Rational Function Complexity Manual

Timothy Daley Andrew Smith

June 18, 2012

Chapter 1

Quick Start

The Rational Function Complexity package is aimed at predicting the yield and number of distinct reads from a genomic library from an initial sequencing experiment. The estimates can then be used to examine the utility of further sequencing, optimize the sequencing depth, or to screen multiple libraries to avoid low complexity samples.

1.1 Installation

1.1.1 Download

Rational Function Complexity is availible at
`http://smithlab.cmb.usc.edu/software/RationalFunctionComplexity/`.

1.1.2 System Requirements

Rational Function Complexity runs on Unix-type system with GNU Scientific Library (GSL), available at `http://www.gnu.org/software/gsl/` and GNU Complilation Collection (GCC) (if you would like to compile it yourself), available at `http://gcc.gnu.org/`. It has been tested on Linux and Mac OS-X.

1.1.3 Installation

Download the source code and decompress it with

```
 $ tar xvfz RationalFunctionComplexity.tar.gz
```

Enter the RationalFunctionComplexity/ directory and run

```
$ make all
```

If compiled successfully, the executable files are available in **RationalFunctionComplexity/**.

1.2 Using Rational Function Complexity

1.2.1 Basic usage:

To generate the complexity plot of a genomic library from a sorted read file in .bed format, use the program *c_curve*. Use -o to specify the output name.

```
$ complexity_plot -o output.txt input.bed
```

To estimate the future yield of a genomic library using an initial experiment in .bed format, use the program *lc_extrap*. Use -o to specify the output of the yield estimates. The options -e and -s set the maximum number of total reads from which yield estimates are desired and the number of reads between estimates, respectively. For confidence intervals of the estimates, -b controls the number of resamples to take and is set to 100 as a default, while -c controls the confidence level and is 95% as default. The last parameter is a .bed file sorted by chromosome, end position, start position, and strand.

```
$ library_complexity -o yield.txt input.txt
```

1.3 File Format

Input files are mapped read files sorted by chromosome, end position, start position, and strand in either .bed format.

Chapter 2

Detailed usage

2.1 c_curve

c_curve is used to compute the expected complexity curve of a mapped read file by subsampling without replacement and counting the distinct reads. Output is a text file with two columns. The first gives the total number of reads and the second the corresponding number of distinct reads.

-o, -output Name of output file, default prints to screen

-v -verbose Prints more information

2.2 lc_extrap

lc_extrap is used to compute the expected yield for theoretical larger experiments and bounds on the number of distinct reads in the library and the associated confidence intervals, computed by bootstrapping. Output is a test file with four columns. The first is the total number of reads, second gives the corresponding average expected number of distinct reads, and the third and fourth give the lower and upper limits of the confidence interval. Specifying verbose will print out the histogram of the input file. To print out the fitted coefficients of the continued fraction approximation, specify verbose and set bootstraps to 0 or 1.

-o, -output Name of yield output file, defaults prints to screen.

-e, -extrapolation_length The maximum number of total reads to compute yield estimates for. The default is 10 billion reads.

-s, -step The step size between yield estimates. Default is 1 million reads.

-b, -bootstraps The number of bootstraps used to compute the confidence intervals, default is 100. To do a single estimate, set to 0 or 1. This will allow for output of estimated coefficients if VERBOSE is also selected.

-c, -c_level Confidence of confidence level. Default is 0.95.

-v, -verbose Print more information.