Zerone tutorial

Pol Cuscó and Guillaume Filion

October 29, 2015

1 Building instructions

Zerone is available as a Linux command line application and as an R package.

1.1 Downloading

We recommend that you use git to keep Zerone updated. You can clone the repository from Github with the following command on a standard terminal.

```
git clone git@github.com:gui11aume/zerone
```

Note that this requires that you already have a Github account and that the computer you are working on has an SSH key registered on GitHub. If this is not the case, follow the instructions from `https://help.github.com/articles/generating-ssh-keys/`.

Alternatively, if you prefer not to use git, you can download the source code from `https://github.com/gui11aume/zerone/archive/master.zip` with the following commands.

```
wget https://github.com/gui11aume/zerone/archive/master.zip
unzip zerone-master.zip
mv zerone-master zerone
```

This should create a directory named `zerone`.

1.2 Compiling

To build Zerone, execute the following from the `zerone` directory.

```
cd zerone
make
```

This should succeed on most Linux systems because `make` is available by default. If this is not the case, you can obtain it by typing `sudo apt-get install make` on the Ubuntu terminal.

Calling `make` should create an executable called `zerone`.

1.3 Testing

To check that the building was successful, test Zerone with the following commands.

```
cd src/test
make test
cd ../..
```

If it passes the tests without any error message, then everything went fine and you are done with the build. If not, something went wrong. In this case, you can explain how to reproduce the problem on `https://github.com/gui11aume/zerone/issues`.

Installing the Zerone R package

To install the Zerone R package, simply run this command from the `zerone` directory.

```
R CMD INSTALL ZeroneRPackage
```

Note that you need to have R installed on your computer. If this is not the case, run the command `sudo apt-get install r-base` on Ubuntu.

2 Zerone basics

2.1 Running Zerone

To run Zerone, you have to specify the files that contain the mapped reads of the ChIP-seq experiment you want to discretize. These can be in BED, SAM/BAM and GEM (.map) formats. You can include as many negative control files and as many experimental replicates as you need. Just enter controls after the `-0` or `--mock` option, and targets after the `-1` or `--chip` option.

For example, you can type in the following commands from the `zerone` directory.

```
./zerone --mock data/mock.sam --chip data/ctcf1.sam,data/ctcf2.sam
```

Where file1.bam and file2.bam are negative controls done without antibody, and file2.bam and file3.bam are two experimental replicates.

Note that path expansion will not work when using comma separated file names, so if you want to use path names starting with ∼, you can simply specify the mock and chip options as many times as needed, as shown below.

```
./zerone -0 data/mock.sam -1 data/ctcf1.sam -1 data/ctcf2.sam
```

Enter the option `-h` or `--help` for usage instructions and `--version` to print the version number.

2.2 Output

Running Zerone as shown before should produce an output like the following.

```
# QC score: 1.289
# advice: accept discretization.
chr1    1       300     1   0   0   0
chr1    301     600     1   0   0   0
chr1    601     900     1   0   0   0
chr1    901     1200    1   0   0   0
...
chr1    998701  999000  2   1   11  4
chr1    999001  999300  2   2   72  34
chr1    999301  999600  2   2   128 69
chr1    999601  999900  2   0   10  5
```

The first two lines contain the result of the **quality control**. It consists of a quality score followed by an advice to either **accept or reject** the discretization, if the score is positive or negative, respectively. If the score is higher than 1 (or lower than -1), the advice is extremely reliable.

The rest of the lines contain the discretization proper. The first three columns specify the chromosome, start and end positions of each window.

The fourth column represents the **enrichment**. Zerone classifies each window into one of three possible states. States 0 and 1 represent two types of background signal. State **2 represents an enriched window**.

The fifth column contains the read count of all the control profiles summed together, and the rest of the columns contain the read count of each of the target profiles.

2.3 List output

With the `-l` or `--list-output` option, Zerone produces an alternative output in which only enriched windows are shown. Also, all contiguous windows are merged together.

```
# QC score: 1.289
# advice: accept discretization.
chr1    237601  238200
chr1    521401  522000
chr1    567301  567900
chr1    714001  714900
...
chr1    975901  976500
chr1    990001  990600
chr1    994201  995400
chr1    998701  999900
```

The Zerone R package

You can load the package in R with this line.

```
library(zerone)
```

Type `?zerone` from an R session to see the documentation.

3 Troubleshooting

In case Zerone crashes, start by recompiling it in debug mode. To do so, run the following commands from the `zerone` directory.

```
make clean
make debug
```

Then repeat the actions that triggered the crash and contact guillaume.filion@gmail.com attaching the debug information.