapping and *Trimmomatic* for adapter trimming—see Supplementary Table S1 for details). Jasmine is split into four analysis modules for flexible and rapid analysis of multiple samples or projects (see work flow in Supplementary Fig. S1).

3.1 Input

Jasmine accepts three types of files as input: (i) raw read data in fastq/fasta format; (ii) adapter trimmed reads in fastq or fasta; (iii) collapsed reads in fasta format. Additionally, a configuration file in XML format is also required to launch the pipeline.

3.2 Analysis modules

The four analysis modules in Jasmine are: *pre-processing*, *mapping*, *parsing* and *post-analysis*.

Pre-processing: two primary sub steps are executed: (i) Adapter trimming via *Trimmomatic* (Bolger *et al.*, 2014) or *cutadapt* (Martin, 2011) to implement adapter trimming. (ii) Collapsing trimmed reads into unique reads using *fastx_collapser* and *fastq_to_fasta* from the *FASTX-Toolkit*. FastQC and MultiQC (Ewels *et al.*, 2016) provide QC reports on trimmed reads. A summary report is generated by Jasmine for the step.

Mapping: The default built-in mapping approach uses *bowtie* (Langmead *et al.*, 2009) to map reads against curated miRNA precursor sequences but users may define other tools for the mapping step if required. Since Jasmine focuses on isomiR analysis, at least one mismatch is recommended.

isomiR parsing: The core module of Jasmine, it parses mapped reads (in SAM format) to identify isomiRs, assign them a proper isomiR name, and generate a detailed report. Reads are classified into two groups; (i) reads mapping to a unique location; (ii) reads mapping to multiple mapped locations within the provided mapping reference.

Post analysis: Provides a summary report using our isomiR nomenclature, including features such as polymorphism and templated extension and count tables for differential expression analysis.

3.3 Output

The output files comprise a miRNA count table, an isomiR count table, a polymorphism table, a templated extension count profile report and an isomiR report file. The latter provides details on isomiR reads, pre-miRNA reference, modifications and assigned isomiR name.

The count tables generated by Jasmine are suffixed with 'Level' according to following classification:

Level 0: only mature miRNAs from known annotations.

Level 1: (i) mature miRNAs; (ii) all different isoforms grouped together as a single isomiR group 'isomiR'.

Level 2: (i) mature miRNAs; (ii) isomiRs further sub-classed by lengths: *same length-*, *longer-* and *shorter-* than reference miRNA.

Level 3: mature miRNAs and isomiRs sub-classed according to the 30 groups shown in Figure 1a.

Level 4: lowest level of isomiR description. Each isomiR is reported separately with all the information needed to reconstruct the sequence from the reference miRBase entry. For polymorphisms, whether the change occurs in the seed region and whether it corresponds to known single-nucleotide variants (SNVs) is also reported.

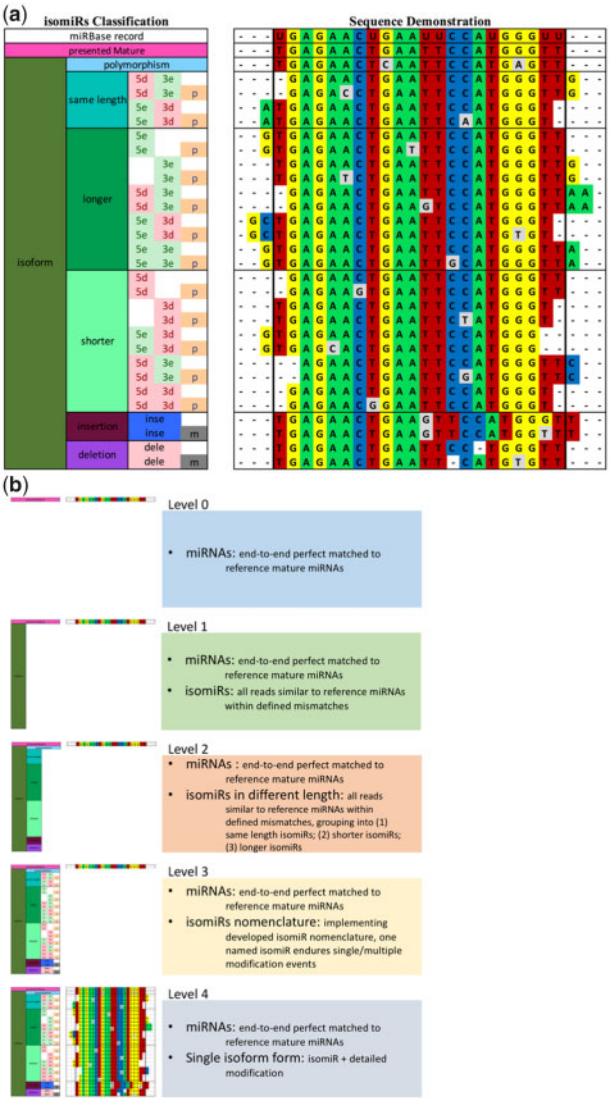

Fig. 1. Schematic of isomiR nomenclature used in Jasmine. (**a**) Left hand rectangle. isomiRs are classified into progressively more refined levels. From left to right. Level 1: miRNAs are classified into mature miRNAs (identical start and end position, and read sequence as found in miRBase) and isoforms; Level 2: Isoforms are sub-classified as 'polymorphic' (poly), 'same length' (sl), 'longer' (lr) and 'shorter' (sr) isomiRs, where Same length isomiRs have the same length as parent mature miRNAs, but are shifted in the 5′ or 3′ direction. Level 3: IsomiRs are further categorized in terms of position and modification type; i.e. deletion, extension and substitution. 5 and 3: 5 and 3 prime-end respectively. e: extension; d: deletion; p: polymorphism in region overlapping mature miRNA; m: mixture. Right hand rectangle shows specific example for each case. (**b**) Summary of the different isomiR classification levels used in the analysis of the (Nogales-Cadenas, 2016) dataset. Jasmine generates count table in five different tab-separated files in tsv format, 'Level 0', 'Level 1', 'Level 2', 'Level 3' and 'Level 4'. See Supplementary Table S4 for detailed explanations of the abbreviations used for each isomiR sub-classes. In Level 4, templated and non-templated information is provided, as well as matched SNVs. 'N > M' is used to describe nucleotide modifications in Level 4, where N = M means templated, N! = M means non-templated and N > D indicates deletion. Nucleotide modifications occurring in the seed region (2–8nt), are be tagged with 'seed' flag

Level 5: This is an additional level which summarizes the information in Level 4 according to the grouping: '*5′-isomiR*', '*3′isomiR*', '*sub.isomiR*' and '*mix.isomiR*'.

4 Using the Jasmine pipeline

4.1 Using Jasmine

The Jasmine pipeline is available as complied jar file or compiling from source. both are available via Bitbucket. An example XML

configuration file and user manual are also provided. Jasmine can be run from the terminal using the command:

```
> java -jar Jasmine.jar jasmine_config.xml
```

where `jasmine_config.xml` is the configuration file for analysis.

4.2 Benefits and drawbacks of using Jasmine

The Jasmine pipeline implements the isomiR nomenclature described in Figure 1. The nomenclature in Jasmine is more comprehensive and flexible for describing isomiR populations in sequencing datasets, compared to other tools, see Table 1, Figure 1b and Supplementary Table S2. Users can select which level is suitable for their specific study.

Jasmines addresses multiple mapping issues by: (i) mapping reads against miRNA precursor sequences; (ii) merging miRNA annotation entries with identical sequences and (iii) calculating sequence similarity before isomiR profiling.

A potential drawback of mapping to precursor sequences is that reads from other genomic regions which are not annotated as miRNAs will not be mapped. However, this can be addressed by excluding miRNAs which can map to other regions. For example, miRNA sequences that map to tRNAs. Also, Jasmine seeks describes isomiR populations in the data, it doesn't attempt to identify 'true isomiRs' However, users can define a read count cut-off to filter noise.

Table 1. Comparison of isomiR nomenclature between Jasmine and other selected tools

Demo sequence	Methods	isomiR nomenclature
UAGCUUAUCAGAC UGAUGUUGA	Jasmine	hsa-miR-21-5p: mature
	mirAligner	hsa-miR-21-5p.ref
	Binarized isomiR	hsa-miR-21-5p|0|0
	isomiR-SEA	hsa-miR-21-5p mirna_exact
	miRNA-MATE	hsa-miR-21-5p canonical miRNA
UGGCUUAUCAGAC UGAUGUUGA	Jasmine	hsa-miR-21-5p: poly
	mirAligner	& hsa-miR-21-5p.mis
	Binarized isomiR	hsa-miR-21-5p|0|0
	isomiR-SEA	not as isomiR
	miRNA-MATE	hsa-miR-21-5p substitution only isomiR
GUAGCUUAUCAGA CUGAUGUUGA	Jasmine	hsa-miR-21-5p: lr-5e
	mirAligner	not as isomiR
	Binarized isomiR	hsa-miR-21-5p|+1|0
	isomiR-SEA	hsa-miR-21-5p iso_5p-only
	miRNA-MATE	hsa-miR-21-5p start-site only isomiR
UAGCUUAUCAGAC UGAUGUUGA	Jasmine	hsa-miR-21-5p: lr-3e
	mirAligner	hsa-miR-21-5p.ad.C
	Binarized isomiR	hsa-miR-21-5p|0|+1
	isomiR-SEA	hsa-miR-21-5p iso_3p-only
	miRNA-MATE	hsa-miR-21-5p end-site only isomiR
UAGCUUAUCAGAC UGAUGUUG	Jasmine	hsa-miR-21-5p: sr-3d
	mirAligner	hsa-miR-21-5p.t3.a
	Binarized isomiR	hsa-miR-21-5p|0|-1
	isomiR-SEA	hsa-miR-21-5p iso_3p-only
	miRNA-MATE	hsa-miR-21-5p end-site only isomiR
UAGCUUAUCAGAC UGAUGUU	Jasmine	hsa-miR-21-5p: sr-3d
	mirAligner	hsa-miR-21-5p.t3.ga
	Binarized isomiR	hsa-miR-21-5p|0|-2
	isomiR-SEA	hsa-miR-21-5p iso_3p-only
	miRNA-MATE	hsa-miR-21-5p end-site only isomiR

4.3 Application of Jasmine to miRNA-seq data

To demonstrate the usage of Jasmine, we apply it on miRNA-seq data generated by Nogales-Cadenas et al. (Nogales-Cadenas *et al.*, 2016). Detailed sample information and instructions are provided in Supplementary Table S3 and repository.

Jasmine generates a series of count table files, which are written to two folders, one for uniquely mapped reads, one for ambiguously mapped reads. In each folder, count tables in tsv format are tagged with Level 0–4, (see details in Fig. 1b).

At Level 1, which classifies reads as miRNAs or isomiRs, the percentages of miRNA and isomiR (miRNA/miRNA+isomiR) are different between the two sample groups. Differences are also observed in individual miRNAs within the same sample group (see Supplementary Fig. S4).

For Level 2, isomiRs are further sub-classed into *same length*, *longer* and *shorter*, based on read length. In this dataset, many miRNAs show distinct differences between sample groups, for all length groups (see Supplementary Fig. S5).

In Level 3, Jasmine further groups reads according to end and modification type. In Supplementary Figure S6, the '*poly*', '*sr-3d*' and '*lr-3e*' are the dominant isomiR types, with many modifications occurring at the 3′ end, consistent with previously published reports.

In Level 4, performs no grouping, working with on unique reads in the sequencing data, similar with miRGFF. The output count table is larger dataset and noisier than other levels (Supplementary Fig. S7), making it harder to discern any pattern in the isomiR data.

Jasmine can also provide intermediate-summarized isomiR population information '*mature*', '*5.isomiR*', '*3.isomiR*', '*sub.isomiR*' and '*mix.isomiR*', used in many other tools (see Supplementary Table S2).

Jasmine also generates detailed information about polymorphisms for investigating A-to-I editing or SNP profiling. The precursor arm usage profile is also reported (see Supplementary Fig. S8).

5 Conclusion

Jasmine is a Java based isomiR-profiling tool for parsing SAM alignment files and reporting isomiR populations using a hierarchal isomiR nomenclature. Jasmine can be configured according to software tools, miRNA annotation set and level of isomiR classification. This gives the user a high degree of flexibility in an analysis, while retaining the ability to compare results across different studies.

Acknowledgements

Part of this work was presented at the NGS conference in Barcelona, Spain, 03–05 April 2017.

Funding

This work was supported by the Helse Sør-Øst grant 2016122 and Norwegian Research Council grant 274715.

Conflict of Interest: none declared.

References

Amsel,D. *et al.* (2017) Evaluation of high-throughput isomiR identification tools: illuminating the early isomiRome of *Tribolium castaneum*. *BMC Bioinformatics*, **18**, 359.

Bartel,D.P. (2018) Metazoan MicroRNAs. *Cell*, **173**, 20–51.

Bolger,A.M. *et al.* (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, **30**, 2114–2120.

Ewels,P. *et al.* (2016) MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, **32**, 3047–3048.

Kozomara,A.*et al*. (2019) miRBase: from microRNA sequences to function. *Nucleic Acids Res*, **47**, D155–D162.

Langmead,B. *et al*. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol*., **10**, R25.

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

Morin,R.D. *et al*. (2008) Application of massively parallel sequencing to microRNA profiling and discovery in human embryonic stem cells. *Genome Res*., **18**, 610–621.

Neilsen,C.T. *et al*. (2012) IsomiRs–the overlooked repertoire in the dynamic microRNAome. *Trends Genet*., **28**, 544–549.

Nogales-Cadenas,R. *et al*. (2016) MicroRNA expression and gene regulation drive breast cancer progression and metastasis in PyMT mice. *Breast Cancer Res*., **18**, 75.

Urgese,G. *et al*. (2016) isomiR-SEA: an RNA-Seq analysis tool for miRNAs/isomiRs expression level profiling and miRNA-mRNA interaction sites evaluation. *BMC Bioinformatics*, **17**, 148.

Vitsios,D.M. and Enright,A.J. (2015) Chimira: analysis of small RNA sequencing data and microRNA modifications. *Bioinformatics*, **31**, 3365–3367.

Zhong,X. *et al*. (2019) miRBaseMiner, a tool for investigating miRBase content. *RNA Biol*., 1–13.