preseq Manual

Timothy Daley Victoria Helus Andrew Smith

August 5, 2013

Quick Start

The **preseq** package is aimed at predicting the yield 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.

The two main programs are `c_curve` and `lc_extrap`. `c_curve` samples reads without replacement from the given mapped sequenced read file or duplicate count file to estimate the yield of the experiment and the subsampled experiments. These estimates are used construct the complexity curve of the experiment. `lc_extrap` uses rational function approximations of Good & Toulmin's [2] non-parametric empirical Bayes estimator to predict the yield of future experiments, in essence looking into the future for hypothetical experiments. `lc_extrap` is used to predict the yield and then `c_curve` can be used to check the yield from the larger experiment.

1 Installation

Download

preseq is available at `http://smithlab.cmb.usc.edu/software/`.

System Requirements

preseq runs on Unix-type system with GNU Scientific Library (GSL), available at `http://www.gnu.org/software/gsl/`. If the input file is in BAM format, BamTools is required, available at `https://github.com/pezmaster31/bamtools`. If the input is a text file of counts in a single column or is in BED format, BamTools is not required. It has been tested on Linux and Mac OS-X.

Installation

Download the source code and decompress it with

```
$ tar -jxvf preseq.tar.bz2
```

Enter the **preseq/** directory and run

```
$ make all
```

The input file may possibly be in BAM format. If the root directory of BamTools is $bamtools, instead run

```
$ make all BAMTOOLS_ROOT=$bamtools
```

Output after typing this command should include the flag `-DHAVE_BAMTOOLS` if the linking is successful. If compiled successfully, the executable files are available in **preseq/**.
If a BAM file is used as input without first having run `$ make all BAMTOOLS_ROOT=/loc/of/bamtools`, then the following error will occur: `terminate called after throwing an instance of 'std::string'`.

2 Using preseq

Basic usage

To generate the complexity plot of a genomic library from a read file in BED or BAM format or a duplicate count file, use the program `c_curve`. Use `-o` to specify the output name.

```
$ ./c_curve -o complexity_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`. The required options are `-o` to specify the output of the yield estimates and the input file, which is either a BED file sorted by chromosome, end position, start position, and strand or a BAM file sorted with BamTools or SAMTools sort function. Additional options are available and are detailed below.

```
$ ./lc_extrap -o future_yield.txt input.txt
```

3 File Format

Sorted read files in BED or BAM format

Input files are sorted mapped read files in BED or BAM format, or a text file consisting of one column giving the observed read counts. The programs require that BED files are sorted by chromosome, end position, start position, and strand. This can be achieved by using the command line function sort as follows:

```
sort -k 1,1 -k 2,2n -k 3,3n -k 6,6 input.bed > input.sort.bed
```

BAM format read files should be sorted by chromosome and start position. This can be done with either SAMTools (available at `http://samtools.sourceforge.net/`) or BamTools sort functions. If the input is in BAM format, then the flag `-B` must be included.
If the input is paired end, the option `-P` can be set. In this case only concordantly mapped reads are counted. This means that if either end does not map or if the mapping locations of the two ends are not compatible, the read is not counted. If a large number of reads are disconcordant, then the default single end should be used. In this case only the mapping location of the first mate is used as the unique molecular identifier [3].

Text files of observed read counts

For more general applications **preseq** allows the input to be a text file of observed read counts, one count per line. To specify this input, the option `-V` must be set.
Such a text file can typically be constructed by command line arguments. Take for example an unmapped sequencing experiment in FASTQ format. To predict the complexity, the unique molecular identifier needs to use only the observed sequence. For instance, a unique molecular identifier used may be the first 20 bases in the observed sequence. A command line argument to construct the counts would then be

```
awk 'if (NR%4==2) print substr($0,1,20);' input.fastq | sort | uniq -c

| awk 'print $1' > counts.txt
```

More complicated unique molecular identifiers can be used, such as mapping position plus a random barcode, but are too complicated to detail in this manual. For questions with such usage, please contact us at `tdaley@usc.edu`

4 Detailed usage

c_curve

`c_curve` is used to compute the expected complexity curve of a mapped read file by subsampling smaller experiments 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
-s, -step	The step size for samples. Default is 1 million reads
-B, -bam	Input file is in BAM format
-P, -pe	Input is a paired end read file
-H, -hist	Input is a text file of the observed histogram
-V, -vals	Input is a text file of read counts

lc_extrap

`lc_extrap` is used to generate the expected yield for theoretical larger experiments and bounds on the number of distinct reads in the library and the associated confidence intervals, which is computed by bootstrapping the observed duplicate count histogram. Output is a text 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.

-o, -output	Name of output file. Default prints to screen
-e, -extrapolation_length	The maximum number of total reads to compute yield estimates for. The default is 10 billion reads.
-v, -verbose	Prints more information
-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
-c, -cval	Level for confidence intervals. Default is 0.95
-B, -bam	Input file is in BAM format
-P, -pe	Input is a paired end read file
-H, -hist	Input is a text file of the observed histogram
-V, -vals	Input is a text file of read counts

`-o`	Computes the expected yield without confidence intervals and avoids bootstrapping for significantly faster computations

5 lc_extrap Examples

Usage and output of `c_curve` is similar, so the following examples are of `lc_extrap` and its different options.

Using a sorted read file in BED (or BAM with the `-B` flag) format as input

```
$ ./lc_extrap -o future_yield.txt input.bed
```

```
TOTAL_READS    EXPECTED_DISTINCT  LOGNORMAL_LOWER_95%CI  LOGNORMAL_UPPER_95%CI
0                           0                                      0
1000000.0                955978.6                      953946.4                           958015.
2000000.0                1897632.0           1892888.4                   1902387.5
3000000.0                2829410.5           2819146.4                   2839712.0
4000000.0                3751924.0           3732334.5                   3771616.2
....



9999000000.0      185394069.4                    76262245.8             450694319.0
```

This example uses a sorted read file in BED format from an initial experiment generated from single sperm cells. As noted above, the default step size between yield estimates is 1 million, the default confidence interval level is 95%, and the default extrapolation length is 10 billion.

Using a sorted read file in BED format as input, including the verbose option

```
$ ./lc_extrap -o future_yield.txt input.bed -v
```

As `lc_extrap` is running, information will print to screen that gives a read counts histogram of the input file which truncates after the first bin value that has zero observations. Included here is the first 10 lines of what would be observed:

```
TOTAL READS     = 536855
DISTINCT READS  = 516200
DISTINCT COUNTS = 48
MAX COUNT       = 269
COUNTS OF 1     = 511413
MAX TERMS       = 100
OBSERVED COUNTS (270)
1       511413
2       2202
3       597
```

Using a sorted read file in BED format as input, with options

```
$ ./lc_extrap -e 15000000 -s 500000 -b 90 -c .90 -o future_yield.txt input.bed


TOTAL_READS    EXPECTED_DISTINCT  LOGNORMAL_LOWER_90%CI  LOGNORMAL_UPPER_90%CI
0                          0                                          0
500000.0             481098.5                                  480329.1
1000000.0      956070.6                                   954493.7
1500000.0      1428183.4                                  1425461.7
2000000.0      1897886.0                                  1892501.7
....



14500000.0     12932529.0                                 12056525.8                                 1
```

Notice the slight changes, with the step sizes of the extrapolation now at 500,000 as specified, and the maximum extrapolations ending at 15,000,000. The confidence intervals are now at a level of 90%.

Using a histogram or read counts as input

`lc_extrap` allows the input file to be an observed histogram. An example of the format of this histogram is as followed:

```
1       1.68166e+07
2       4.55019e+06
3       1.93787e+06
4       1.07257e+06
5       708034
6       513134
7       384077
8       282560
9       206108
10      146334
```

The following command will give output of the same format as the above examples.

```
$./lc_extrap -o future_yield.txt -H histogram.txt
```

Similarly, both `lc_extrap` and `c_curve` allow the option to input read counts, which is just the histogram but with the bin values excluded (text file should contain ONLY the observed counts in a single column). Command should be run with the `-V` flag:

```
$./lc_extrap -o future_yield.txt -V counts.txt
```

6 preseq application examples

Screening multiple libraries

This section provides a more detailed example using data from different experiments to illustrate how preseq might be applied. Because it is important to avoid spending time on low complexity samples, it is important to decide after observing an initial experiment whether or not it is beneficial to continue with sequencing. The data in this example comes from a study (accession number SRA061610) using single cell sperm cells amplified by Multiple Annealing and Looping Based Amplification Cycles (MALBAC) [4] and focuses on three libraries coming from different experiments from the study (SRX205369, SRX205370, SRX205372).
These libraries help show what would be considered a relatively poor library and a relatively good library, as well as compare the complexity curves obtained from running `c_curve` and `lc_extrap`, to show how `lc_extrap` would help in the decision to sequence further. The black diagonal line represents an ideal library, in which every read is a distinct read (though this cannot be achieved in reality). The full experiments were down sampled at 5% to obtain a mock initial experiment of the libraries, as shown here, where we have the complexity curves of the initial experiments generated by `c_curve`:

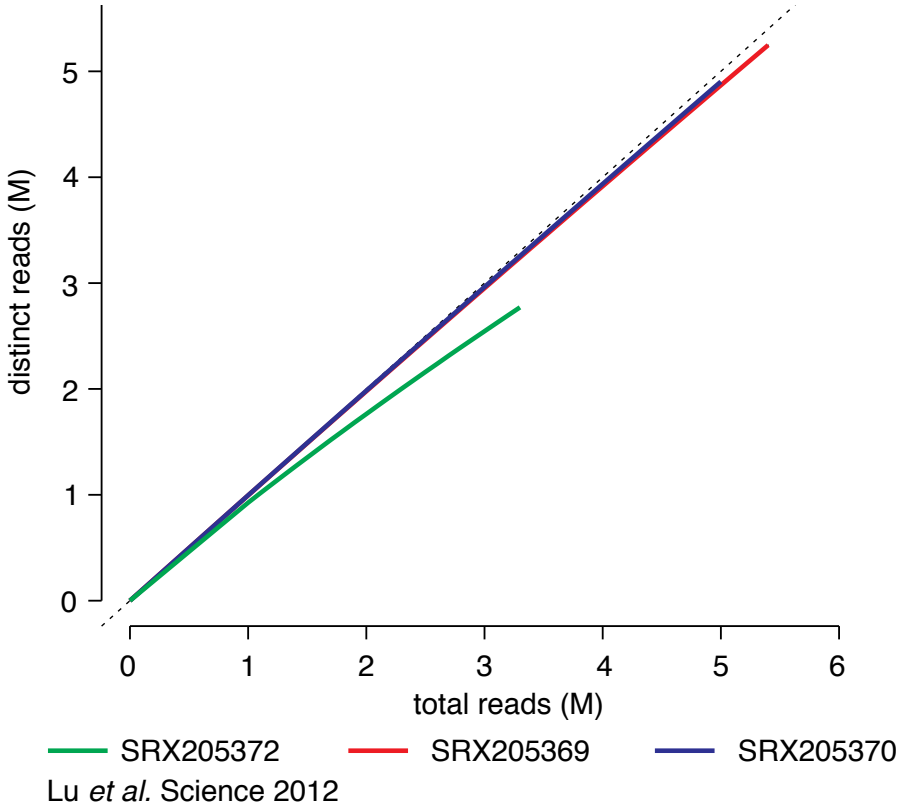

Figure 1: Initial observed complexities

With such a relatively small amount of reads sequenced, it is hard in the first stages of a study to guess at whether it is not worth sequencing a library further, as all three libraries seem to be relatively good.

This is a comparison of the full experiment complexity curves and the extrapolated complexity curves created using information from the initial experiments above as input. The dashed lines indicate the complexity curves predicted by `lc_extrap`, and the solid lines are the expected complexity curves of the full experiments, obtained using `c_curve`. Note that the dashed curves follow the solid curves very closely, only differing slightly towards the end, meaning `lc_extrap` gives a good predicted yield curve. Using this, it is clear that if the initial experiments were the only available data and `lc_extrap` was run, SRX205372 would likely be discarded, as it is a poor library, and SRX205369 and SRX205370 would probably be used for further sequencing, as their complexity curves indicate that sequencing more would yield enough information to justify the costs. If the researcher were to only want to sequence one library deep, then SRX205370 would be an obvious choice.

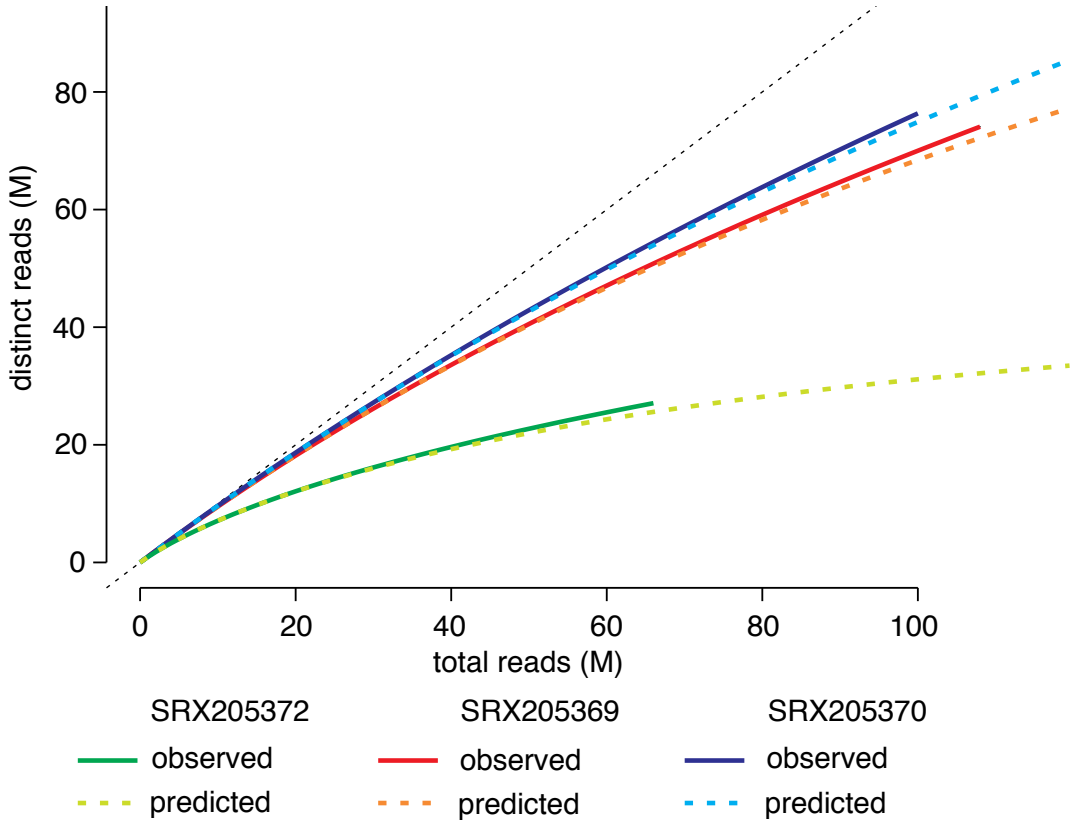

Figure 2: Estimated versus observed library complexities.

Saturation of reads and junctions for RNA sequencing experiments

A recent paper from the Rinn lab [5] developed a targeted capture RNA sequencing protocol to deeply investigate chosen portions of the transcriptome. A comparison of the results from a standard RNA sequencing experiment (RNA-seq; SRA accession SRX061769) and a targeted capture RNA sequencing experiment (Capture-seq; SRA accession SRX061768) reveals a startling amount of transcriptional complexity missed by standard RNA sequencing in the targeted regions. A large number of rare transcriptional events such as alternative splices, alternative isoforms, and long non-coding RNAs were newly identified with the targeted sequencing.

A current vigorous debate exists on whether these rare events are truly transcriptional events or are merely due to sequencing or transcriptional noise (see [6] and [1]). We do not seek to address these issues, but merely to comment on the true complexity of rare transcriptional events in sequencing libraries identified by current protocols.

We took the two Illumina sequencing libraries from [5] and mapped them according to the protocol given. We downsampled 10% of the library and compared the estimated library complexities (single end) with the observed library complexity for both libraries. We also took the junction information contained in the file junctions.bed in the Tophat output folder to estimate the junction complexity. To obtain a value text file suitable for preseq, we simply cut out the 5th column of the junctions.bed file.

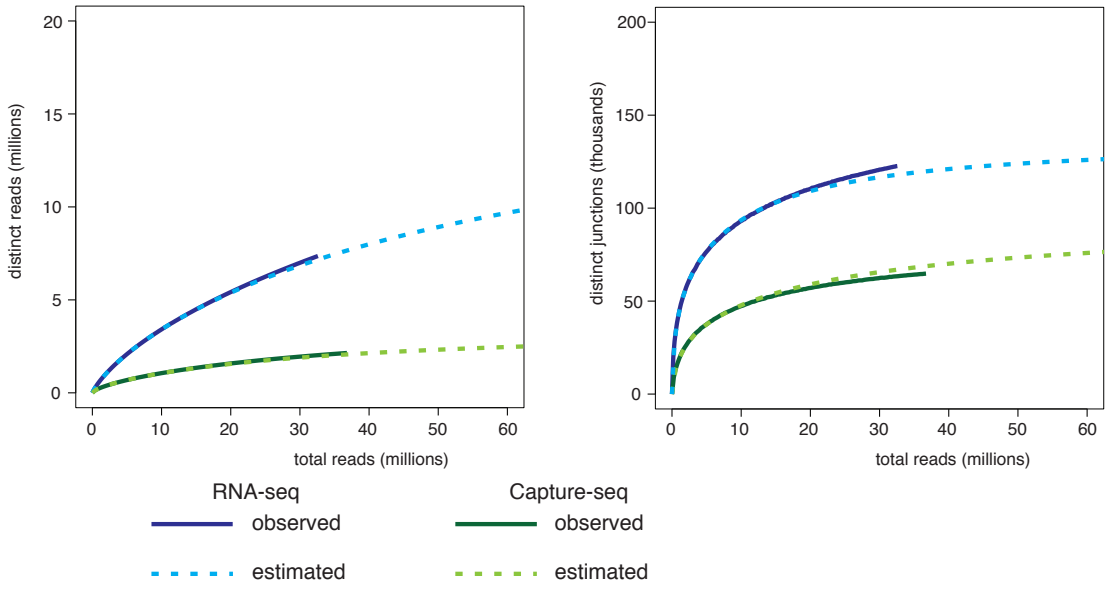

Figure 3: A comparison of complexities of standard RNA-seq and targeted capture RNA-seq.

We see from the estimated library that the RNA-seq library is far from saturated, while it appears that the Capture-seq library may be close. On the other hand, the junction complexity of both libraries indicates that the full scope of juctions identified by Tophat is far from saturated in both libraries. This indicates that large number of rare junctions still remain to be identified in the libraries.

References

[1] Michael B Clark, Paulo P Amaral, Felix J Schlesinger, Marcel E Dinger, Ryan J Taft, John L Rinn, Chris P Ponting, Peter F Stadler, Kevin V Morris, Antonin Morillon, et al. The reality of pervasive transcription. *PLoS biology*, 9(7):e1000625, 2011.

[2] I. J. Good and G. H. Toulmin. The number of new species, and the increase in population coverage, when a sample is increased. *Biometrika*, 43:45–63, 1956.

[3] T. Kivioja, A. Vähärautio, K. Karlsson, M. Bonke, M. Enge, S. Linnarsson, and J. Taipale. Counting absolute numbers of molecules using unique molecular identifiers. *Nature Methods*, 9:72–74, 2012.

[4] Sijia Lu, Chenghang Zong, Wei Fan, Mingyu Yang, Jinsen Li, Alec R Chapman, Ping Zhu, Xuesong Hu, Liya Xu, Liying Yan, et al. Probing meiotic recombination and aneuploidy of single sperm cells by whole-genome sequencing. *Science*, 338(6114):1627–1630, 2012.

[5] T.R. Mercer, D.J. Gerhardt, M.E. Dinger, J. Crawford, C. Trapnell, J.A. Jeddeloh, J.S. Mattick, and J.L. Rinn. Targeted RNA sequencing reveals the deep complexity of the human transcriptome. *Nature Biotechnology*, 30(1):99–104, 2011.

[6] Harm van Bakel, Corey Nislow, Benjamin J Blencowe, and Timothy R Hughes. Most dark matter transcripts are associated with known genes. *PLoS biology*, 8(5):e1000371, 2010.