RnaChipIntegrator Documentation

Release 2.0.0

Peter Briggs, Ian Donaldson, Leo Zeef

Jan 06, 2020

Contents

`RnaChipIntegrator` is a Python bioinformatics utility that performs integrated analyses of 'gene' data (a set of genes or genomic features, for example gene expression data or canonical gene lists) with 'peak' data (a set of regions, for example ChIP peaks) to identify the nearest genes or features to each peak, and vice versa.

CHAPTER 1

What it does

`RnaChipIntegrator` was designed to integrate genes/transcripts from expression analysis (RNA-seq, microarrays) with ChIP-seq binding regions, however it is flexible enough to allow the comparison of any genome coordinate based data sets.

`RnaChipIntegrator` answers the questions:

- "Which genes are close to each of my ChIP-seq regions?", and
- "Which ChIP-seq regions are close to each of my genes?".

The first data set, called **'genes'**, is strand specific and the genome coordinates correspond to the transcription start site (TSS) and transcription end site (TES), depending on the strand.

Note: For strand and genome coordinates:

- The start coordinate of a gene on the forward or '+' strand relates to the TSS;
- The start coordinate of a gene on the reverse or '-' strand relates to the TES.

This is primarily gene or transcript annotation for the whole genome. However, other non-gene features, such as CpG islands, can be used.

The second data set we call **'peaks'** are strand non-specific, including only the start and end coordinate. This is primarily the coordinates of ChIP-seq binding regions (a.k.a. peaks).

See the *Input files* section for more information about the input file formats.

Example use cases ('gene' versus 'peak') include:

- RNA-seq expressed genes versus ChIP-seq binding regions
- Microarray expressed genes versus ChIP-chip binding regions
- Total gene annotation versus ChIP-seq binding regions
- Gene promoters versus CpG island annotation

CHAPTER 2

Getting started

The easiest way to get the latest version of `RnaChipIntegrator` is to use Python's `pip` utility to install the latest version of the program directly from the Python Package Index (PyPI), by doing:

```
pip install RnaChipIntegrator
```

Note: You may need to have root privileges to install to the system directories, in which case preface this command with `sudo` i.e.:

```
sudo pip install RnaChipIntegrator
```

or you can do:

```
pip install --user RnaChipIntegrator
```

to install it under your home area.

Alternatively you can use Python's `virtualenv` mechanism to install a non-root version (this example creates one under `.venv`):

```
virtualenv .venv; . .venv/bin/activate
pip install RnaChipIntegrator
```

Note: For an introduction to `pip` and `virtualenv`, see for example:

- http://www.dabapps.com/blog/introduction-to-pip-and-virtualenv-python/
- https://www.biostars.org/p/109179/

To update an existing version of the program to a newer one, use:

```
pip install -U RnaChipIntegrator
```

For other ways of installing please refer to the `INSTALL` document included with the distribution.

CHAPTER 3

Usage

3.1 Simple usage

The easiest form of usage is:

```
RnaChipIntegrator GENES PEAKS
```

where `GENES` and `PEAKS` are tab-delimited files containing the gene and peak data respectively (see *Input files* for details of these files).

This will produce two output files:

- `GENES_peak_centric.txt`: reports the nearest genes for each peak ('peak-centric' analysis)
- `GENES_gene_centric.txt`: reports the nearest peaks for each gene ('gene-centric' analysis)

In both cases the files will contain one peak/gene pair per line (see *Output files* for details of these files).

The program has various options that can be applied to control the analyses that are performed and the outputs from each run, as outlined in the following sections.

3.2 Specifying distance cutoff (`--cutoff`)

The `--cutoff` option specifies a maximum distance in bp that a gene/peak pair can be apart and still be included in the analyses; gene/peak pairs which are further apart than this distance will not be reported.

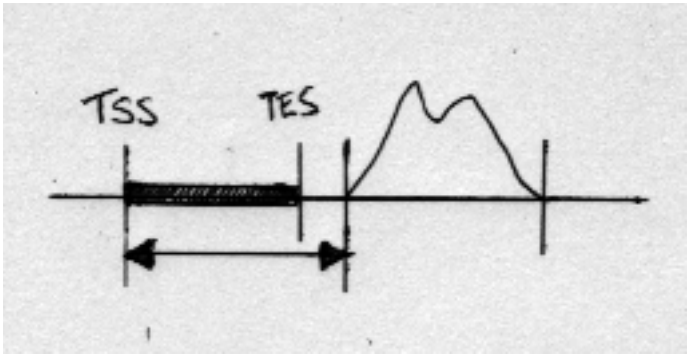
For example:

```
RnaChipIntegrator --cutoff=130000 GENES PEAKS
```

Note: If a maximum cutoff distance is not explicitly specified then the default is 1000000 bp. Set the distance to 0 to turn off the cutoff limit and include all pairs regardless of distance.

3.3 Specifying how distances are measured between peaks and genes (--edge)

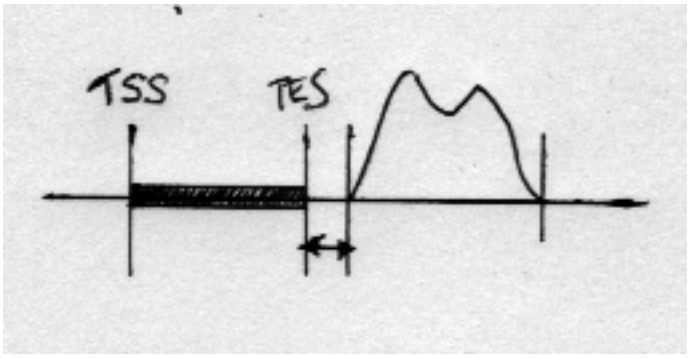
By default the distance between a peak and a gene is calculated as the distance from the gene TSS to the nearest peak edge, for example:



Alternatively distances can be calculated as the shortest distance between either of the peak edges to either the TSS or the TES of the gene, by specifying the `--edge=both` option:

```
RnaChipIntegrator --edge=both GENES PEAKS
```

For example for the same arrangement as above this would generate a much smaller closest distance:



Note: Using `--edge=both` essentially makes the analyses 'strand-agnostic'.

3.4 Only using differentially expressed genes (--only-DE)

If the input genes data contains a differential expression flag (see *'Genes' data file*) then this can be used in the analysis by turning on the `--only-DE` option:

```
RnaChipIntegrator --only-DE GENES PEAKS
```

which will only included the flagged genes in the analyses.

Note: Without the `--only-DE` option, all genes will be used regardless of the presence of a differential expression flag.

3.5 Limiting the number of results to report (--number)

By default, all gene/peak pairs that are located within the specified cut-off distance (see *Specifying distance cutoff (–cutoff)*) will be reported in the output files.

To restrict the maximum number of pairs that are reported per gene or peak use the `--number` to specify a limit. Even if more pairs are found, only this number of pairs will be output.

Warning: Be aware that if used, this number limit is applied rigidly. For example, even if the fourth and fifth gene/peak pairs both have the same distance separation then using `--number=4` will only include the first of these and reject the second.

3.6 Specifying the promoter region (--promoter_region)

As part of its peak-centric analyses, for each peak/gene pair `RnaChipIntegrator` reports whether the peak overlaps the promoter region of the gene.

By default, within the program the promoter region of a gene is defined as starting 1000 bp upstream of the gene TSS and ending 100 bp downstream of the TSS.

The `--promoter_region` option can be used to define a different set of limits for this region, using the general format:

```
--promoter_region=UPSTREAM,DOWNSTREAM
```

For example:

```
--promoter_region=1500,200
```

would define a promoter region starting 1500 bp upstream of the TSS and ending 200 bp downstream.

3.7 Running either peak-centric or gene-centric analysis only (--analyses)

By default `RnaChipIntegrator` runs both peak-centric and gene-centric analyses.

However it is possible to restrict the program to just one or other of these, by using the `--analyses` option.

For example to run only the peak-centric analyses:

```
--analyses=peak_centric
```

Or, to run only the gene-centric analyses:

```
--analyses=gene_centric
```

The advantage of restricting the analyses is that it reduces the program run time, and limits the outputs to only those specifically requested.

3.8 Specifying multiple distance cutoffs (--cutoffs)

RnaChipIntegrator can peform its analyses over multiple cutoff distances by using the --cutoffs option to supply a comma-separated list of distances, for example:

```
RnaChipIntegrator --cutoffs=50000,100000,150000 GENES PEAKS
```

The selected analyses will be repeated for each of the specified cutoff distances, and the distance will be reported as an additional field for each gene/peak pair in the output files (see *Additional fields for batch operation*).

Note that --cutoffs is an alternative to the --cutoff option and the two cannot be used together.

Note: This option can be used along with --peaks and genes (see *Specifying multiple peaks and/or genes files (–peaks and –genes)*), to apply several cutoff distances to analyses of multiple peaks and/or genes files.

3.9 Specifying multiple peaks and/or genes files (--peaks and --genes)

In normal operation RnaChipIntegrator operates on a single pair of files specifying the gene and peak data.

However it can also operate on multiple peaks and/or genes files within a single run, by using the --peaks and --genes options.

For example, to analyse a pair of genes sets against the same set of peaks:

```
RnaChipIntegrator --genes GENES1 GENES2 --peak PEAKS
```

which would result in the program performing two analyses (i.e. GENES1 versus PEAKS and GENES2 versus PEAKS).

Analysing several sets of peaks against a single set of genes would look like:

```
RnaChipIntegrator --genes GENES --peak PEAKS1 PEAKS2 PEAKS3
```

which would result in the program performing three analyses (i.e. GENES versus PEAKS1, PEAKS2 and PEAKS3).

Analysing multiple sets of genes against multiple sets of peaks would look like:

```
RnaChipIntegrator --genes GENES1 GENES2 --peak PEAKS1 PEAKS2 PEAKS3
```

This would result in the program performing six analyses (i.e. GENES1 versus PEAKS1, PEAKS2 and PEAKS3 then GENES2 versus the three peaks files).

Note that --peaks and --genes must always be used together, and instead of specifying a single pair of files at the end of the command line.

In all cases where there is more than one file then the name of the appropriate file(s) will be reported as an additional field for each gene/peak pair in the output files (see *Additional fields for batch operation*).

Note: These options can be used along with --cutoffs (see *Specifying multiple distance cutoffs (–cutoffs)*), to repeat each set of analyses at various cutoff distances.

3.10 Specifying multiple cores in batch modes (--nprocessors)

RnaChipIntegrator can use multiple cores in 'batch' modes (that is, any run which performs more than one analysis because multiple distance cutoffs and/or multiple peaks or genes files were specified on the command line).

In these modes the number of cores to use can be supplied via the --nprocessors option, for example:

```
RnaChipIntegrator --cutoffs=50000,100000,150000 --nprocessors=2 GENES PEAKS
```

3.11 Changing the output files and formats

There are a number of options to produce additional output files, and to modify the format and output content depending on requirements:

- *Excel spreadsheet (–xlsx)*
- *Summary files (–summary)*
- *Compact output format (–compact)*
- *Output padding (–pad)*
- *Specifying feature type other than 'gene' etc (–feature)*
- *Specifying an ID for input peaks (–peak_id)*

3.12 Using RnaChipIntegrator in Galaxy

In addition to the command-line version, we have also provided a tool which allows RnaChipIntegrator to be run within the popular Galaxy bioinformatics platform:

- https://toolshed.g2.bx.psu.edu/view/pjbriggs/rnachipintegrator/

The tool can be installed into a local instance of Galaxy directly from the Galaxy Toolshed

See the documentation at http://getgalaxy.org/ on how to get a local Galaxy up and running, and how to install tools from the Toolshed.

CHAPTER 4

Input files

RnaChipIntegrator expects two input files: a list of genes and a list of peaks.

4.1 'Genes' data file

The 'genes' data file must be a tab-delimited file with at least 5 columns of data for each gene or genomic feature (one per line):

```
ID  chr  start  end  strand
```

where:

- chr is chromosome the gene appears on
- start and end define the limits of the gene
- strand is the strand direction (either + or −)
- ID is a name which is used to identify the gene in the output.

Optionally there can be a sixth column:

```
ID  chr  start  end  strand  DE_flag
```

If DE_flag is present then it can be used to indicate whether the gene should be considered to be differentially expressed (DE_flag = 1) or not (DE_flag = 0); see *Only using differentially expressed genes (–only-DE)*.

Note that any additional columns are ignored.

Note that lines in the input file are ignored in the following cases:

- Line starts with the hash character # (considered to be a comment or header line)
- First line has non-integer values for start and end, or an invalid value for the strand (considered to a header line)

The following are critical errors which will cause the program to terminate prematurely:

- Line has values in either the `start` or `end` columns which aren't integers, or a value in the `strand` column which isn't either a + or − character (except if it's the first line in the file)
- Line has a `start` value which is greater than the `end` value
- Line doesn't contain at least five columns.

The program issues a warning for each problem line that it encounters.

4.2 'Peaks' data file

The 'peaks' data file must be a tab-delimited file with at least 3 columns of data for each peak (one per line). By default the first 3 columns should be:

```
chrom  start  end
```

where:

- `chrom` is the chromosome that the peak appears on
- `start` and `end` define the limits of the peak region

Warning: `start` and `end` positions must differ by at least 1bp, and the `end` must come after the `start`.

Any additional columns found in the file are ignored (unless the `--peak_id` option is used to specify an additional column with names to associate with each peak - see *Specifying an ID for input peaks (–peak_id)*.)

Note that lines in the input file are ignored in any of the following cases:

- Line starts with the hash character # (considered to be a comment line)
- Line has values in either the `start` or `end` columns which aren't integers
- Line doesn't contain at least three columns.

The program issues a warning for each line that is skipped.

Note: In previous versions of `RnaChipIntegrator` a distinction was made between peak 'regions' and peak 'summits', depending on whether the `start` and `end` positions defined a region of width 1 (i.e. a summit) or greater than 1 (i.e. a region).

For this version of the program no distinction is made and the same analyses are performed regardless of whether the data define summits or regions.

Note: The `--peak_cols` option can be used to specify an arbitrary set of three columns to use for the chromosome and start and end positions. For example:

```
--peak_cols=2,4,5
```

will use the values from the 2nd, 4th and 5th columns for `chrom`, `start` and `end` respectively.

CHAPTER 5

Output files

5.1 Overview

The default output of the program consists of a pair of tab-delimited files:

- **<BASENAME>_peak_centric.txt**
  shows all of the genes associated with each peak ('peak-centric' analysis)
- **<BASENAME>_gene_centric.txt**
  shows all of the peaks associated with each gene ('gene-centric' analysis)

By default the output file `BASENAME` is taken from the name of the input 'genes' file; use the `--name` option to set a custom basename.

Additional files may be produced depending on the options that have been specified on the command line.

The format and content of each file is described in the following sections.

5.2 Genes associated with each peak ('peak-centric' output)

By default the 'peak-centric' output file has one line for each peak/gene pair that is being reported, with the following columns of data for each:

Name	Description
peak.id	(Optional) peak ID (if `--peak_id` option was used)
peak.chr	chromosome of the peak
peak.start	peak start position
peak.end	peak end position
gene.id	gene ID
strand	gene strand direction
TSS	gene TSS position
TES	gene TES position
dist_closest	closest distance between peak and gene considering all edges (zero if there is overlap)
dist_TSS	distance between peak and gene TSS
dist_TES	distance between peak and gene TES
direction	'U' if peak is upstream (5') of gene; 'D' if peak is downstream (3') of gene; '.' if overlapping
overlap_gene	1 if peak overlaps the gene, 0 if not
overlap_promoter	1 if peak overlaps the promoter region, 0 if not

Each peak will appear as many times as there are nearest genes being reported for that peak. For example:

```
...
chr1  9619046  9619167  NM_178399_3110035E14Rik  +  9591248  9617222  ...
chr1  9619046  9619167  NM_008651_Mybl1          -  9690280  9635825  ...
chr1  9619046  9619167  NM_175236_Adhfe1         +  9538049  9570746  ...
chr1  9619046  9619167  NM_021511_Rrs1           +  9535513  9537532  ...
...
```

If there are no closest genes for a peak (based on the distance cutoff) then the peak will still be reported but the remainder of the fields will be filled with placeholders:

```
chr17  23681171  23681172  ---  ---  ---  --- ...
```

Use the `--compact` option to output all the genes for each peak on a single line (*Compact output format (–compact)*).

5.3 Peaks associated with each gene ('gene-centric' output)

By default the 'gene-centric' file has one line for each gene/peak pair that is being reported, with the following columns of data for each:

Name	Description
gene.id	gene ID
gene.chr	chromosome of the gene
gene.start	gene start position
gene.end	gene end position
gene.strand	gene strand direction
peak.id	(Optional) peak ID (if `--peak_id` option was used)
peak.chr	chromosome of the peak
peak.start	peak start position
peak.end	peak end position
dist_closest	closest distance between peak and gene considering all edges (zero if there is overlap)
dist_TSS	distance between peak and gene TSS
direction	'U' if gene is upstream (5') of peak; 'D' if gene is downstream (3') of peak; '.' if overlapping
in_the_gene	'YES' if peak overlaps the gene, 'NO' if not

Each gene will appear as many times as there are nearest peaks being reported for that gene. For example:

```
...
BC021773_Glb1l  chr1  75193364  75207353  -  chr1  75481920  75482054  ...
BC021773_Glb1l  chr1  75193364  75207353  -  chr1  75481920  75482054  ...
...
```

If there are no closest peaks to a gene (based on the distance cutoff) then the gene will still be reported but the remainder of the fields will be filled with placeholders:

```
BC028767_3110009E18Rik  chr1  122017764  122114603  +  ---  ---  --- ...
```

Use the `--compact` option to output all the peaks for each genes on a single line (see *Compact output format (–compact)*).

5.4 Summary files (`--summary`)

Using the `--summary` option outputs an additional pair of tab-delimited files:

- `BASENAME_peak_centric_summary.txt`
- `BASENAME_gene_centric_summary.txt`

These will only contain the 'top' (i.e. closest) gene/peak pairs, with the same columns of data as the 'full' versions of the files.

5.5 Excel spreadsheet (`--xlsx`)

Using the `--xlsx` option outputs an additional Excel spreadsheet file `BASENAME.xlsx`, which contains the results from all the tab-delimited files (including the summaries, if `--summary` was also specified), plus a 'notes' sheet with additional information about the results from each analysis.

5.6 Compact output format (`--compact`)

By default each gene/peak pair will be output on a separate line, for example:

```
#chr   start    end      gene.id     strand  TSS      TES      dist_closest dist_TSS␣
↪dist_TES  overlap_gene  overlap_promoter
chr2R  4959563  4959564  CG8084-RA   +       4956606  4965060  0            2957    ␣
↪5496      1             0
chr2R  4959563  4959564  CG8193-RA   -       4932214  4929765  27349        27349   ␣
↪29798     0             0
chr3R  12882217 12882218 CG3937-RB   -       12921260 12917257 35039        39042   ␣
↪35039     0             0
...
```

Specifying the `--compact` option changes the ouput so that all the genes closest to each peak (and vice versa) are written on a single line, for example:

```
#chr   start    end      gene.id_1  gene.id_2  gene.id_3  gene.id_4
chr2R  4959563  4959564  CG8084-RA  CG8193-RA
chr3R  12882217 12882218 CG3937-RB
```

Warning: `--compact` is not compatible with `--summary`.

5.7 Output padding (`--pad`)

If the `--pad` option is specified then where fewer than the maximum number of pairs would be reported, additional 'blank' lines are inserted to make up the number of lines to the maximum.

For example:

```
#chr   start    end      gene.id     strand  TSS      TES      dist_closest dist_TSS
↪dist_TES  overlap_gene     overlap_promoter
chr2R  4959563  4959564  CG8084-RA   +       4956606  4965060  0            2957
↪5496      1                0
chr2R  4959563  4959564  CG8193-RA   -       4932214  4929765  27349        27349
↪29798     0                0
chr2R  4959563  4959564  ---         ---     ---      ---      ---          ---
↪--        ---              ---
chr2R  4959563  4959564  ---         ---     ---      ---      ---          ---
↪--        ---              ---
```

5.8 Specifying feature type other than 'gene' etc (`--feature`)

By default the program uses the term 'gene' in its outputs regardless of the nature of the genomic features being examined. This term can be changed to refer to a different feature type by using the `--feature` option.

For example:

```
--feature=transcript
```

in which case the word 'gene' will be replaced by 'transcript' in output headers and so on.

Note: The feature type is purely cosmetic and has no effect on the input or output file formats, or the analyses performed.

5.9 Specifying an ID for input peaks (`--peak_id`)

If the `--peak_id` option is specified on the command line then this indicates a column in the input peaks file which should be used as names for each of the peaks in that file.

For example, if the input peaks file looks like:

```
#Chrom      Start    End      Name
chr1        9619046  9619167  P0001
chr1        9619175  9619382  P0002
chr1        10617233         10617437        P0003
...
```

then using:

```
--peak_id=4
```

will associate the names `P0001`, `P0002`, `P0003...` with the corresponding peaks.

When specified, this ID is carried through to the output file as an additional field, for example:

```
...
P0001       chr1    9619046 9619167 NM_021511_Rrs1  +       9535513 9537532 81514
↪83533   81514   D       0       0
P0002       chr1    9619175 9619382 NM_178399_3110035E14Rik +       9591248
↪96172221953     27927   1953    D       0       0
...
```

5.10 Writing results to separate files in batch mode (`--split-outputs`)

By default in 'batch' mode (i.e. when multiple cutoff distances and/or multiple peak or genes files are supplied) all results for the gene-centric analyses will be written to a single file (and similarly for the peak-centric analyses).

To force `RnaChipIntegrator` to write the results of each batch to a separate file, use the `--split-outputs` option. When this option is specified a set of files will be generated for each peak, gene and cutoff with appropriate names to indicate which files and cutoff were used.

5.11 Additional fields for batch operation

When `RnaChipIntegrator` is run in 'batch' mode (that is, any mode where multiple cutoffs have been supplied via the `--cutoffs` option, and/or multiple input files have been supplied the `--peaks` and `genes` options), extra fields will be added for each reported peak/gene pair to distinguish which analysis the result came from:

Name	Description
peak_file	Source file for the peak (if more than one peaks file was supplied via `--peaks`)
gene_file	Source file for the gene (if more than one genes file was supplied via `--genes`)
cutoff	maximum cutoff distance (if more than one cutoff distance was supplied via `--cutoffs`)

For example:

```
#peak_file      gene_file       cutoff  gene.id         gene.chr gene.start gene.
↪end gene.strand ...
/data/peaks1.txt /data/genes1.txt 50000   AF064749_Col6a3 chr1     92566771
↪92800755 -           ...
```

See the *Specifying multiple distance cutoffs (–cutoffs)* and *Specifying multiple peaks and/or genes files (–peaks and –genes)* sections for more information on these options.

Note: Each of the additional fields will only appear if it is required in order to distinguish between the different analyses. For example, `cutoff` will only appear if more than one maxium cutoff distance was supplied.

5.12 Interpreting 'upstream' and 'downstream'

One of the attributes reported for each peak/gene pair found in the analyses is the 'directionality' (in the `direction` column), which can be either 'upstream' (`U`), 'downstream' (`D`) or overlapped.

The intepretation of 'upstream' and 'downstream' for a given pairing depends on the 'centricity' of the analysis and the strand direction.

For peak-centric analyses, the direction is from the point of view of the peak:

```
                    ---Downstream-->    <---Upstream---

+ strand:  5' |----Gene1-------------Peak-------------Gene2----> 3'
```

In the example above, the peak is downstream of `Gene1` and upstream of `Gene2`.

(An analogy is that of a river which flows from the 5' to the 3' end; the 'downstream' direction is the direction of flow from start to end, while the 'upstream' direction is the opposite, from end to start.)

For the - strand this is reversed:

```
- strand:  3' <----Gene3-------------Peak-------------Gene4----| 5'

                     ---Upstream--->    <---Downstream---
```

i.e. the peak is upstream of `Gene3` and downstream of `Gene4`.

For gene-centric analyses, the direction is from the point of view of the gene i.e. for the + strand:

```
                    ---Downstream-->    <---Upstream---

+ strand:  5' |----Peak1-------------Gene-------------Peak2----> 3'
```

(Here the gene is downstream of `Peak1` and upstream of `Peak2`).

For the - strand:

```
- strand:  3' <----Peak3-------------Gene-------------Peak4----| 5'

                     ---Upstream--->    <---Downstream---
```

(The gene is upstream of `Peak3` and downstream of `Peak 4`).

CHAPTER 6

Known problems

6.1 Command installs as 'rnachipintegrator' not 'RnaChipIntegrator'

When installed, the package should provide a command called `RnaChipIntegrator`; however in some circumstances it appears that the command name is converted to all lower-case, and is installed as `rnachipintegrator` instead.

It's not clear why this happens but may be related to the version of `pip` that is used to install the software: the behaviour has been observed when using `pip` version 7.1.2 but not with version 9.0.1.

In these cases the workaround is to use `rnachipintegrator` rather `RnaChipIntegrator` when running the program.

(See issue #48)

CHAPTER 7

Citing RnaChipIntegrator

If you use `RnaChipIntegrator` in your published work, please cite the following reference:

- Briggs PJ, Donaldson IJ, Zeef LAH. RnaChipIntegrator (version 2.0.0). Available at: https://github.com/fls-bioinformatics-core/RnaChipIntegrator

The version number of the program can be obtained by executing the command:

```
RnaChipIntegrator --version
```

CHAPTER 8

Additional information

8.1 Technical details

`RnaChipIntegrator` has been tested against the following versions of Python:

- 2.7
- 3.5
- 3.6
- 3.7
- 3.8

It requires the external `xlsxwriter` library in order to generate the `.xlsx` Excel spreadsheets:

- http://xlsxwriter.readthedocs.io/index.html

This library will be installed automatically if using `pip`.

The source code for the program is hosted on GitHub at

- https://github.com/fls-bioinformatics-core/RnaChipIntegrator

The `devel` branch holds the developmental code and can be installed directly from GitHub using `pip`:

```
pip install git+https://github.com/fls-bioinformatics-core/RnaChipIntegrator.git@devel
```

The unit tests for the code can be run using:

```
python setup.py test
```

(Note that this requires the `nose` package.)

Additionally there are a set of integration tests for the utility in the `examples` subdirectory. These can be run by executing the `run_examples.sh` script.

Both the integration and unit tests are also run on the Travis-CI continuous integration server each time a change is made to the code; the test results can be found at https://travis-ci.org/fls-bioinformatics-core/RnaChipIntegrator/

8.2 Documentation

The documentation under the `docs` subdirectory is generated using the `sphinx` package, and can be built by doing either:

```
python setup.py sphinx_build
```

or:

```
cd docs
make html
```

both of which create the documentation in the `docs/_build` subdirectory.

A copy of the documentation is also hosted on ReadTheDocs at http://rnachipintegrator.readthedocs.io/en/latest/

8.3 Credits

`RnaChipIntegrator` was written by Peter Briggs, Ian Donaldson and Leo Zeef in the Bioinformatics Core Facility (BCF) in the Faculty of Life Sciences, University of Manchester, with additional contributions from Casey Bergman.

8.4 Licensing

This software is licensed under the Artistic License 2.0; see the `LICENSE` document.

8.5 Version history and changes

See the CHANGELOG.