COMPSRA: a **COM** prehensive **P**latform for **S**mall **R**NA-seq data **A**nalysis (v1.0)

Jiang Li[1], Alvin Kho[2], and Kelan Tantisira[1,3]

[1]The Channing Division of Network Medicine, Department of Medicine, Brigham & Women's Hospital and Harvard Medical School, Boston, MA, USA.
[2]Boston Children's Hospital.
[3]Division of Pulmonary and Critical Care Medicine, Department of Medicine, Brigham and Women's Hospital, and Harvard Medical School, Boston, MA, USA.

September 7, 2019

Contents

1 Introduction

COMPSRA was composed of five functional modules: Quality Control, Alignment, Annotation, Microbe and Function. They are integrated into a pipeline and each module can also process independently (Figure 1).

Quality Control: To deal with fastq files and filter out the adapter sequences and reads with low quality. FASTQ files from the small RNA sequencing of biological samples are the default input. First, the adapter portions of a read are trimmed along with any randomized bases at ligation junctions that are produced by some small RNA-seq kits (e.g., NEXTflex^{TM} Small RNA-Seq kit). The read quality of the remaining sequence is evaluated using its corresponding Phred score. Poor quality reads are removed according to quality control parameters set in the command line (-rh 20 –rt 20 –rr 20). Users can specify qualified reads of specific length intervals for input into subse-quent modules.

Alignment: To align the clean reads to the reference genome. COMPSRA uses STAR as its default RNA sequence aligner with default parameters which are customizable on the command line. Qualified reads from the QC module output are first mapped to the human genome hg19/hg38, and then aligned reads are quantified and annotated in the Annotation Module. Reads that could not be mapped to the human genome are saved into a FASTA file for input into the Microbe Module.

Annotation: To annotate different kinds of circulating RNAs based on the alignment result. COMPSRA currently uses several different small RNA databases for annotating human genome mapped reads and provides all the possible annotations: miRBase (Kozomara and Griffiths-Jones, 2011) for miRNA; piRNABank (Sai Lakshmi and Agrawal, 2008); piRBase (Zhang, et al., 2014) and piRNACluster (Rosenkranz, 2016) for piRNA; gtRNAdb (Chan and Lowe, 2016) for tRNA; GENCODE release 27 (Harrow, et al., 2012) for snRNA and snoRNA; circBase (Glazar, et al., 2014) for circular RNA. To conform the different reference human genome versions in these databases, we use an automatic LiftOver created by the UCSC Genome Browser Group. All the databases used are already pre-built, enabling speedy annotation.

Microbe: To predict the possible species of microbes existed in the samples. The qualified reads that could not be mapped to the human genome in the Alignment Module are aligned to the nucleotide (nt) database (Coordinators, 2013) from UCSC using BLAST. The four major microbial taxons archaea, bacteria, fungi and viruses are supported.

Function: To perform differential expression analysis and other functional studies to be extended. The read count of each RNA molecule that is identified in the Annotation Module is outputted as a tab-delimited text file according to RNA type. With more than one sample FASTQ file inputs, the out-put are further aggregated into a data matrix of RNA molecules as rows and samples as columns showing the read counts of an RNA molecule across different samples. The user can mark each sample FASTQ file column as either a case or a control in the command line, and perform a case versus control differential expression analysis for each RNA molecule using the Mann-Whitney rank sum test (Wilcoxon Rank Sum Test) as the default statistical test.

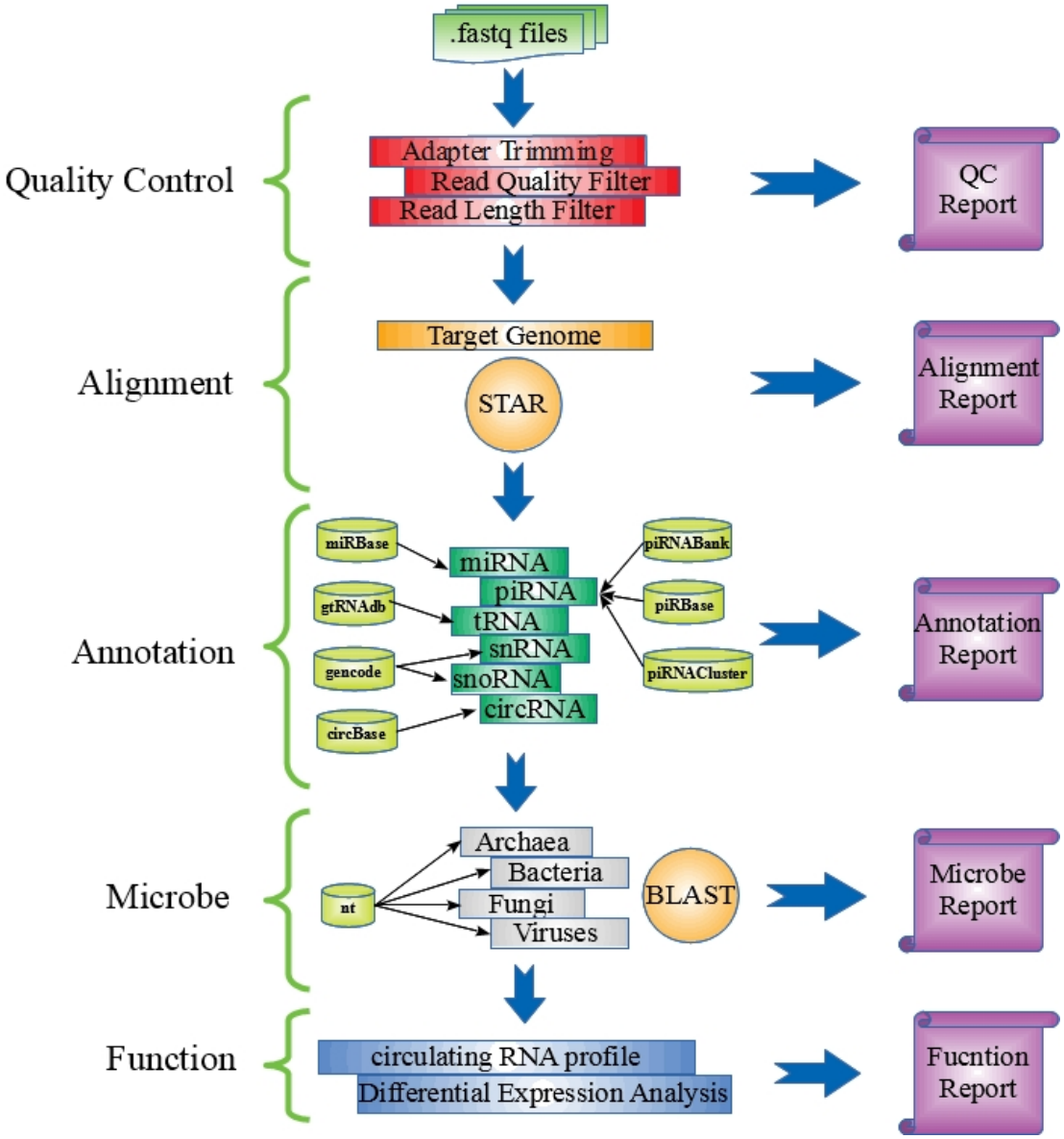

Figure 1: The structure of COMPSRA.

2 Installation

2.1 JAVA Virtual Machine

COMPSRA was achieved by Java language, so Java Runtime Environment (JRE) version 8 (or up) is required. The JRE can be downloaded in ORACLE website (http://www.oracle.com/technetwork/java/javase/downloads/index.html).

2.2 COMPSRA

You can download COMPSRA from https://regepi.bwh.harvard.edu/circurna/ or the GitHub website https://github.com/cougarlj/COMPSRA.

2.3 STAR

COMPSRA will take STAR (v2.5.3a) as the default aligner. STAR can be downloaded from Google Code (https://github.com/alexdobin/STAR/releases/tag/2.5.3a). You can install STAR by yourself in the COMPSRA plug directory or by COMPSRA itself.

2.4 Build-in Installation

When you installed the JAVA Virtual Machine and COMPSRA successfully, you could use the COMPSRA build-in installation system which was collected into COMPSRA ToolKit (-tk).
Download STAR: You can download STAR directly from the command: java -jar COMPSRA.jar -tk -dr -ck star

Download Annotation Database: COMPSRA supports the annotation of miRNA, piRNA, tRNA, snoRNA, snRNA and circRNA with both hg38 and hg19 version. The database can be represented as the combination of RNA name and genome version, connecting by "_", such as miRNA_hg38, piRNA_hg38 and snoRNA_hg19. So you can download these database by the command: java -jar COMPSRA.jar -tk -dr -ck miRNA_hg38,piRNA_hg38,tRNA_hg38,...

Download human genome for STAR: Before running STAR for alignment, you should download the reference genome and build the index. You can download the human reference genome hg38 by the command: java -jar COMPSRA.jar -tk -dr -ck star_hg38. COMPSRA can build the index when it runs for the first time. If failed, please enter the STAR installation directory and set it executive through the command chmod.

3 Examples

Note: We demonstrate COMPSRA in the ~/COMPSRA directory in Linux OS.

```
~$ mkdir COMPSRA
~$ cd COMPSRA/
```

3.1 Preparation

3.1.1 Download COMPSRA

```
~/COMPSRA$ wget https://regepi.bwh.harvard.edu/circurna/COMPSRA_V1.0.zip
~/COMPSRA$ unzip COMPSRA_V1.0.zip
```

Note: After uncompressing the zip file, you can find the following materials:

COMPSRA.jar : This is the compiled jar package of COMPSRA.

COMPSRA_tutorial_v1.0.pdf : The tutorial of COMPSRA v1.0.

bundle_v1 : This directory contains all the resources that COMPSRA may used, such as databases, reference genome, plugs.

example : This directory contains examples for demonstration.

3.1.2 Download Resources

Download miRNA prebuilt databases:

```
~/COMPSRA$ java -jar COMPSRA.jar -tk -dr -ck miRNA_hg38
```

Download piRNA prebuilt databases:

```
~/COMPSRA$ java -jar COMPSRA.jar -tk -dr -ck piRNA_hg38
```

Download tRNA prebuilt databases:

```
~/COMPSRA$ java -jar COMPSRA.jar -tk -dr -ck tRNA_hg38
```

Download snoRNA prebuilt databases:

```
~/COMPSRA$ java -jar COMPSRA.jar -tk -dr -ck snoRNA_hg38
```

Download snRNA prebuilt databases:

```
~/COMPSRA$ java -jar COMPSRA.jar -tk -dr -ck snRNA_hg38
```

Download circRNA prebuilt databases:

```
~/COMPSRA$ java -jar COMPSRA.jar -tk -dr -ck circRNA_hg38
```

Download all prebuilt databases:

```
~/COMPSRA$ java -jar COMPSRA.jar -tk -dr -ck miRNA_hg38,piRNA_hg38,
tRNA_hg38,snoRNA_hg38,snRNA_hg38,circRNA_hg38
```

3.1.3 Download and install STAR

Download STAR:

```
~/COMPSRA$ java -jar COMPSRA.jar -tk -dr -ck star
```

Download human reference genome hg38:

```
~/COMPSRA$ java -jar COMPSRA.jar -tk -dr -ck star_hg38
```

3.2 Test Examples

3.2.1 Run COMPSRA module by module

QC Module:

```
~/COMPSRA$ java -jar COMPSRA.jar -ref hg38 -qc -ra TGGAATTCTCGGGTGCCAAGG
-rb 4 -rh 20 -rt 20 -rr 20 -rlh 8,17 -in ./example/sample01.fastq -out ./example_out/
```

Alignment Module:

```
~/COMPSRA$ java -jar COMPSRA.jar -ref hg38 -aln -mt star -mbi
-in ./example_out/sample01/sample01_17to50_FitRead.fastq.gz
-out ./example_out/sample01/sample01_17to50_FitRead
```

*(Note: **-mbi** was only needed for the first time when you run COMPSRA and built the index files for STAR. This process will cost about 3 hours.)*

Annotation Module:

```
~/COMPSRA$ java -jar COMPSRA.jar -ref hg38 -ann -ac 1,2,3,4,5,6
-in ./example_out/sample01/sample01_17to50_FitRead_STAR_Aligned.out.bam
-out ./example_out/sample01/sample01_17to50_FitRead_STAR_Aligned
```

(Note: The top three modules can run together in a pipeline.)

```
~/COMPSRA$ java -jar COMPSRA.jar -ref hg38
-qc -ra TGGAATTCTCGGGTGCCAAGG -rb 4 -rh 20 -rt 20 -rr 20 -rlh 8,17
-aln -mt star
-ann -ac 1,2,3,4,5,6
-in ./example/sample01.fastq
-out ./example_out/
```

*(Note: When running multiple samples, you can write the input file names into a single file and use **-inf** instead of **-in**. Also, the three modules can be conducted in one command.)*

```
~/COMPSRA$ java -jar COMPSRA.jar -ref hg38
-qc -ra TGGAATTCTCGGGTGCCAAGG -rb 4 -rh 20 -rt 20 -rr 20 -rlh 8,17
-aln -mt star
-ann -ac 1,2,3,4,5,6
-inf ./example/sample.list
-out ./example_out/
```

Function Module:

```
~/COMPSRA$ java -jar COMPSRA.jar -ref hg38
-fun -fd -fdclass 1,2,3,4,5,6 -fdcase 1-6 -fdctrl 7-12 -fdnorm cpm -fdtest mwu -fdann
-pro COMPSRA_DEG -inf ./example/sample.list -out ./example_out/
```

*(Note: If you only want to merge the count files, you can use **-fm -fms**.)*

```
~/COMPSRA$ java -jar COMPSRA.jar -ref hg38
-fun -fm -fms 1-12 -fdclass 1,2,3,4,5,6 -fdann -pro COMPSRA_MERGE
-inf ./example/sample.list
-out ./example_out/
```

3.2.2 Run COMPSRA in a pipeline

```
~/COMPSRA$ java -jar COMPSRA.jar -ref hg38
-qc -ra TGGAATTCTCGGGTGCCAAGG -rb 4 -rh 20 -rt 20 -rr 20 -rlh 8,17
-aln -mt star -ann -ac 1,2,3,4,5,6
-fun -fd -fdclass 1,2,3,4,5,6 -fdcase 1-6 -fdctrl 7-12 -fdnorm cpm
-fdtest mwu -fdann -pro ALL_DEG -inf ./example/sample.list -out ./example_out/
```

(Note: To merge count files, you can still run COMPSRA in a pipeline.)

```
~/COMPSRA$ java -jar COMPSRA.jar -ref hg38
-qc -ra TGGAATTCTCGGGTGCCAAGG -rb 4 -rh 20 -rt 20 -rr 20 -rlh 8,17
-aln -mt star -ann -ac 1,2,3,4,5,6
-fun -fm -fms 1-12 -fdclass 1,2,3,4,5,6
-fdann -pro ALL_MERGE -inf ./example/sample.list -out ./example_out/
```

3.2.3 Microbe Module (Optional)

(Note: To run Microbe Module, you may need to download more resources. Here, we take archaea as an example.)

Step 1: Download and install BLAST.

```
~/COMPSRA$ java -jar COMPSRA.jar -tk -dr -ck blast
```

Step 2: Download taxonomy information.

```
~/COMPSRA$ java -jar COMPSRA.jar -tk -dr -ck blast_taxonomy
```

Step 3: Download microbial prebuilt database. In COMPSRA, we have prebuilt four microbial databases: blast_archaea, blast_bacteria, blast_fungi, blast_viruses.

```
~/COMPSRA$ java -jar COMPSRA.jar -tk -dr -ck blast_archaea
```

Step 4: Run Microbe Module in COMPSRA with **-mic**.

```
~/COMPSRA$ java -jar COMPSRA.jar -mic -mtool Blast -mdb archaea
-in ./example_out/sample01/sample01_17to50_FitRead_STAR_Aligned_UnMapped.bam
-out ./example_out/
```

(Note: You can still add the Microbe Module into the whole pipeline.)

```
~/COMPSRA$ java -jar COMPSRA.jar -ref hg38
-qc -ra TGGAATTCTCGGGTGCCAAGG -rb 4 -rh 20 -rt 20 -rr 20 -rlh 8,17
-aln -mt star
-ann -ac 1,2,3,4,5,6
-mic -mtool Blast -mdb archaea
-in ./example/sample01.fastq
-out ./example_out/
```

(Note: For multiple samples, take a file list as input.)

```
~/COMPSRA$ java -jar COMPSRA.jar -ref hg38
-qc -ra TGGAATTCTCGGGTGCCAAGG -rb 4 -rh 20 -rt 20 -rr 20 -rlh 8,17
-aln -mt star
-ann -ac 1,2,3,4,5,6
-mic -mtool Blast -mdb archaea
-inf ./example/sample.list
-out ./example_out/
```

4 Options

4.1 General Settings

4.1.1 -h/--help

To display the help information of COMPSRA.

4.1.2 -t/--threads *n*

To set the maximum of threads that COMPSRA will use when running. The default setting is 1.

4.1.3 -pro/--project_name *ProjectName*

To set the project name. The default setting is COMPSRA.

4.1.4 -ref/--ref_genome *hg19/hg38*

To set the reference genome that is used for alignment. Currently, COMPSRA supports hg19 (http://hgdownload.soe.ucsc.edu/goldenPath/hg19/bigZips/chromFa.tar.gz) and hg38 (http://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/hg38.fa.gz) genome version.

4.1.5 -in/--input *file1,file2,...,fileN*

To set the input file. The valid format is fastq file or SAM file.

4.1.6 -inf/--in_file *file.list*

To set the input files through a file list. In the file list, each line should only contain one file without any delimiter.

4.1.7 -out/--output */my/output/path/*

To set the output files. If no setting, COMPSRA will create an output directory in the user working path and take the input prefix in default.

4.2 Quality Control

4.2.1 -qc/--quality_control

To open or close the quality control module.

4.2.2 -ra/--rm_adapter *seq*

To remove the adapter sequences at the 3' (3-prime) end. The commonly used adapter sequences from different kits are listed below:

TruSeq Small RNA (Illumina) TGGAATTCTCGGGTGCCAAGG

Small RNA Kits V1 (Illumina) TCGTATGCCGTCTTCTGCTTGT

Small RNA Kits V1.5 (Illumina) ATCTCGTATGCCGTCTTCTGCTTG

NEXTflex Small RNA Sequencing Kit v3 for Illumina Platforms (Bioo Scientific) TGGAATTCTCGGGTGCCAAGG

LEXOGEN Small RNA-Seq Library Prep Kit (Illumina) TGGAATTCTCGGGTGCC AAGGAACTCCAGTCAC

4.2.3 -rb/--rm_bias *n*

To remove *n* random bases in both 5' (5-prime) and 3' (3-prime) ends after removing the adapter sequence.

4.2.4 -rh/--rm_low_quality_head *score*

To remove the low quality bases with the score less than *score* from 5' (5-prime) end.

4.2.5 -rt/--rm_low_quality_tail *score*

To remove the low quality bases with the score less than *score* from 3' (3-prime) end.

4.2.6 -rr/--rm_low_quality_read *score*

To remove the low quality reads with the average score less than *score*.

4.2.7 -rhh/--rm_head_hard *n*

To remove *n* bases from the 5' (5-prime) end.

4.2.8 -rth/--rm_tail_hard *n*

To remove *n* bases from the 3' (3-prime) end.

4.2.9 -rlh/--rm_read_hard *D1,D2,...,Dn*

To divide the reads into several groups according to *[0,D1),[D1,D2),...,[Dn-1,Dn]*.

4.3 Alignment

4.3.1 -aln/--alignment

To open or close the alignment module.

4.3.2 -mt/--mapping_tool *star*

To set the aligner used in COMPSRA. The default aligner is *star (v2.5.3a)*

4.3.3 -mp/--mapping_param

To set parameters of the aligner. The default settings for *star* are listed below:

- *star*

  --runMode alignReads
  --outSAMtype BAM Unsorted
  --outSAMattributes Standard
  --readFilesCommand zcat
  --outSAMunmapped Within
  --outReadsUnmapped None
  --alignEndsType Local
  --alignIntroMax 1
  --alignIntroMin 2
  --outFilterMismatchNmax 1
  --outFilterMultimapScoreRange 1
  --outFilterScoreMinOverLread 0.66
  --outFilterMatchNminOverLread 0.66
  --outFilterMismatchNoverLmax 0.05
  --outFilterMatchNmin 16
  --outFilterMultimapNmax 1000000

4.3.4 -midx/--mapping_index *R1,R2,...,Rn*

To set the read group that will be used for alignment. The default value is "last" , which means the group with the longest reads. Otherwise, the number *Rn* denotes the index of region when setting the parameter -rlh/-rm_read_hard *D1,D2,...,Dn*.

4.3.5 -mref/--mapping_reference *hg19/hg38*

To set the reference genome in alignment. The default value is the same as the parameter -ref/-ref_genome *hg19/hg38*.

4.4 Annotation

4.4.1 -ann/--annotation

To open/close the annotation module.

4.4.2 -ac/--ann_class *A1,A2,...,An*

To set the small RNA categories that will be annotated. The index of small RNA is listed:

1 miRNA

2 piRNA

3 tRNA

4 snoRNA

5 snRNA

6 circRNA

4.4.3 -aol/--ann_overlap *n*

To set the overlap rate between reads and gene regions. The default value is 1.0.

4.4.4 -aic/--ann_inCluster

To show whether or not piRNAs are in the piRNA clusters when annotating piRNAs. The default value is false.

4.4.5 -atd/--ann_threshold *n*

To set the threshold of read counts of small RNAs. If set, only the small RNAs with the read count more than *n* are displayed. The default value is 1.

4.4.6 -armsm/--ann_remove_sam

If added, the original sam file from alignment module will be removed.

4.5 Microbe

4.5.1 -mic/--microbe

To open/close the microbe module.

4.5.2 -mtool/--mic_tool *blast*

To set the tool that will be used for microbe profiling. Currently, only *blast* is supported.

4.5.3 -mdb/--mic_database *viruses,bacteria,fungi,archaea*

To set the microbial databases used in blast.

4.6 Function

4.6.1 -fun/--function

To open/close the function module.

4.6.2 -fd/--fun_diff_expr

To open/close the function of differential expression analysis.

4.6.3 -fdclass/--fun_diff_class *A1,A2,...,An*

To set the small RNAs that will be performed the differential expression analysis. The format is the same as the parameter -ac/-ann_class *A1,A2,...,An*.

4.6.4 -fdcase/--fun_diff_case *ID1,ID2,...,IDn*

To set the IDs of case samples.

4.6.5 -fdctrl/--fun_diff_control *ID1,ID2,...,IDn*

To set the IDs of control samples.

4.6.6 -fdtest/--fun_diff_test *mwu*

To set the statistic test between case and control samples. Currently, only Mann-Whitney U test is supported.

4.6.7 -fdmic/--fun_diff_mic

If added, COMPSRA will detect the annotation files of microbes. It is valid when running function module separately.

4.6.8 -fmtool/--fun_mtool *blast*

To set the tool that was used for microbe profiling. This parameter can facilitate COMPSRA to decide the input files.

4.6.9 -fmdb/--fun_mdb *viruses,bacteria,fungi,archaea*

To set the microbial databases used in blast. This parameter can facilitate COMPSRA to decide the input files.

4.6.10 -fdann/--fun_diff_ann

If added, COMPSRA will detect the annotation files of all small RNAs. It is valid when running function module separately.

4.6.11 -fm/--fun_merge

To open/close the function of merging.

4.6.12 -fms/--fun_merge_samples *ID1,ID2,...,IDn*

To extract read counts from each sample and merge them in one file by different kinds of small RNAs. The categories are set by the parameter -fdclass/-fun_diff_class *A1,A2,...,An*.

5 FAQ

5.0.1 How much memory does COMPSRA need?

COMPSRA does not cost lots of memory, but if STAR was taken as aligner, and 30G memory is considered at least for human genome.

References