CircHunter Manual

Carlo De Intinis

July 15, 2017

0 CONTENTS

1 System requirements **2**
1.1 Docker installation 2
1.2 Giving docker non-root access 2
1.3 CircHunter installation 2
1.4 Launching CircHunter 2

2 Input data **2**
2.1 Circular RNA file 2
2.2 Genome file 3
2.3 Isoform data file 3

3 CLI usage **4**
3.1 Execution modes 4
3.2 Arguments 5
3.3 Extra functions 7

4 Syntax examples **7**

5 Graphical User Interface **7**

6 Toy Example **8**

1 System requirements

1.1 Docker installation

CircHunter requires the free version of the program Docker installed on the system. Depending on the user's installed Linux distribution, the installation process may be different. Please refer to the *Get Docker* section of the Docker website (www.docker.com).

1.2 Giving docker non-root access

When docker is installed, it will be required to be given non-root access.
To do so, a docker group must be created (if it does not already exist):

```
sudo groupadd docker
```

Then, the connected user "$USER" is added to the docker group.

```
sudo gpasswd -a $USER docker
```

Run `newgrp docker` or log out and log in to activate the changes to groups. To check if the procedure was succesful, the command `docker run hello-world` should be executed.

1.3 CircHunter installation

The CircHunter software suite is contained inside a Docker repository that can be obtained running the following command:

```
docker pull carlodeintinis/circhunter:latest
```

1.4 Launching CircHunter

When the installation process is complete, it is possible to use the Command Line Interface (CLI) to execute the provided launcher script (circhunter.sh) with all the required arguments for the analysis. Alternatively, the CLI can be used to launch the CircHunter graphical interface contained in the provided jar file (CircHunterGUI.jar).

The CircHunter launcher script and graphical user interface can be downloaded from the GitHub repository https://github.com/carlo-deintinis/circhunter.
On a system with git installed it is possible to download the repository by entering the command `git clone https://github.com/carlo-deintinis/circhunter`.

2 Input data

In this section, the requirements of input files to use with CircHunter are described in detail. However, as a general rule, all files must be in plain text (no encoded files).

2.1 Circular RNA file

The file containing circular RNA information must be composed of the following TAB separated columns:

- chromosome name specified as chr## (i.e. chr1, chr12)
- start position
- end position
- circRNA name specified as chr##_start_end (i.e. chr1_10_100)
- genomic strand specified as 1 (forward) or -1 (reverse)

Every row of the file is used for a single circRNA, with EOL characters to distinguish circRNAs from one another.

2.2 GENOME FILE

Three main genomes are officially supported by CircHunter: hg18, hg19 and hg38. Of these, **hg19** comes preinstalled, the others are downloaded by the program when they are requested via the argument *-as*.

A downloaded genome or a not supported genome may be provided via the argument *-sg*, however please note that CircHunter default behaviour is to remove every genome entry that does not comply with the accepted chromosome names (from chr1 to chr22, chrX and chrY).

All required genome data is shown in Table 1.

Label	Description
ENSG	Ensembl Gene ID
ENST	Ensembl Transcript ID
ENSE	Ensembl Exon ID
Chr	Chromosome name
Exon chr start	Start position of exon
Exon chr end	End position of exon
Strand	Strand of exon (1 or -1)
Exon Rank in Transcript	Rank of the exon
Gene start	Start position of exon's gene
Gene end	End position of exon's gene
Transcript start	Start position of exon's transcript
Transcript end	End position of exon's transcript

Table 1: Genome exon export data requirements. All fields must be TAB separated.

2.3 ISOFORM DATA FILE

The isoform data file is used by CircHunter to obtain a unique classification of circRNAs. If this file is not provided, CircHunter will download the required data using an available internet connection. All required data is shown in Table 2.

Label	Description
ENSG	Ensembl Gene ID
ENST	Ensembl Transcript ID
Chr	Chromosome name
External transcript name	Isoform identifier

Table 2: Isoform data file requirements. All fields must be TAB separated.

3 CLI USAGE

CircHunter is launched via the script **circhunter.sh**, to which all the arguments for the execution of the analysis must be provided. The user must specify an *execution mode* to select the type of analysis and some *arguments* that are used to provide file locations and analysis options.

3.1 EXECUTION MODES

CircHunter relies on four main execution modes that can be invoked through specific flags. These modes are described below.

-c, - -classification **circRNA classification**

Requires: -cr. *Optional: -sg, -as, -id, -of*

CircHunter will output a transcript-wise classification of the provided circRNAs. Outputs a file named *circRNA_classification* where every row represents a transcript associated with a circRNA and consequently the classification of that circRNA referring to the transcript. This mode also outputs a unique circRNA classification: every cirRNA is assigned to one particular classification, in order of importance. Refer to Section **??** for additional information regarding the classification process.

-s, - -sequences **circRNA backsplicing junction sequences**

Requires: -cr *Optional: -sg, -as, -of*

CircHunter outputs the backsplicing junction sequences of all the provided circRNAs. The sequences' span is of +70/-70bp and all sequences are obtained considering the full genomic sequence (intronic sequences are not removed). Sequences are outputted in a FASTA format file named *circRNA_backsplicing_sequences.fasta*.

-r, - -readcount **read count of backsplicing junction sequences**
Requires: -rs, -bj Optional: -hc, -of
The HashCirc suite is executed. The RNA-Seq data is used in conjunction with the backsplicing junction sequences provided in order to obtain a count of the RNA-Seq reads that represent the backsplicing junction sequences. Of the two steps required, the first one may be parallelized modifying the appropriate *-hc* value. Results are saved in multiple files named *readcount*i where *i* is the part number (determined by the parallelization process).

-f, - -full **full**
Requires: -cr Optional: -sg, -as, -id, -of
CircHunter will execute the **circRNA classification** execution mode followed by the **circRNA backsplicing junction sequences** mode, allowing the user to obtain both results in a single step. Output files from both execution modes are obtained, with the same names specified above.

3.2 ARGUMENTS

In addition to the execution mode, CircHunter can use the arguments described in this section. Please refer to Section 2 for additional information regarding required file contents and formats.

-h, - -help **help**

Shows a quick help screen in the CLI.

-cr, - -circrna **supplied circRNA list file**

File containing the coordinates of cirRNAs to investigate.

-sg, - -suppliedgenome **supplied exon export file**
Supported genomes: hg18, hg19, hg38
File containing a Biomart exon export. CircHunter will filter out any export referring to non-canonical chromosome names. A specific set of genes instead of a full genome can be supplied. Please note that the hg19 genome comes preinstalled within the CircHunter's Docker environment, so it is not necessary to supply it if a genome-wide research is the researcher's goal.

-id, - -isoformdata **supplied gene isoform data**
Supported genomes: hg18, hg19, hg38
File containing the isoform data required for the circRNA univocal classification process. If this data is not provided by the user, CircHunter will download it.

-rs, - -rnaseq **supplied RNA-Seq file**

RNA-Seq file supplied by the researcher. There is no need to split the file for parallelization purposes, this can be accomplished with the *-hc* argument.

-bj, - -bksjunctions **supplied circRNA backsplicing junction sequences**

File containing the sequence that represents the circRNA backsplicing junctions. These sequences can be obtained from execution mode *-s*.

-of, - -outputfolder **supplied output folder**
Defaults to: ∼/CircHunter/data
Use this argument to specify a custom output folder.

-as, - -assembly **genome assembly to use for backsplicing sequences**
Available choices: hg18, hg19 (default), hg38
CircHunter can use three different genomes. Of these, hg19 is the default and it is selected whenever *-as* is missing. Also, hg19 is already installed in CircHunter's Docker environment, hg18 and hg38 can be provided by the user with the *-sg* argument or can be downloaded autonomously by CircHunter during the analysis process. The download can require entire minutes, depending on the speed provided by the Biomart servers and the user connection.

-hc, - -hashcirc **arguments to pass to hashcirc**
Syntax: -hc 1 2 3 4 5 6
Various arguments for the HashCirc step can be specified by the user. These arguments are listed below.

1. *k*-mer size • *k*-mer length (bp)
2. thread number • Number of processes to parallelize (limited by CPU logic core number).
3. hash size • Dimensions of the hash table
4. collision list size • Dimensions of the collision list
5. *k*-mer number • Number of *k*-mers that must be matched to the sequence to consider the sequence itself as represented in the RNA-Seq data.
6. matches • Number of perfect matches required in the *k*-mer to consider it matched to a sequence.

3.3 EXTRA FUNCTIONS

-cd, - -cleandocker **Clean docker images**

Used to recover disk space

Every time CircHunter is launched, a Docker image is saved on the system. To recover the lost disk space, run this function with the following syntax: `bash circhunter.sh -cd`.

4 SYNTAX EXAMPLES

`bash circhunter.sh -c -cr circRNA_file -as hg19`

CircHunter will execute **- -classification** mode and classify every circRNA in the *circRNA_file* that was provided by the user. The program will use the preinstalled *hg19* genome. Isoform data and exon information are downloaded using an available internet connection.

`bash circhunter.sh -c -cr circRNA_file -sg genome_file -as hg19 -id isoform_data_file`

Same as above, but CircHunter uses the provided genome file and isoform data file instead of downloading the data.

`bash circhunter.sh -s -cr circRNA_file -as hg19 -of path/outputfolder`

CircHunter executes **- -sequences** mode and obtains the backsplicing junction sequences of all circRNAs provided with the *circRNA_file*. A output folder has been provided with the **-of** argument (*path/outputfolder*).

`bash circhunter.sh -f -cr circRNA_file -as hg38 -sg genome_file`

CircHunter executes **- -full** mode: the provided *circRNA_file* in conjunction with the provided genome (*genome_file*) are used to classify circRNAs and obtain the backsplicing junction sequences of the circRNAs.

`bash circhunter.sh -r -rs rnaseq_file -bj backsplicing_junction_file -hc 27 3 100 10 5 6`

CircHunter will execute **- -rnaseq** mode and counts of the reads representing the provided circRNA backsplicing junctions will be obtained. The arguments necessary to hashcirc execution have been provided with the **-hc** argument.

5 GRAPHICAL USER INTERFACE

CircHunter presents a GUI that can be used to set an analysis process with ease. The GUI is packed in the file **CircHunterGUI.jar** and needs a java installation in order to run.

The gui can be launched via command line by moving to the folder containing the GUI file and entering the command `java -jar "CircHunterGUI.jar"`.

Alternatively the GUI can be launched from a file manager, but the file must be marked as executable by the user. This is done by entering the command `chmod+x CircHunterGUI.jar`.

6 TOY EXAMPLE

CircHunter presents a toy example to test the correct execution of the program on the local machine. Figure 1 shows a graphical representation of the fake circRNAs and genome features that are encoded in the example.

The example can be run by the user by entering the command below in the CLI. Please note that this command is valid in the main folder of CircHunter, otherwise the file paths must be adjusted by the user

```
bash circhunter.sh -f -cr toyexample/toy_circRNA -sg toyexample/toy_genome -id
toyexample/toy_isoformdata -as hg19
```

Table 3 shows the circRNAs included in the toy example, whereas Table 4 shows the expected result of the univocal classification process.

chromosome	start pos.	end pos.	circRNA name	strand
chr1	10	60	chr1_10_60	1
chr1	70	90	chr1_70_90	1
chr1	70	120	chr1_70_120	-1
chr1	100	160	chr1_100_160	1
chr1	135	190	chr1_135_190	1
chr1	165	195	chr1_165_195	1

Table 3: circRNAs included in the toy example.

circRNA name	classification	isoform	isoform rank
chr1_100_160	intergenic	Isoform-001	001
chr1_10_60	multiexon	Isoform-001	001
chr1_135_190	intronic	OtherIsoform-001	001
chr1_165_195	intronic	OtherIsoform-001	001
chr1_70_120	multiexon	Isoform-201	201
chr1_70_90	monoexon	Isoform-001	001

Table 4: Expected results of the **-f** run of the toy example data

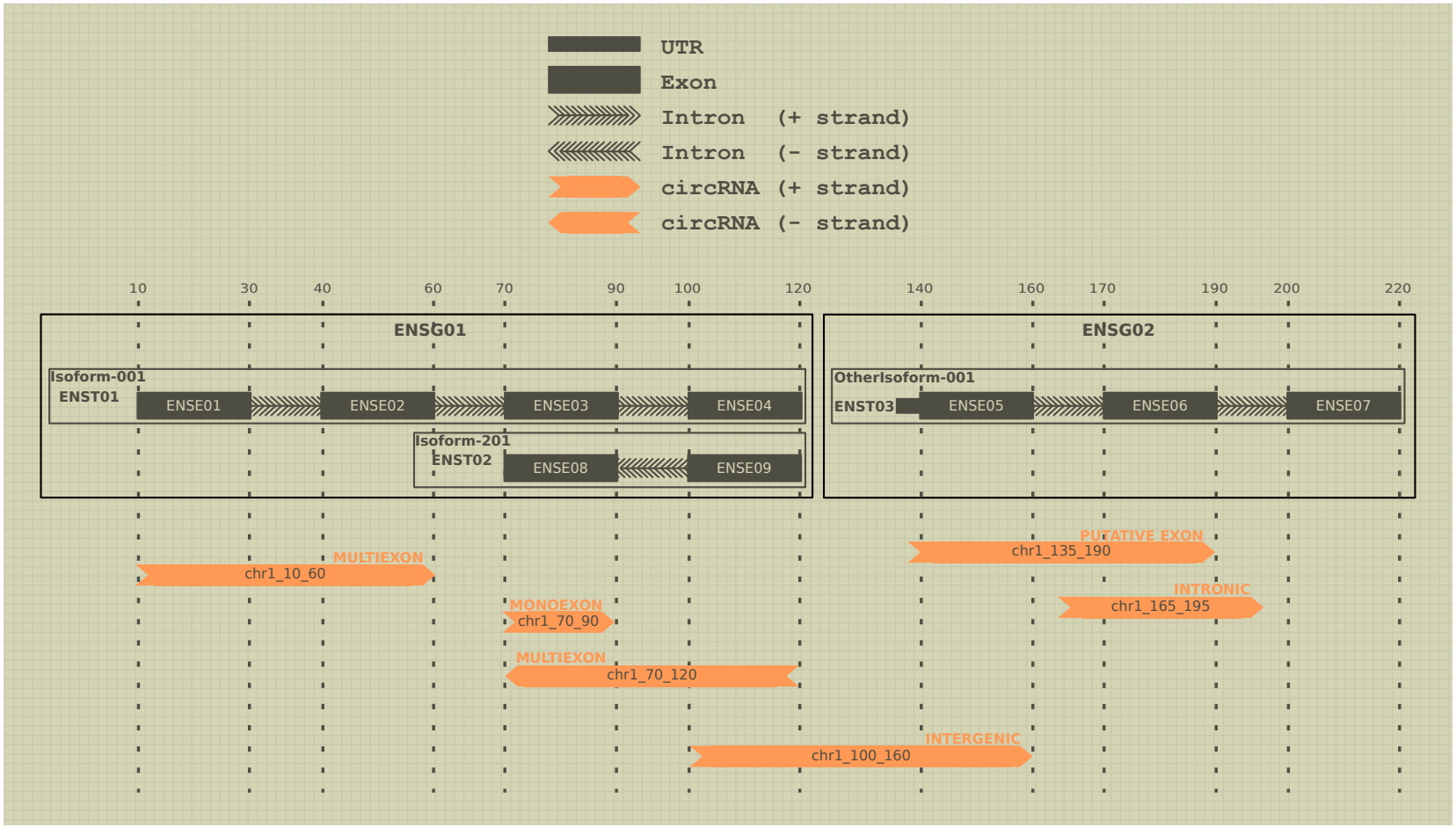

Figure 1: Representation of the example genome represented in the toy example files. The classification noted on the figure is the result of the univocal classification process of CircHunter. Note that the program is strand-aware and a circRNA won't align on the opposite strand.