CIRI-cookbook

Release 1.1.0

Jinyang Zhang

Nov 30, 2023

CIRI2

CHAPTER

ONE

ABOUT

CIRI2: Circular RNA identification based on multiple seed matching

CIRI (circRNA identifier) is a novel chiastic clipping signal based algorithm,which can unbiasedly and accurately detect circRNAs from transcriptome data by employing multiple filtration strategies.

1.1 Release Notes

What's new in CIRI2?

1. CIRI2 supports multi-thread and is convenient for large data set.
2. CIRI2 supports variable read lengths and can process combined data sets after trimming.
3. CIRI2 outputs strand of predicted circRNAs.

1.2 Citations:

If you use CIRI2, please cite the following papers:

1. Yuan Gao†, Jinfeng Wang† and Fangqing Zhao*. CIRI: an efficient and unbiased algorithm for de novo circular RNA identification. Genome Biology (2015) 16:4.
2. Yuan Gao, Jinyang Zhang and Fangqing Zhao*. Circular RNA identification based on multiple seed matching. Briefings in Bioinformatics (2017) DOI: 10.1093/bib/bbx014.

CHAPTER

TWO

USAGE

2.1 Commands and arguments

How to run CIRI2:

```
perl CIRI2.pl -I in.sam -O outfile -F ref.fa (-R ref_dir/)
```

*CIRI2 can determine automatically to use PE or SE mode according to input SAM.

The arguments of CIRI2 are as followings:

```
-I, --in
    input SAM file name (required; generated by BWA-MEM)
-O, --out
    output circRNA list name (required)
-F, --ref_file
    FASTA file of all reference sequences. Please make sure this file is
    the same one provided to BWA-MEM. Either this argument or -R/--ref-dir
    is required.
-R, --ref_dir
    directory of reference sequence(s). Please make sure fasta files in
    this directory are from the FASTA file(s) provided to BWA-MEM. Either
    this argument or -F/--ref-file is required.
-A, --anno
    input GTF/GFF3 formatted annotation file name (optional)
-G, --log
    output log file name (optional)
-H, --help
    show this help information
-S, --max_span
    max spanning distance of circRNAs (default: 200000)
-high, --high_strigency
    use high strigency: only output circRNAs supported by more than 2
    distinct PCC signals (default)
-low, --low_strigency
    use low strigency: only output circRNAs supported by more than 2
    junction reads
-0, --no_strigency
    output all circRNAs regardless junction read counts or PCC signals
-U, --mapq_uni
    set threshold for mappqing quality of each segment of junction reads
    (default: 10; should be within [0,30])
-E, --rel_exp
    set threshold for relative expression calculated based on counts of
```

(continues on next page)

(continued from previous page)

```
    junction reads and non-junction reads (optional: e.g. 0.1)
-M, --chrM
    tell CIRI2 the ID of mitochondrion in reference file(s) (default: chrM)
-T, --thread_num
    set number of threads for parallel running (default: 1)
-Q, --quiet
    keep quiet when running
-D, --output_all
    keep the temporary files after running (more disk space would be needed)
```

Reads mapped to mitochondrial sequence (provided by –chrM will be dropped).

2.2 Preparation for using CIRI2

- Trim reads first if necessary. This step is optional. For data sets with good sequencing quality, this step will not largely influence prediction of CIRI2.
- Align the reads to reference to generate SAM file using BWA-MEM tool , which is a split mapping algorithm (http://bio-bwa.sourceforge.net/bwa.shtml).

Recommended protocols for running BWA-MEM:

```
bwa index -a bwtsw ref.fa
bwa mem -T 19 ref.fa reads.fq > aln-se.sam (single-end reads)
bwa mem -T 19 ref.fa read1.fq read2.fq > aln-pe.sam (paired-end reads)
```

When using a nohup command, please make sure a log file is specifically designated so that the output file is a clean SAM record. Recommended protocols as following:

```
bwa index -a bwtsw ref.fa
bwa mem -T 19 ref.fa reads.fq 1> aln-se.sam 2> aln-se.log
bwa mem -T 19 ref.fa read1.fq read2.fq 1> aln-pe.sam 2> aln-pe.log
```

IMPORTANT: Please do not try processing the output SAM (e.g. sorting or filtering), which would confuse CIRI2. CIRI2 has been optimized to analysis SAM file without modification, and efficient filtrations have been included in CIRI2.

2.3 Annotation formats

Although CIRI2 can de novo detect circRNA, it can also use annotation as complementary filtrations.

Please make sure you are using exactly the same version of genomic sequences and their annotations when running CIRI2.

We recommend .gtf annotations generated by ensembl. Here is the link for their latest annotations of model organisms: ftp://ftp.ensembl.org/pub/current_gtf

CHAPTER

THREE

AN EXAMPLE OF RUNNING CIRI2

Before we start, please make sure you have installed parallel Perl 5.8 or higher and use Mac OS X or Linux operation system.

Please download CIRI2 and test_data2.zip from https://sourceforge.net/projects/ciri/files/CIRI2/ and then unzip the three files in test_data2.zip. test.sam is a SAM file of alignment records generated by BWA-MEM. The only parameter of BWA-MEM was -T 19. chr1.fa is the FASTA file of hg19 chromosome 1 downloaded from UCSC and chr1.gtf is annotation for chromosome 1 extracted from version 18 gencode gtf file.

Enter the directory and type as following in your terminal:

```
perl CIRI2.pl -I test.sam -O outfile -F chr1.fa -A chr1.gtf
```

CIRI2 can finish circRNA detecting within a minute, and then you will see the following results in outfile:

```
circRNA_ID        chr        circRNA_start        circRNA_end        #junction_
↪reads        SM_MS_SMS        #non_junction_reads        junction_reads_
↪ratio        circRNA_type        gene_id        strand        junction_reads_ID
chr1:16770127|16775696        chr1        16770127        16775696        8        5_
↪6_0        192        0.077        exon        ENSG00000157191.15,
↪        +        GFGG-GA2-1:2:33:2000:1585,GFGG-GA2-1:2:33:2000:1587,GFGG-GA2-
↪1:2:33:2000:1589,GFGG-GA2-1:2:33:2000:1590,GFGG-GA2-1:2:33:2000:1576,GFGG-GA2-
↪1:2:33:2000:1580,GFGG-GA2-1:2:33:2000:1583,GFGG-GA2-1:2:33:2000:1593,
chr1:16775588|16778510        chr1        16775588        16778510        6        5_
↪5_0        203        0.056        exon        ENSG00000157191.15,
↪        +        GFGG-GA2-1:2:33:2000:1276,GFGG-GA2-1:2:33:2000:1277,GFGG-GA2-
↪1:2:33:2000:1280,GFGG-GA2-1:2:33:2000:1284,GFGG-GA2-1:2:33:2000:1279,GFGG-GA2-
↪1:2:33:2000:1283,
...
```

Or CIRI2 can search circRNAs without the annotation gtf:

```
perl CIRI2.pl -I test.sam -O outfile2 -F chr1.fa
```

The outfile2 is as follows:

```
circRNA_ID        chr        circRNA_start        circRNA_end        #junction_
↪reads        SM_MS_SMS        #non_junction_reads        junction_reads_
↪ratio        circRNA_type        gene_id        strand        junction_reads_ID
chr1:16770127|16775696        chr1        16770127        16775696        8        5_
↪6_0        192        0.077        n/a        /n/a        +        GFGG-GA2-
↪1:2:33:2000:1585,GFGG-GA2-1:2:33:2000:1587,GFGG-GA2-1:2:33:2000:1589,GFGG-GA2-
↪1:2:33:2000:1590,GFGG-GA2-1:2:33:2000:1576,GFGG-GA2-1:2:33:2000:1580,GFGG-GA2-
↪1:2:33:2000:1583,GFGG-GA2-1:2:33:2000:1593,
chr1:16775588|16778510        chr1        16775588        16778510        6        5_
↪5_0        203        0.056        n/a        /n/a        +        GFGG-GA2-
↪1:2:33:2000:1276,GFGG-GA2-1:2:33:2000:1277,GFGG-GA2-1:2:33:2000:1280,GFGG-GA2-
↪1:2:33:2000:1284,GFGG-GA2-1:2:33:2000:1279,GFGG-GA2-1:2:33:2000:1283,
```

(continues on next page)

(continued from previous page)

```
...
```

Columns of output file are split by tabs ("\t" in shell and perl). Each column gives information of a predicted circRNA:

Running time and summary are recorded in oufile.log.

CHAPTER

FOUR

TEST OF CIRI2

You can also test CIRI2 using the framework added since version 2.0.5, though the test itself is not required for running CIRI2. If you test CIRI2 using this framework, please note modules Test::More and Test::Class as well as Perl 5.10 or higher will be needed.

Enter the directory and type as follows in your terminal:

```
perl data/test_CIRI.pl
```

CHAPTER

FIVE

Q&A

(1) How to use multiple threads in CIRI2?

CIRI2 supports multi-thread. To use it, just designate thread number using the argument -T.

It should be noted that using multiple threads will inevitably need more RAM due to the design of parallel Perl, although we have optimized RAM utility in CIRI2.

As a suggestion, please choose to use multi-thread if you are using moderate or high-performance server, which will largely reduce running time for large data sets. For reference, it usually takes no more than 4 hours for 40 Gb data (SAM file >100 Gb) generated from RNaseR treated sample using 4 threads.

If you only have a mini-server or even a 16Gb-RAM iMac, you can still use CIRI2 without paralleling to complete processing large data sets in acceptable time. It usually takes no more than 10 hours to process the above data set using single thread.

We do not recommend to use more than 20 threads in CIRI2, which will cost a large RAM but indeed speed up little.

(2) How to set parameters in BWA-MEM and CIRI2 for different data?

An important argument of BWA-MEM for junction read mapping is -T, which gives the threshold of alignment score of output.

Our simulation study shows that a lower -T than default (30) can improve sensitivity of CIRI2. We recommend -T 19 for most data set (such as read length 60).

We have applied CIRI2 to simulated data and real data. Here are some recommended parameters for short or single-end reads.

1. Single-end reads result in higher FDR comparing to paired-end reads for lack of PEM information as one of filters when using default parameters. Thus we provided parameter -U (e.g. -U 15), to set mapping quality thresholds of a junction read and help control FDR.
2. Although short read length (<60 bp) may lead to lower sensitivity, alteration of parameters for BWA-MEM (e.g. -k 15 and -T 15) to allow alignments for low mapping scores can improve performance for this type of sequencing data.

(3) How to generate a genome annotation file for non-model organisms.

If the genome you use does not have annotation in ensembl, we would recommend you to convert the annotated file to one of the following formats.

1. gtf format:

```
chr1    HAVANA  gene    11869   14412   .       +       .       gene_id
↪"ENSG00000223972.4"; transcript_id "ENSG00000223972.4"; gene_type "pseudogene";␣
↪gene_status "KNOWN"; gene_name "DDX11L1"; transcript_type "pseudogene"; transcript_
↪status "KNOWN"; transcript_name "DDX11L1"; level 2; havana_gene "OTTHUMG00000000961.
↪2";
chr1    HAVANA  transcript      11869   14409   .       +       .       gene_id
↪"ENSG00000223972.4"; transcript_id "ENST00000456328.2"; gene_type "pseudogene";␣
↪gene_status "KNOWN"; gene_name "DDX11L1"; transcript_type "processed_transcript";␣
↪transcript_status "KNOWN"; transcript_name "DDX11L1-002"; level 2; tag "basic";␣
↪havana_gene "OTTHUMG00000000961.2"; havana_transcript "OTTHUMT00000362751.1";
chr1    HAVANA  exon    11869   12227   .       +       .       gene_id
↪"ENSG00000223972.4"; transcript_id "ENST00000456328.2"; gene_type "pseudogene";␣
↪gene_status "KNOWN"; gene_name "DDX11L1"; transcript_type "processed_transcript";␣
↪transcript_status "KNOWN"; transcript_name "DDX11L1-002"; exon_number 1;  exon_id
↪"ENSE00002234944.1";  level 2; tag "basic"; havana_gene "OTTHUMG00000000961.2";␣
↪havana_transcript "OTTHUMT00000362751.1";
```

The last column (column 9) should contains the key words such as "gene_id" and "transcript_id", and each of them is split by a "; ".

1. gff format:

```
AC_000023.1     RefSeq  region  1       202526509       .       +       .       ␣
↪ID=id0;Name=1;Dbxref=taxon:10090;chromosome=1;gbkey=Src;genome=chromosome;mol_
↪type=genomic DNA;strain=mixed
AC_000023.1     BestRefSeq      gene    3222752 3677460 .       -       .       ␣
↪ID=gene0;Name=Xkr4;Dbxref=GeneID:497097,MGI:3528744;description=X Kell blood group␣
↪precursor related family member 4;gbkey=Gene;gene=Xkr4;gene_synonym=AY534250,Gm210,
↪mKIAA1889,XRG4;partial=true
AC_000023.1     BestRefSeq      mRNA    3222752 3677460 .       -       .       ␣
↪ID=rna0;Name=NM_001011874.1;Parent=gene0;Note=The RefSeq transcript aligns at 78%25␣
↪coverage compared to this genomic sequence;Dbxref=GeneID:497097,Genbank:NM_
↪001011874.1,MGI:3528744;exception=annotated by transcript or proteomic data;
↪gbkey=mRNA;gene=Xkr4;product=X Kell blood group precursor related family member 4;
↪transcript_id=NM_001011874.1
AC_000023.1     BestRefSeq      exon    3677303 3677460 .       -       .       ␣
↪ID=id1;Parent=rna0;Note=The RefSeq transcript aligns at 78%25 coverage compared to␣
↪this genomic sequence;Dbxref=GeneID:497097,Genbank:NM_001011874.1,MGI:3528744;
↪exception=annotated by transcript or proteomic data;gbkey=mRNA;gene=Xkr4;product=X␣
↪Kell blood group precursor related family member 4;transcript_id=NM_001011874.1
```

The last column (column 9) should contains the key words such as "gene=" and "transcript_id=", and each of them is split by a ";".

1. another gff format:

```
Chr1    TAIR10  chromosome      1       30427671        .       .       .       ␣
↪ID=Chr1;Name=Chr1
Chr1    TAIR10  gene    3631    5899    .       +       .       ID=AT1G01010;
↪Note=protein_coding_gene;Name=AT1G01010
Chr1    TAIR10  mRNA    3631    5899    .       +       .       ID=AT1G01010.1;
↪Parent=AT1G01010;Name=AT1G01010.1;Index=1
Chr1    TAIR10  protein 3760    5630    .       +       .       ID=AT1G01010.1-
↪Protein;Name=AT1G01010.1;Derives_from=AT1G01010.1
Chr1    TAIR10  exon    3631    3913    .       +       .       Parent=AT1G01010.1
Chr1    TAIR10  five_prime_UTR  3631    3759    .       +       .       ␣
↪Parent=AT1G01010.1
```

The last column (column 9) should contains the key words such as "Parent=", and each of them is split by a ";".

Any questions about CIRI2 please mail to gaoyuan06@mails.ucas.ac.cn.

CHAPTER

SIX

ABOUT

CIRI-AS is a detection tool for circRNA internal components and alternative splicing events.

CHAPTER

SEVEN

USAGE

7.1 How to run CIRI-AS

- with annotation as input:

```
perl CIRI_AS.pl -S in.sam -C in.ciri -O output -F ref.fa (-R ref_dir/) -A anno.gtf
```

- without annotation as input:

```
perl CIRI_AS.pl -S in.sam -C in.ciri -O output -F ref.fa (-R ref_dir/)
```

7.2 Commands and arguments

The arguments of CIRI-AS are as followings:

```
--sam/-S                        input SAM file (required; generated by BWA-MEM using␣
↪PAIRED END mode)
--ciri/-C                        input circRNA list (required; generated by CIRI)
--out/-O                        prefix of output files (required)
--ref_dir/-R                directory of reference sequences (Please make sure FASTA␣
↪files in this directory are the same ones provided to CIRI. Either this argument or␣
↪--ref-file/-F is required.)
--ref_file/-F                FASTA file of all reference sequences (Please make sure␣
↪this file is the same one provided to CIRI. Either this argument or --ref-dir/-R is␣
↪required.)
--anno/-A                        GTF formatted annotation file (optional)
--output_all/-D        if output all processing info (Choose 'yes' would require more␣
↪disk space. default: no)
--log/-G                        output log file name (optional)
--help/-H                        show help information
```

7.3 Preparation for using CIRI-AS

CIRI-AS detects circRNAs internal components and alternative splicing events by processing SAM file generated by BWA-MEM as well as circRNA list generated by CIRI (please see manual of CIRI for its usage details).

CIRI-AS is applicable only to paired-end sequencing data.

7.4 Annotation formats

When annotation file is provided, CIRI_AS can calculate insert length and provide correction value of psi for alternative spliced exons accordingly.

Like CIRI, CIRI-AS can understand GTF/GFF formatted annotation.

We recommend .gtf annotations generated by ensembl. Here is the link for their latest annotations of model organisms: ftp://ftp.ensembl.org/pub/current_gtf

Details of annotation format please see manual of CIRI.

Please make sure you are using exactly the same version of genomic sequences and their annotations when running CIRI-AS.

CHAPTER

EIGHT

AN EXAMPLE OF RUNNING CIRI-AS

Before we start, please make sure you have installed Perl 5.12 or higher and use Mac OS X or Linux operation system.

Please download CIRI-AS and test_data_CIRI_AS.zip and then gunzip the four files in test_data_CIRI_AS.zip.

- `test.sam` is a SAM file of alignment records generated by BWA-MEM. The only parameter of BWA-MEM was -T 19.
- `test.ciri` is a circRNA list generated by CIRI by processing test.sam using default parameters.
- `chr1.fa` is the FASTA file of hg19 chromosome 1 downloaded from UCSC and chr1.gtf is annotation for chromosome 1 extracted from version 18 gencode gtf file.

Enter the directory and type as following in your terminal:

```
perl CIRI_AS.pl -S test.sam -C test.ciri -O outfile -F chr1.fa -A chr1.gtf
```

Or without the annotation gtf:

```
perl CIRI_AS.pl -S test.sam -C test.ciri -O outfile -F chr1.fa
```

CIRI-AS can finish detection within a few minutes, and you will see the following output file:

```
outfile.list
outfile_AS.list
```

Columns of output file are split by tabs ("\t" in perl).

for `outfile.list`:

Each column gives information of detected cirexons and corresponding circRNAs.

- #start_supporting_BSJ_read indicates the count of BSJ read pairs that support the start of this cirexon.
- #end_supporting_BSJ_read indicates the count of BSJ read pairs that support the end of this cirexon.
- sequencing_depth_median indicates the median of sequencing depth within the cirexon. It should be noted that the sequencing depth may also contain sequencing for linear counterparts.

for outfile_AS.list:

- Each column gives information of detected alternative splicing events within circRNAs.

CHAPTER

NINE

ABOUT

Manual of CIRI-full v2.0

If you have any questions, please contact

- Yi Zheng @ Beijing Institutes of Life Science, Chinese Academy of Sciences.
- Email: zhengyi12@mails.ucas.ac.cn

CIRI-full is an accurate, high-throughput approach that uses both BSJ and reverse overlap (RO) features to reconstruct and quantify full-length circular RNAs from RNA-seq data sets. In CIRI-full, the BSJ feature is employed to detect cirexons and to determine the boundaries of circRNAs. The RO feature, deduced from the overlapped sequence of paired-end reads, is used to explore the detailed landscape within boundary sites. The alignments of both BSJ & RO merged reads will be visualized. The relative abundance of isoforms within one circRNA will be estimated according to the coverage and spliced events of BSJ & RO merged reads.

CHAPTER

TEN

INSTALLATION

CIRI-full is developed in JAVA, and it can be performed in any system which has Java SE Runtime Environment.It requires:

```
bwa:                A read mapping tool, which generates SAM file for CIRI-full, CIRI␣
↪& CIRI-AS https://sourceforge.net/projects/bio-bwa/files/
CIRI2:                A circRNA detection tool  https://sourceforge.net/projects/ciri/
CIRI-AS:        A tool to detect cirexon and alternative splicing events in circRNAs␣
↪https://sourceforge.net/projects/ciri/
```

CIRI2 and CIRI-AS are already packed with the CIRI-full software.

After downloading the CIRI-full package, you can extract it by typing:

```
unzip CIRI-full.zip        cd CIRI-full2. Preparation for running CIRI-full
```

Before running CIRI-full, you need to run CIRI and CIRI-AS to detect circRNAs and their associated BSJs and cirexons from your sequence data.

Here is a recommend protocol to run CIRI and CIRI-AS:

```
# Index the reference genome:
bwa index -a bwtsw reference.fa
# Split mapping using bwa-mem:
bwa mem -T 19 -t number_thread reference.fa read_1.fq read_2.fq > read.sam
#  2.3 Running CIRI & CIRI-AS
perl CIRI.pl -I read.sam -O prefix.ciri -F reference.fa -A annotation.gtf -T number_
↪thread
perl CIRI_AS.pl -S read.sam -C prefix.ciri -F reference.fa -A annotation.gtf -O␣
↪prefix -D yes
```

For detailed instructions on above tools, please read the manuals of bwa, CIRI and CIRI-AS.

CHAPTER

ELEVEN

RUNNING CIRI-FULL PIPELINE

The CIRI-full Pipeline module is an automatic pipeline for detecting and reconstructing circRNAs. This pipeline includes `CIRI`, `CIRI-AS` and `CIRI-full` tools, which will finally generate reconstructed full-length circRNA sequences and the annotation of all identified circRNAs.

Before running the Pipeline module, please make sure that `bwa` is added to `$PATH`

The Pipeline module runs from a command line as follows:

```
java -jar CIRI-full.jar Pipeline [options]
```

Options:

```
-1        reads1 of paired-end reads (required, equal length, fastq or fastq.gz␣
↪format)
-2        reads2 of paired-end reads (required, equal length, fastq or fastq.gz␣
↪format)
-r        reference genome in fasta format, the same file used in preparation step␣
↪when building bwa index (required).
-a        annotation file of reference genome in GTF format (optional).
-o        prefix of output files (optional, default: out)
-d        directory of output files (required)
-t        number of threads used in CIRI and bwa mem (optional, default: 1)
-0        output all circRNAs including those with only one BSJ read support␣
↪(optional, option for CIRI)
```

Four folders will be created under the dictionary set by `-d` option, `CIRI_output/`, `CIRI-AS_output/`, `CIRI-full_output/` and `sam/`, which contain the output files of `CIRI`, `CIRI-AS`, `CIRI-full` and `bwa`.

For detailed information of these files, please refer to the following instructions.

CHAPTER

TWELVE

RUNNING CIRI-FULL STEP-BY-STEP

CIRI-full includes three modules, `RO1`, `RO2` and `Merge`. These modules should be performed sequentially in the following order: `RO1`, `RO2` and `Merge`.

12.1 The RO1 module

This module is designed to identify 5'-RO feature on paired-end reads from RNA-seq data set and then, merge these RO containing paired-end reads into long single-end reads.

The RO1 module runs from a command line as follows:

```
java -jar CIRI_full.jar RO1 [options]
```

Options:

```
-1        read1 of paired-end reads (required, equal length)        -2        read2␣
↪of paired-end reads (required, equal length)
-o        prefix of output files (optional,default: out)
-minM        sets the number of minimum 5'-RO length (optional, integer, default 13)
-minI        sets the minimum identity percentage of 5'-RO alignment (optional,␣
↪default 95)
```

RO1 module will generate two output files:

```
prefix_ro1_align.txt
prefix_ro1.fq
```

Description of `prefix_ro1_align.txt`:

Each column gives the alignment information of each read pair which contain 5'-RO feature.

- #read_id
- #alignment_identity
- #start_position_on_read1
- #end_position_on_read1
- #start_position_on_read2
- #end_position_on_read2
- #read_length

Description of `prefix_ro1.fq`

Read pairs with 5'-RO feature are merged into long sequences in FASTQ format. These sequences are taken as candidate RO merged-reads and will be filtered in the following steps.

12.2 The RO2 module

The RO2 module is to analyze the alignment results of candidate RO merged-reads and screen out authentic ones for reconstructing full-length circRNAs.

Data preparation before running the RO2 module:

RO2 module filters RO merged-reads based on the SAM file generated by `bwa-mem`.

A recommended protocol for running bwa-mem:

```
bwa index -a bwtsw reference.fa        bwa mem -T 19 reference.fa prefix_ro1.fq >␣
↪prefix_ro1.sam
```

Note that `prefix_ro1.fq` file is the output file in the previous step (the RO1 module).

The RO2 module runs from a command line as follows:

```
java -jar CIRI_full.jar RO2 [options]Options:
```

Options:

```
-r        reference genome in fasta format, the same file used in the preparation␣
↪step when building bwa index (required).
-s        SAM alignment of prefix.ro1.fq generated by bwa mem (required).
-l        the read length of given RNA-seq paired end data (required).
-range        maximum spanning distance of circRNAs on the reference(optional,␣
↪integer, default 100000).
-o        prefix of output files (required)
```

RO2 module will generate following output files:

```
prefix_ro2.sam
prefix_ro2_info.list
```

Description of `prefix_ro2.sam`:

This file is the SAM alignment of authentic RO reads.

Description of `prefix_ro2_info.list`:

This file gives the detailed alignment information of authentic RO reads.

Columns are separated by tabs:

- #Read_ID
- #Chr
- #BSJ_position
- #Strand
- #Reconstructed_state
- #Cirexon

- #Mapping_order
- #Splice_site_state+
- #Splice_site_state-

#Splice_site_state+/- represents the mapping boundary deviation from the GT/AG splicing site, where -1 indicates that GT/AG splicing site cannot be detected on the current strand; positive value represents the distance between GT/AG splice site and split mapping position.

12.3 The Merge module

The Merge module combines the results of RO2 and CIRI-AS to reconstruct full-length circRNAs.

The Merge module runs from a command line as follows:

```
java -jar CIRI_full.jar Merge [options]
```

Options:

```
-a        annotation file of the reference genome in GTF format (optional).
-c        output file of CIRI (required)
-as        output_all file generated in CIRI-AS (using -D yes argument. This file has␣
↪a suffix "_jav.list" ) (required)
-ro        RO read information file (prefix_ro2_info.list) generated by RO2 module ␣
↪(required)
-o        prefix of output files (required)
-r        reference genome file (in FASTA format) (required)
```

The Merge module will generate three output files.

```
prefix_merge_circRNA_detail.anno
```

Description of `prefix_merge_circRNA_detail.anno`

This file contains mapping information of BSJ reads (detected by CIRI) and RO merged-reads (detected by RO). Reads are clustered according to the BSJ position. Columns are separated by tabs:

- #BSJ
- #Chr
- #Start
- #End
- #GTF-annotated_exon
- #Cirexon
- #Coveage
- #BSJ_reads_information
- #RO_reads_information
- #Original_gene

CHAPTER

THIRTEEN

RUNNING CIRI-VIS

CIRI-vis is a tool for visualizing alignments of BSJ & RO merged reads and estimating the related abundance of isoforms according to the output of CIRI-full (prefix_merge_circRNA_detail.anno) or CIRI-AS (prefix_jav.list).

CIRI-vis.jar runs from a command line as follows:

```
java -jar CIRI-vis.jar [Options]
```

Options:

```
-i                The path of input file of CIRI-vis. (required)
-l                The path of library length file. (required for isoform␣
↪quantification)
-r                The path of reference genome sequence in FASTA format. (required␣
↪for output circRNA sequence)
-list                The list of chosen circRNA BSJ. (optional)
-d                The dictionary of output. Default currentdir/stdir
-max        The maximum expression (BSJ reads number) of circRNA that displayed by␣
↪CIRI-vis. Default 999999999
-min        The minimum expression (BSJ reads number) of circRNA that displayed by␣
↪CIRI-vis. Default 10. **Note: please only use one of -min, -exp, -rank**
-rank        Only display the expression top X% of circRNA
-exp                Only display the top expression circRNA that contain X% of BSJ␣
↪reads.
-iso                The maximum number of considering isoform, default 10. High value␣
↪will make the quantification slower
```

CIRI-vis will output a set of pdf file, a ".list" file and a ".fa" file(if reference genome file is available) in a new created folder(set by "-d" parameter):

- One pdf file display circRNA isoforms on one BSJ.
- ".list" file shows detail information of each isoform.
- ".fa" file shows the sequences of fully reconstructed circRNA isoforms.

Description of `prefix.list`:

This file gives the detailed information of circRNA isoforms. Columns are separated by tabs:

Description of `prefix.fa`:

This FASTA format file will be generated if reference genome sequence is available. It contains the sequence of fully reconstructed isoform. They were named in this format:

```
>(Image_name)#(BSJ) length=(isoform_length) (isoform_BSJ_read_count)/(circRNA_BSJ_
↪read_count)
```

If you want to display only a subset of circRNA, please use parameter "-list" to give CIRI-vis a list of BSJ position. The format should be like:

```
chr10:74474869|74475660
chr8:141856359|141900868
```

Notes:

- IF you ran CIRI-full Pipeline in previous step, the input file will be named prefix_merge_circRNA_detail.anno under CIRI-full_output folder.
- IF you only ran CIRI-AS with '-d yes' parameter in previous step, the input file will be named prefix_jav.list under your CIRI-AS output folder.
- Library length file is necessary for isoform expression estimation. library length file will be prefix _library_length.list under your CIRI-AS output folder

CHAPTER

FOURTEEN

HOW TO RUN THE TEST DATA SET USING CIRI-FULL

Test data sets (FASTQ file, annotation file and reference sequence) are packaged with the CIRI-full software, which can be found in the "CIRI-full_test/" folder. Temporary and final results are given in the "CIRI-full/test_output/" folder.

Here are the commands for running the test data sets:

```
cd CIRI-full_v2.0/CIRI-full_test/
bwa index test_ref.fajava -jar ../CIRI-full.jar Pipeline -1 test_1.fq.gz -2 test_2.fq.
↪gz -a test_anno.gtf -r test_ref.fa -d test_output/ -o testunset DISPLAY
java -jar ../CIRI-vis.jar -i test_output/CIRI-full_output/test_merge_circRNA_detail.
↪anno -l ../CIRI-vis_test/test_library_length.list -r test_ref.fa -d test_output/
↪CIRI-vis_out -min 1
```

If you want to run CIRI-full step by step, you can use the following commands:

```
cd CIRI-full_v2.0/CIRI-full_test/
mkdir test_output
bwa index test_ref.fa
bwa mem -T 19 test_ref.fa test_1.fq.gz test_2.fq.gz > test_output/test.sam
perl ../bin/CIRI2.pl -I test_output/test.sam -O test_output/test.ciri -F test_ref.fa -
↪A test_anno.gtf
perl ../bin/CIRI_AS_v1.2.pl -S test_output/test.sam -C test_output/test.ciri -F test_
↪ref.fa -A test_anno.gtf -O test_output/test -D yes
java -jar ../CIRI-full.jar RO1 -1 test_1.fq.gz -2 test_2.fq.gz -o test_output/test
bwa mem -T 19 test_ref.fa test_output/test_ro1.fq > test_output/test_ro1.sam
java -jar ../CIRI-full.jar RO2 -r test_ref.fa -s test_output/test_ro1.sam -l 250 -o␣
↪test_output/test
java -jar ../CIRI-full.jar Merge -c test_output/test.ciri -as test_output/test_jav.
↪list -ro test_output/test_ro2_info.list -a test_anno.gtf -r test_ref.fa -o test_
↪output/test
unset DISPLAY
java -jar ../CIRI-vis.jar -i test_output/CIRI-full_output/test_merge_circRNA_detail.
↪anno -l ../CIRI-vis_test/test_library_length.list -r test_ref.fa -min 1
```

Note: Please make sure you are using exactly the same version of genomic sequences and their annotations when running CIRI-full.

CHAPTER

FIFTEEN

ABOUT

CIRI-vis is a tool for visualizing alignments of BSJ & RO merged reads and estimating the related abundance of isoforms according to the output of CIRI-full (prefix_merge_circRNA_detail.anno) or CIRI-AS (prefix_jav.list).

15.1 Author

Authors: Yi Zheng(zhengyi@biols.ac.cn), Fangqing Zhao(zhfq@biols.ac.cn)

Maintainer: Yi Zheng

15.2 Release Notes

- Version 1.4

CHAPTER

SIXTEEN

INSTALLATION

16.1 Prerequisites

```
Softwares:
    JavaSE >= 1.6
    bwa
    CIRI2
    CIRI-AS
    CIRI-Full
```

16.2 Install CIRI-vis

CIRI-vis is developed in JAVA, and it can be performed in any system which has Java SE Runtime Environment.

CIRI-vis is already packed with the CIRI-full software under /bin. After downloading the CIRI-full package, you can extract it by typing:

```
unzip CIRI-full_v2.0.zip
```

you can find CIRI-vis.jar in the folder.

CHAPTER

SEVENTEEN

COMMANDS AND ARGUMENTS

17.1 Running CIRI-vis

Input file requirements:

- IF you runned CIRI-full Pipeline in previous step, the input file will be named:XXX_merge_circRNA_detail.anno under CIRI-full_output folder
- IF you only run CIRI-AS with '-d yes' parameter in previous step, the input file will be named XXX_jav.list under your CIRI-AS output folder
- Library length file is nessaracy for isoform expression estimation. library length file will be XXX_library_length.list under your CIRI-AS output folder

CIRI-vis.jar runs from a command line as follows:

```
Usage: java -jar CIRI-vis.jar [Options]

Options:

  -i            The path of input file of CIRI-vis. (required)
  -l        The path of library length file. (required for isoform quantification)
  -r        The path of reference genome sequence in FASTA format. (required for␣
↪output circRNA sequence)
  -list     The list of choosen circRNA BSJ.(It is needed when more than one sample)
  -d        The dictionary of output. Default currentdir/stdir
  -o        The prefix of output. Default stout (optional)
  -type     The format of figure. you can select pdf or svg or both. Default pdf␣
↪(optional)
  -max      The maximum expression (BSJ reads number) of circRNA that displayed by␣
↪CIRI-vis. Default 999999999 (optional)
  -min      The minimum expression (BSJ reads number) of circRNA that displayed by␣
↪CIRI-vis. Default 5. Note: please only use one of -min, -exp, -rank (optional)
  -rank     Only display the expression top X% of circRNA (optional)
  -exp      Only display the top expression circRNA that contain X% of BSJ reads.␣
↪(optional)
  -iso      The maximum number of considering isoform, default 10. High value will␣
↪make the quantification slower (optional)
  -ran      Set random seed, default 0.(optional)
```

Examples:

For one sample

```
java -jar CIRI-vis.jar -i A_merge_circRNA_detail.anno -l A_library_length.list -r Ref.
↪fa -d out -o prefix
```

For more than one samples

```
java -jar CIRI-vis.jar -i A_merge_circRNA_detail.anno B_merge_circRNA_detail.anno -l␣
↪A_library_length.list B_library_length.list -r Ref.fa -d out -o prefix -list test.
↪txt
```

The format of the "test.txt" (input of -list) should be:

```
chr10:74474869|74475660
chr8:141856359|141900868
...
```

17.2 Output files

Description of `prefix.list`:

This file gives the detailed information of circRNA isoforms. Columns are separated by tabs:

When more than one sample

Description of `prefix.fa`:

This FASTA format file will be generated if reference genome sequence is available. It contains the sequence of fully reconstructed isoform. They were named in this format:

```
>(Image_name)#(BSJ) length=(isoform_length) (isoform_BSJ_read_count)/(circRNA_BSJ_
↪read_count)
```

CHAPTER

EIGHTEEN

EXAMPLE USAGE

Test data sets (FASTQ file, annotation file and reference sequence) are packaged with the CIRI-full software, which can be found in the CIRI-full_test/ folder. Temporary and final results are given in the CIRI-full/test_output/ folder.

Here are the commands for running the test data sets:

```
cd CIRI-full_v2.0/CIRI-full_test/
bwa index test_ref.fa
java -jar ../CIRI-full.jar Pipeline -1 test_1.fq.gz -2 test_2.fq.gz -a test_anno.gtf -
↪r test_ref.fa -d test_output/ -o test
unset DISPLAY
java -jar CIRI-vis.jar -i test_output/CIRI-full_output/test_merge_circRNA_detail.anno␣
↪-l ../CIRI-vis_test/test_library_length.list -r test_ref.fa -d test_output/CIRI-vis_
↪out -min 1
```

CHAPTER

NINETEEN

ABOUT

CIRIquant: comprehensive toolkit for circRNA quantification and differential expression analysis

CIRIquant is a comprehensive analysis pipeline for circRNA detection and quantification in RNA-Seq data

19.1 Author

Authors: Jinyang Zhang(zhangjinyang@biols.ac.cn), Fangqing Zhao(zhfq@biols.ac.cn)

Maintainer: Jinyang Zhang

19.2 Release Notes

- Version 1.1: Added support for stranded library and GFF3 format input.
- Version 1.0: The first released version of CIRIquant.

19.3 License

19.4 Citing CIRIquant

- Zhang, J., Chen, S., Yang, J. et al. Accurate quantification of circular RNAs identifies extensive circular isoform switching events. Nat Commun 11, 90 (2020) doi:10.1038/s41467-019-13840-9

CHAPTER

TWENTY

INSTALLATION

NOTES: Only python2 is supported

20.1 Prerequisites

```
Softwares:
- bwa
- hisat2
- stringtie
- samtools >= 1.9 (*older version of samtools may use deprecated parameters in `sort`␣
↪and `index` commands*)

Python packages:
- PyYAML
- argparse
- pysam
- numpy
- scipy
- scikit-learn
```

20.2 Use the released version (Recommended)

Download the latest released version of CIRIquant from GitHub

The released package is a packed conda environment including all dependencie, make sure you have installed anaconda in your environment

```
# Download packed package
wget https://github.com/bioinfo-biols/CIRIquant/releases/download/v1.1.3/CIRIquant_v1.
↪1.3.tar.gz
mkdir -p CIRIquant_env
tar zxvf CIRIquant_v1.1.3.tar.gz -C CIRIquant_env

# Configuration environments (required)
conda activate ./CIRIquant_env
cd CIRIquant_env
make
```

Activate CIRIquant environment and test

```
conda activate /path/to/CIRIquant_env
which CIRIquant
```

20.3 Install CIRIquant and dependencies using conda (Not recommended)

Save the following content to a file called environment.yml:

```
name: CIRI
channels:
- defaults
- bioconda
- conda-forge
dependencies:
- bioconda::bwa=0.7.17
- bioconda::hisat2=2.2.0
- bioconda::stringtie=2.1.1
- bioconda::samtools>=1.10
- bioconda::bioconductor-edger=3.28.0
- bioconda::bioconductor-limma=3.42.0
- conda-forge::r-statmod=1.4.35
- conda-forge::r-base=3.6
- conda-forge::r-optparse=1.6.6
- python=2.7.15
- pip=20.0.2
- perl=5.26.2
- pip:
  - CIRIquant>=1.1.2
  - numexpr==2.6.9
  - numpy==1.16.4
  - pysam==0.15.2
  - PyYAML==5.4
  - scikit-learn==0.20.3
  - scipy==1.2.2
  - argparse>=1.2.1
```

After you have saved the file just run:

```
# this installs the dependencies specified in the yml file
conda env create -f environment.yml
# this activates the conda environment
conda activate CIRI

# this will return the path bwa, hisat2, stringtie or samtools are installed to
# these paths need to be specified in the CIRI configuration file when running the␣
↪tool
which bwa
which hisat2
which stringtie
which samtools
```

20.4 Install CIRIquant using pip

```
pip install CIRIquant
```

20.5 Install CIRIquant from source code

Use the setup.py for CIRIquant installation (clean install using `virutalenv` is highly recommended).

```
# create and activate virtual env
pip install virtualenv
virtualenv -p /path/to/your/python2/executable venv
source ./venv/bin/activate

# Install CIRIquant and its requirement automatically
tar zxvf CIRIquant.tar.gz
cd CIRIquant
python setup.py install

# Manual installation of required pacakges is also supported
pip install -r requirements.txt
```

The package should take approximately 40 seconds to install on a normal computer.

CHAPTER

TWENTYONE

USAGE 1: CIRCRNA QUANTIFCATION

21.1 Basic options

```
usage: CIRIquant [-h] [--config FILE] [-1 MATE1] [-2 MATE2] [-o DIR]
                 [-p PREFIX] [-t INT] [-a INT] [-l INT] [--ciri3] [-v]
                 [--version] [-e LOG] [--bed FILE] [--circ FILE] [--tool TOOL]
                 [--RNaseR FILE] [--bam BAM] [--no-gene] [--no-fsj]
                 [--bsj-file FILE]

optional arguments:
  -h, --help            show this help message and exit
  --config FILE         Config file in YAML format
  -1 MATE1, --read1 MATE1
                        Input mate1 reads (for paired-end data)
  -2 MATE2, --read2 MATE2
                        Input mate2 reads (for paired-end data)
  -o DIR, --out DIR     Output directory, default: ./
  -p PREFIX, --prefix PREFIX
                        Output sample prefix, default: input sample name
  -t INT, --threads INT
                        Number of CPU threads, default: 4
  -a INT, --anchor INT  Minimum anchor length for junction alignment, default:
                        5
  -l INT, --library-type INT
                        Library type, 0: unstranded, 1: read1 match the sense
                        strand,2: read1 match the antisense strand, default: 0
  -v, --verbose         Run in debugging mode
  --version             show program's version number and exit
  -e LOG, --log LOG     Log file, default: out_dir/prefix.log
  --bed FILE            bed file for putative circRNAs (optional)
  --circ FILE           circRNA prediction results from other softwares
  --tool TOOL           circRNA prediction tool, required if --circ is
                        provided
  --RNaseR FILE         CIRIquant result of RNase R sample
  --bam BAM             hisat2 alignment to reference genome
  --no-gene             Skip stringtie estimation for gene abundance
  --no-fsj              Skip FSJ extraction to reduce run time
  --bsj-file FILE       output BSJ read IDs to file (optional)
```

NOTE:

- For now, –circ and –tool options support results from `CIRI2` / `CIRCexplorer2` / `DCC` / `KNIFE` / `MapSplice` / `UROBORUS` / `circRNA_finder` / `find_circ`

- For tools like `DCC` and `circRNA_finder`, please manually remove duplicated circRNAs with same junction postion but have opposite strands.
- Gene expression values are needed for normalization, do not use `--no-gene` if you need to run DE analysis afterwards.

21.2 Example YAML config

A YAML-formated config file is needed for CIRIquant to find software and reference needed.

A valid example of minimal config file:

```
reference:
  fasta: /home/zhangjy/Data/database/hg19.fa
  gtf: /home/zhangjy/Data/database/gencode.v19.annotation.gtf
  bwa_index: /home/zhangjy/Data/database/hg19/_BWAtmp/hg19
  hisat_index: /home/zhangjy/Data/database/hg19/_HISATtmp/hg19
```

An example of supported config file:

```
// Example of config file
name: hg19
tools:
  bwa: /home/zhangjy/bin/bwa
  hisat2: /home/zhangjy/bin/hisat2
  stringtie: /home/zhangjy/bin/stringtie
  samtools: /home/zhangjy/bin/samtools

reference:
  fasta: /home/zhangjy/Data/database/hg19.fa
  gtf: /home/zhangjy/Data/database/gencode.v19.annotation.gtf
  bwa_index: /home/zhangjy/Data/database/hg19/_BWAtmp/hg19
  hisat_index: /home/zhangjy/Data/database/hg19/_HISATtmp/hg19
```

21.3 Example circRNA bed file

For quantification of user-provided circRNAs, a list of junction sites in bed format is required, the 4th column must be in "chrom:startlend" format. For example:

```
chr1    10000   10099   chr1:10000|10099    .   +
chr1    31000   31200   chr1:31000|31200    .   -
```

21.4 Example Usage

21.4.1 Recommended: Predict circRNAs using CIRI2 (packaged in CIRIquant)

```
CIRIquant -t 4 \
          -1 ./test_1.fq.gz \
          -2 ./test_2.fq.gz \
          --config ./chr1.yml \
```

(continues on next page)

(continued from previous page)

```
          -o ./test \
          -p test
```

21.4.2 Quantify circRNAs using provided BED format input

```
CIRIquant -t 4 \
          -1 ./test_1.fq.gz \
          -2 ./test_2.fq.gz \
          --config ./chr1.yml \
          -o ./test \
          -p test \
          --bed your_circRNAs.bed
```

21.4.3 Quantify circRNAs using results from other tools

For example, if you have `find_circ` results of predicted circRNAs.

```
CIRIquant -t 4 \
          -1 ./test_1.fq.gz \
          -2 ./test_2.fq.gz \
          --config ./chr1.yml \
          -o ./test \
          -p test \
          --circ find_circ_results.txt \
          --tool find_circ
```

21.5 Output format

The main output of CIRIquant is a GTF file, that contains detailed information of BSJ and FSJ reads of circRNAs and annotation of circRNA back-spliced regions in the attribute columns

Description of each columns's value

The attributes containing several pre-defined keys and values:

CHAPTER

TWENTYTWO

USAGE 2: RNASE R EFFECT CORRECTION

When you have both RNase R treated and untreated samples, CIRIquant can estimate the before-treatment expression levels of circRNAs detected in RNase R data.

In order to remove RNase R treatment effect, two steps are needed:

1. Run CIRIquant with RNase R treated sample.
2. Run CIRIquant with untreaded total RNA sample, specific `--RNaseR` option using the output gtf file in Step1

Then, CIRIquant will output estimated expression levels of circRNAs detected in RNaseR data, and the header lines will include additional information of RNase R treatment effciency.

22.1 Example usage

```
# Step1. Run CIRIquant with RNase R treated data
CIRIquant --config ./hg19.yml \
          -1 ./RNaseR_treated_1.fq.gz \
          -2 ./RNaseR_treated_2.fq.gz \
          --no-gene \
          -o ./RNaseR_treated \
          -p RNaseR_treated \
          -t 6

# Step2. Run CIRIquant with untreated total RNA
CIRIquant --config ./hg19.yml \
          -1 ./TotalRNA_1.fq.gz \
          -2 ./TotalRNA_2.fq.gz \
          -o ./TotalRNA \
          -p TotalRNA \
          -t 6 \
          --RNaseR ./RNaseR_treated/RNaseR_treated.gtf
```

CHAPTER

TWENTYTHREE

USAGE 3: DIFFERENTIAL EXPRESSION ANALYSIS

23.1 Study without biological replicate

For sample without replicate, the differential expression & differential splicing analysis is performed using `CIRI_DE`

```
Usage:
  CIRI_DE [options] -n <control> -c <case> -o <out>

  <control>         CIRIquant result of control sample
  <case>            CIRIquant result of treatment cases
  <out>             Output file

Options (defaults in parentheses):

  -p                p value threshold for DE and DS score calculation (default: 0.05)
  -t                numer of threads (default: 4)

Example usage:
  CIRI_DE -n control.gtf -c case.gtf -o CIRI_DE.tsv
```

The output format `CIRI_DE` is in the format below:

23.2 Study with biological replicates

For study with biological replicates, a customed analysis pipeline of edgeR is recommended and we provide `prep_CIRIquant` to generate matrix of circRNA expression level / junction ratio and `CIRI_DE_replicate` for DE analysis

Step1: Prepare CIRIquant output files

One should provide a text file listing sample information and path to CIRIquant output GTF files

```
CONTROL1 ./c1/c1.gtf C 1
CONTROL2 ./c2/c2.gtf C 2
CONTROL3 ./c3/c3.gtf C 3
CASE1 ./t1/t1.gtf T 1
CASE2 ./t2/t2.gtf T 2
CASE3 ./t3/t3.gtf T 3
```

The first three columns is required by default. For paired samples, you could also add a column of subject name.

Note: If you are planning to use CIRI_DE for differential expression, then group name in column 3 must be either "C" or "T".

Then, run `prep_CIRIquant` to summarize the circRNA expression profile in all samples

```
Usage:
  prep_CIRIquant [options]

  -i                the file of sample list
  --lib             where to output library information
  --circ            where to output circRNA annotation information
  --bsj             where to output the circRNA expression matrix
  --ratio           where to output the circRNA junction ratio matrix

Example:
  prep_CIRIquant -i sample.lst \
                 --lib library_info.csv \
                 --circ circRNA_info.csv \
                 --bsj circRNA_bsj.csv \
                 --ratio circRNA_ratio.csv
```

These count matrices (CSV files) can then be imported into R for use by DESeq2 and edgeR (using the DESeqDataSetFromMatrix and DGEList functions, respectively).

Step2: Prepare StringTie output

The output of StringTie should locate under `output_dir/gene/prefix_out.gtf`. You need to use prepDE.py from stringTie to generate the gene count matrix for normalization.

For example, one can provide a text file `sample_gene.lst` containing sample IDs and path to StringTie outputs:

```
CONTROL1 ./c1/gene/c1_out.gtf
CONTROL2 ./c2/gene/c2_out.gtf
CONTROL3 ./c3/gene/c3_out.gtf
CASE1 ./t1/gene/t1_out.gtf
CASE2 ./t2/gene/t2_out.gtf
CASE3 ./t3/gene/t3_out.gtf
```

Then, run `prepDE.py -i sample_gene.lst` and use `gene_count_matrix.csv` generated under current working directory for further analysis.

Step3: Differential expression analysis

For differential analysis using `CIRI_DE_replicate`, you need to install a R environment and `edgeR` package from Bioconductor.

```
usage: CIRI_DE_replicate [-h] --lib FILE --bsj FILE --gene FILE --out
                             FILE --out2 FILE

optional arguments:
  -h, --help   show this help message and exit
  --lib FILE   library information
  --bsj FILE   circRNA expression matrix
  --gene FILE  gene expression matrix
  --out FILE   output result of circRNA differential expression analysis
  --out2 FILE  output result of gene differential expression analysis

Example:
  CIRI_DE_replicate \
          --lib  library_info.csv \
          --bsj  circRNA_bsj.csv \
          --gene gene_count_matrix.csv \
```

(continues on next page)

(continued from previous page)

```
        --out  circRNA_de.tsv \
        --out2 gene_de.tsv
```

Please be noted that the output results is **unfiltered**, and you could apply a more stringent filter on expression values to get a more convincing result.

CHAPTER

TWENTYFOUR

TEST DATA

24.1 Download test dataset

Test dataset can be downloaded from Github.

```
wget https://github.com/Kevinzjy/CIRIquant/releases/download/v0.2.0/test_data.tar.gz
tar zxvf test_data.tar.gz
```

24.2 circRNA quantification

Folder `quant` contain the test dataset for circRNA quantification.

24.2.1 1. Generate hisat2 and bwa index

```
cd ./test_data/quant
bwa index -a bwtsw -p chr1.fa chr1.fa
hisat2-build ./chr1.fa ./chr1.fa
```

24.2.2 2. Customize the configuration

Replace the path of `bwa`/`hisat2`/`stringtie`/`samtools` in `chr1.yml` with your own version.

24.2.3 3. Run test dataset

Test data set can be retrived under `test_data/quant` folder, you can replace the path of required software in the `chr1.yml` with your own version

```
CIRIquant -t 4 \
          -1 ./test_1.fq.gz \
          -2 ./test_2.fq.gz \
          --config ./chr1.yml \
          --no-gene \
          -o ./test \
          -p test
```

The demo dataset should take approximately 5 minutes on a personal computer. It has been tested on my PC with Intel i7-8700 processor and 16G of memory, running Ubuntu 18.04 LTS.

The structure of output directory `./test` should be like this:

```
test
├── align
│   ├── test.bam
│   ├── test.sorted.bam
│   └── test.sorted.bam.bai
├── circ
│   ├── test.ciri
│   ├── test.ciri.bed
│   ├── test_denovo.bam
│   ├── test_denovo.sorted.bam
│   ├── test_denovo.sorted.bam.bai
│   ├── test_index.1.ht2
│   ├── test_index.2.ht2
│   ├── test_index.3.ht2
│   ├── test_index.4.ht2
│   ├── test_index.5.ht2
│   ├── test_index.6.ht2
│   ├── test_index.7.ht2
│   ├── test_index.8.ht2
│   ├── test_index.fa
│   └── test_unmapped.sam
├── CIRIerror.log
├── test.bed
├── test.gtf
└── test.log
```

Then, you can check the main output in `./test/test.gtf`.

24.3 Differential expression analysis

Folder `DE` contain the test dataset for differential expression analysis

```
cd ./test_data/DE

# Test for DE-score and DS-score calculation
CIRI_DE -n ctrl.gtf \
        -c case.gtf \
        -o CIRI_DE.tsv

# Test for RNase R correction
CIRI_DE -n ctrl_corrected.gtf \
        -c case_corrected.gtf \
        -o CIRI_DE_corrected.tsv
```

CHAPTER

TWENTYFIVE

ABOUT

25.1 CIRI-long: circular RNA identifier using long-read sequencing data

Circular RNA Identification for Long-Reads Nanopore Sequencing Data

25.2 Author

Authors: Jinyang Zhang(zhangjinyang@biols.ac.cn), Fangqing Zhao(zhfq@biols.ac.cn)

Maintainer: Jinyang Zhang

25.3 Release Notes

- version 1.1.0: Add convert_bed.py, update pyccs dependency, fixed bugs
- version 1.0.3: Add output of circRNA isoform usage index, fixed bugs
- version 1.0.2: Add fast mode for ccs detection and option for user-provided circRNA annotation
- version 1.0.1: Fixed bug
- version 1.0: First released version

25.4 License

25.5 Citing CIRI-long

- Zhang, J., Hou, L., Zuo, Z., Ji, P., Zhang, X., Xue, Y., & Zhao, F. (2021). Comprehensive profiling of circular RNAs with nanopore sequencing and CIRI-long. Nature Biotechnology. https://doi.org/10.1038/s41587-021-00842-6

CHAPTER

TWENTYSIX

INSTALLATION

26.1 Dependency

- `gcc 4.8+` or `clang 3.4+` and `cmake 3.2+` is needed
- **Only python>=3.7 is supported**
- CIRI-long requires pysam lib, which need executable binary and header of zlib, bzip2, xz, please refer to documentation of pysam for installation instructions
- all python dependencies are listed in requirements.txt
- `samtools` version `1.9` or higher
- python binding of minimap2

26.2 Install CIRI-long from source code

Installation under `virtualenv` is highly recommended. You can simply clone the whole repository, then use `make` to start a complete installation

```
git clone https://github.com/bioinfo-biols/CIRI-long.git CIRI-long
cd CIRI-long

# Create virtual environment
python3 -m venv venv

# Activate virtualenv
source ./venv/bin/activate

# Install CIRI-long
make

# Test for installation
make test
```

The package should take less than 1 min to install on a normal computer.

26.3 Install CIRI-long using pip

```
pip install CIRI-long
```

CHAPTER

TWENTYSEVEN

USAGE

27.1 Basic usage

```
usage: CIRI-long [-h] [-v] {call,collapse} ...

positional arguments:
  {call,collapse}  commands

optional arguments:
  -h, --help       show this help message and exit
  -v, --version    show program's version number and exit
```

CIRI-long have two main functions, including (1) candidate circRNAs identification and (2) isoform collapsing.

27.2 Step1. circRNA identification

27.2.1 Basic options

```
usage: CIRI-long call [-h] [-i READS] [-o DIR] [-r REF] [-p PREFIX] [-a GTF] [--
↪canonical] [-t INT] [--debug]

optional arguments:
  -h, --help            show this help message and exit
  -i READS, --in READS  Input reads.fq.gz
  -o DIR, --out DIR     Output directory, default: ./
  -r REF, --ref REF     Reference genome FASTA file
  -p PREFIX, --prefix PREFIX
                        Output sample prefix, (default: CIRI-long)
  -a GTF, --anno GTF    Genome reference gtf, (optional)
  -c CIRC, --circ CIRC  Additional circRNA annotation in bed/gtf format,
                        (optional)
  -t INT, --threads INT
                        Number of threads, (default: use all cores)
  --debug               Run in debugging mode, (default: False)
```

NOTE:

- A bwa index for reference genome is required, please use `bwa index` command to generate bwa index before running CIRI-long.

27.2.2 Example Usage:

Demo dataset can be downloaded from the GitHub release

```
# Download demo dataset
wget https://github.com/bioinfo-biols/CIRI-long/releases/download/v0.6-alpha/CIRI-
↪long_test_data.tar.gz

# Decompress demo dataset
tar zxvf CIRI-long_test_data.tar.gz
cd test_data

# Build bwa index before running CIRI-long
bwa index -a bwtsw mm10_chr12.fa mm10_chr12.fa

# Run CIRI-long to identify circular reads from sequencing reads
CIRI-long call -i test_reads.fa \
               -o ./test_call \
               -r mm10_chr12.fa \
               -p test \
               -a mm10_chr12.gtf \
               -t 8
```

27.2.3 Output Files

The output directory should have the following structure:

```
test_call
├── test.cand_circ.fa
├── test.json
├── test.log
├── test.low_confidence.fa
└── tmp
    ├── ss.idx
    ├── test.ccs.fa
    └── test.raw.fa

1 directory, 7 files
```

27.2.4 Using non-canonical splice signals

If you would like to use other splice signals, please modify the dict `SPLICE_SIGNAL` in align.py in format: {(5'SS, 3'SS): Priority}

Default configuration:

```
SPLICE_SIGNAL = {
    ('GT', 'AG'): 0,  # U2-type
    ('GC', 'AG'): 1,  # U2-type
    ('AT', 'AC'): 2,  # U12-type
    ('GT', 'AC'): 2,  # U12-type
    ('AT', 'AG'): 2,  # U12-type
}
```

27.2.5 Using additional circRNA annotations

From version `v1.0.2`, CIRI-long call also provide additional circRNA annotations in BED/GTF format for BSJ correction with `--circ` option. CircRNA annotations can be downloaded from circAtlas or other databases. The GTF-format output of CIRIquant is also supported.

NOTE: If using results from other tools/databases, please make sure the coordinate system is compatible with our CIRI-series tools:

> The coordinate system of circRNAs is different in most circRNA tools. For instance, if a circRNA is derived from chr1:1000-2000, it should be reported as chr1:1000-2000 in CIRI-series and some tools (DCC/KNIFE/Mapsplice), but reported as chr1:999-2000 in other tools (CIRCexplorer2/UROBORUS/circRNA_finder/find_circ).
>
> Thus, if you want to use circRNAs identified from tools in the latter group, you need to add 1 extra base to the start coordinate of circRNAs (the position with smaller coordinate regardless of the strand information), then use the altered coordinates as input.

27.3 Step2. isoform collapse

27.3.1 Basic Options

```
usage: CIRI-long collapse [-h] [-i LIST] [-o DIR] [-p PREFIX] [-r REF] [-a GTF] [--
↪canonical] [-t INT] [--debug]

optional arguments:
  -h, --help            show this help message and exit
  -i LIST, --in LIST    Input list of CIRI-long results
  -o DIR, --out DIR     Output directory, default: ./
  -p PREFIX, --prefix PREFIX
                        Output sample prefix, (default: CIRI-long)
  -r REF, --ref REF     Reference genome FASTA file
  -a GTF, --anno GTF    Genome reference gtf, (optional)
  -c CIRC, --circ CIRC  Additional circRNA annotation in bed/gtf format,
                        (optional)
  -t INT, --threads INT
                        Number of threads, (default: use all cores)
  --debug               Run in debugging mode, (default: False)
```

One should provide a text file listing sample name and path to CIRI-long output files `*.cand_circ.fa`, seperated by space.

```
sample1_name /path/to/sample1/cand_circ.fa
sample2_name /path/to/sample2/cand_circ.fa
```

27.3.2 Example Usage

For exmaple, you can create a file name `test.lst` with the following content:

```
test ./test_call/test.cand_circ.fa
```

Then run `CIRI-long collapse` to aggregate results from one or multiple samples.

```
 CIRI-long collapse -i ./test.lst \
                    -o ./test_collpase \
                    -p test \
                    -r ./mm10_chr12.fa \
                    -a ./mm10_chr12.gtf \
                    -t 8
```

27.3.3 Output Files

The output directory should have the following structure:

```
test_collpase
├── test_collpase.expression
├── test_collpase.isoforms
├── test_collpase.info
├── test_collpase.log
├── test_collpase.reads
└── tmp
    ├── ss.idx
    └── test_collpase.corrected.pkl

1 directory, 6 files
```

27.3.4 Output Format

The main output

The main output of CIRI-long is a GTF file (e.g. `test_collpase.info`), that contains detailed information of circRNAs and annotation of circRNA back-spliced regions in the attribute columns

Description of each columns's value

The attributes containing several pre-defined keys and values:

Expression matrix

- `test_collpase.expression` contains the summarized expression level of circRNAs in all samples in `tsv` format.
- `test_collpase.isoforms` contains the summarized isoform usage index of assembled isoforms in all samples in `tsv` format.
- `Isoform usage index = Isoform_reads / Sum of all isoforms from the same BSJ`

27.4 Step3. Output visualization

From version `v1.1.0`, CIRI-long included the 'misc/convert_bed.py', users can convert the GTF-formatted `circRNA.info` to bed format, and visualize using softwares like IGV / Jbrowse2

```
python3 misc/convert_bed.py collapse_out/sample.info sample_circ.bed
```

CHAPTER

TWENTYEIGHT

CIRI-LONG NANOPORE LIBRARY PREPARATION

28.1 1. Total RNA Extraction & Ribosomal RNA Depletion

- Total RNA is isolated using TRIzol (Invitrogen)
- RiboErase kit (human/mouse/rat, KAPA Biosystems) is used to remove rRNA from **1ug** of total RNA.
- Elute rRNA-depleted RNA from beads with 17uL nuclease free water.

28.2 2. Poly(A) Tailing & RNase R Treatment

Then, additional poly(A) tails are added to the linear transcripts to increase RNase R digestion efficiency.

28.2.1 2.1 Add Poly(A) Tails To Linear RNAs

E.coli Poly(A) Polymerase (NEBNext) is used to add poly(A) tails to the 3' end of linear RNAs, which can increase the RNase R digestion ability to RNAs with secondary structures.

- Add the following components in the order specified:
- Ribosomal-depleted total RNA is incubated at 37ºC with 1uL of *E.coli* Poly(A) Polymerase (NEBNext) for 30min.
- Stop the reaction by proceeding to the cleanup step.

28.2.2 2.2 Purification After Poly(A) Treatment

Agencourt RNAClean XP kit (Beckman) is used to remove contamination after poly(A) treatment.

- Add 2.2uL of Agencourt RNAClean XP per 1.0 uL of sample.
- Wash beads + RNA fragments twice with 75% Ethanol to remove contaminants.
- Elute purified RNA from beads with 20uL H2O.

28.2.3 2.3 RNase R Treatment To Effectively Digest Linear RNAs

Polyadenylted RNA is treated using RNase R (Epicentre) to remove linear RNAs.

- Polyadenylated RNA was incubated with RNase R at 37ºC for 15 min.

28.2.4 2.4 Purification after RNase R Treatment

2.2x bead-based cleanup is used to remove contamination after RNase R treatment as described above (See Part 2.2).

- Add 2.2uL of Agencourt RNAClean XP per 1.0 uL of sample.
- Wash beads + RNA fragments twice with 75% Ethanol to remove contaminants.
- Elute purified RNA with 5uL H2O.

28.3 3. SMARTer Reverse Transcription

Then, RNase R-treated RNA is reverse transcribed using random hexamers and SMARTer cDNA synthesis kit (Takara Bio) according to the manufacturer's instructions. The 3' SMART CDS Primer II A `5'-AAGCAGTGGTATCAACGCAGAGTACT(30)N-1N-3'` was replaced with `5'-AAGCAGTGGTATCAACGCAGAGTACNNNNNN-3'` to amplify circular RNAs without poly(A) sequences.

- Prepare reaction as follows:
- Incubated at 72ºC for 3min, 25ºC for 10min, hold at 42ºC.
- Add the following mixture:
- Incubation at 42ºC for 90 min.
- Denatured at 70ºC for 10 min.

28.4 4. cDNA PCR Amplification

To obtain sufficient cDNA products for sequencing, PCR amplification is performed using **2uL** of cDNA with NEB-Next LongAmp Taq 2x master mix and **SMARTer primers** under the following conditions:

28.5 5. Fragment Size Selection

Afterward, 0.5x Agencourt AMPure XP magnetic beads (Beckman) is used for size selection of the cDNA fragments:

- Add 0.5uL of AMPure XP per 1.0 uL of sample.
- Wash beads + cDNA fragments twice with 75% Ethanol to remove contaminants.
- Elute purified cDNA with 20-30uL H2O.

28.6 6. Nanopore Sequencing

Finally, cDNA libraries are prepared according to the ONT protocol `SQK-LSK109` and barcoded with `EXP-NBD104` / `EXP-NBD114` kits, and nanopore sequencing is performed using the MinION (MN26543) platform with a `FLOW-MIN106` flow cell. Please refer to the Nanopore Community for detailed instructions.

CHAPTER

TWENTYNINE

CIRCATLAS V2.0

circAtlas v2.0: An integrated resource of circRNAs in vertebrates.

Circular RNAs (circRNAs) are a novel class of RNAs with important biological implications. Currently, a huge number of circRNAs was already identified from high-throughput RNA-seq data sets. However, functional annotation and prioritization of these circRNAs for further expreimental validation as well as functional investigation is the bottleneck step for current circRNA studies. Now, we developed a new resource database and web tool named circAtlas 2.0 that integrated over one millions of circRNAs across 6 species (human, macaca, mouse, rat, pig, chicken) and a variety of tissues. The circAtlas 2.0 can fill this gap by integrating the most comprehensive circRNAs and their expression and functional profiles in vertebrates, which provides a foundation for circRNA studies and serves as a powerful starting point to investigate their biological significance.

Please refer to http://circatlas.biols.ac.cn/tutorial for detailed information.