Complexity analysis of HR5109

Timothy Daley

June 8, 2015

To obtain feature counts I first split multiple features with an awk script to obtain one line per feature (reads can appear multiple times.

```
awk '{where=match($4,"serghei");
if(where)
{
  split($4, a, "serghei");
  for(stuff in a) print $1"\t"$2"\t"$3"\t"a[stuff];
}
else
{
  print $0;
}
}' mapped_HR5109_features_unsep.txt > mapped_HR5109_features_sep.txt
```

This gives 9 different features with the following counts:

feature	counts
CDS	3194397
UTR	7609716
acrossB	195801
exon	10220484
intron	5784570
intron_retention	1913020
junction	32199494
mateAcrossB	1414948
multiMapped	2544843

Of these features, the ones with location information that we can use to identify duplicate events are exons, junctions, CDSs, and UTRs. To obtain the counts we use a simple bash script. For example the junction counts can be obtained by the following script.

```
awk '{if (match($4, "junction")) print $4}' mapped_HR5109_features_sep.txt | sort | uniq -c | awk '{pri
```

The feature counts can be fed into the preseq program with the -V option.

```
for feature in exon junction CDS UTR; do echo $feature; ~/panfs/programs/preseq lc_extrap -V -v -s 1000
```

We plot the library complexity as a function of total fragments sequenced, calculated to be x by samtools.

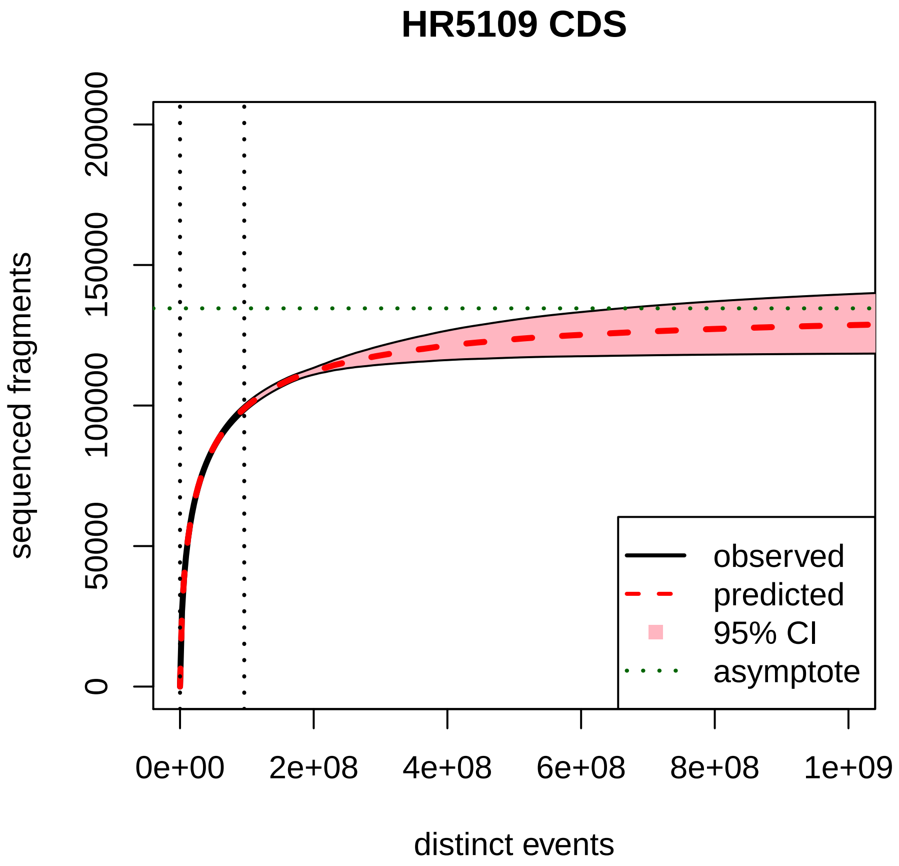
```
# CDS
mapped_HR5109_features_CDS_counts_lc_extrap = read.table(file="mapped_HR5109_features_CDS_counts_lc_ext
tail(mapped_HR5109_features_CDS_counts_lc_extrap)
```

```
##        TOTAL_READS EXPECTED_DISTINCT LOWER_0.95CI UPPER_0.95CI
## 99995   9999400000          134572.5     118520.6     152798.4
## 99996   9999500000          134572.5     118520.6     152798.4
## 99997   9999600000          134572.5     118520.6     152798.4
## 99998   9999700000          134572.5     118520.6     152798.4
## 99999   9999800000          134572.5     118520.6     152798.4
## 100000  9999900000          134572.5     118520.6     152798.4
```

```
mapped_HR5109_features_CDS_counts_c_curve = read.table(file="mapped_HR5109_features_CDS_counts_c_curve.t
tail(mapped_HR5109_features_CDS_counts_c_curve)
```

```
##    total_reads distinct_reads
## 27     2600000          94540
## 28     2700000          95330
## 29     2800000          96086
## 30     2900000          96813
## 31     3000000          97511
## 32     3100000          98181
```

```
scaling_val = 96070080/3194397
plot(scaling_val*mapped_HR5109_features_CDS_counts_c_curve$total_reads, mapped_HR5109_features_CDS_coun
polygon(c(scaling_val*mapped_HR5109_features_CDS_counts_lc_extrap$TOTAL_READS, rev(scaling_val*mapped_HI
lines(scaling_val*mapped_HR5109_features_CDS_counts_c_curve$total_reads, mapped_HR5109_features_CDS_cou
lines(scaling_val*mapped_HR5109_features_CDS_counts_lc_extrap$TOTAL_READS, mapped_HR5109_features_CDS_c
abline(v = 0, lty=3, lwd=2)
abline(v = 96070080, lty=3, lwd=2)
abline(h = tail(mapped_HR5109_features_CDS_counts_lc_extrap$EXPECTED_DISTINCT, 1), lty=3, lwd=2, col="da
legend("bottomright", legend=c("observed", "predicted", "95% CI", "asymptote"), lty=c(1, 2, NA, 3), pch=
```



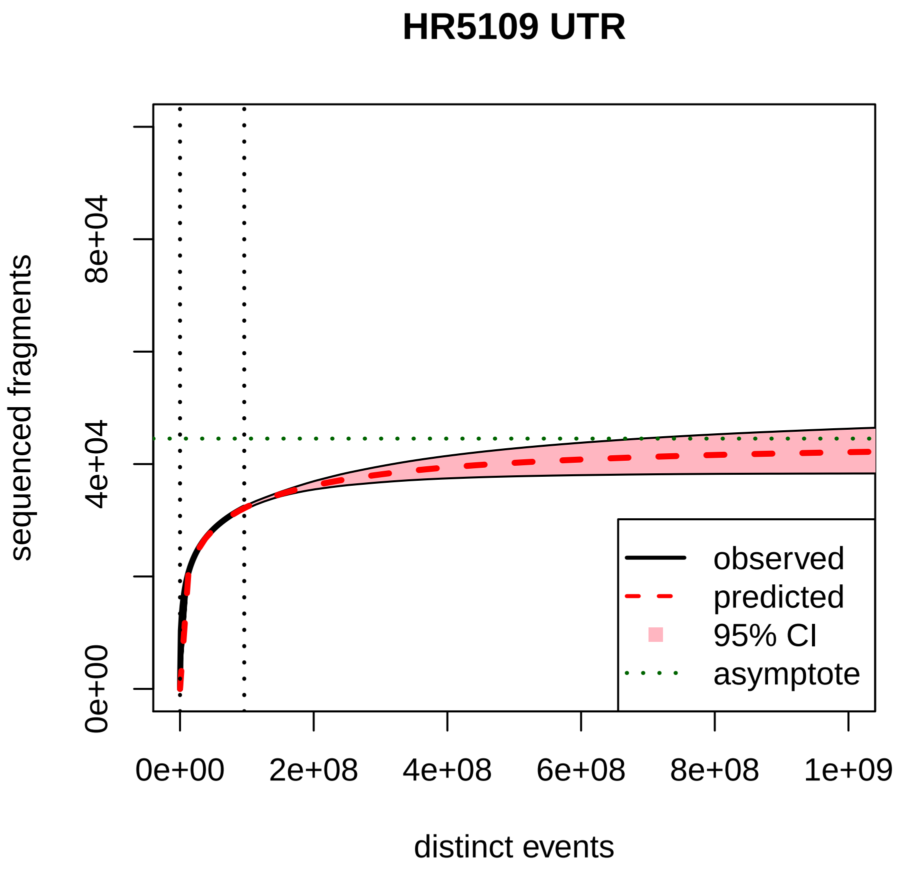
```
# UTR
mapped_HR5109_features_UTR_counts_lc_extrap = read.table(file="mapped_HR5109_features_UTR_counts_lc_ext
tail(mapped_HR5109_features_UTR_counts_lc_extrap)
```

```
##       TOTAL_READS EXPECTED_DISTINCT LOWER_0.95CI UPPER_0.95CI
## 9995    9.994e+09           44530.9      37451.8      52948.1
## 9996    9.995e+09           44530.9      37451.8      52948.1
## 9997    9.996e+09           44530.9      37451.7      52948.1
## 9998    9.997e+09           44530.9      37451.7      52948.1
## 9999    9.998e+09           44530.9      37451.7      52948.1
## 10000   9.999e+09           44530.9      37451.7      52948.1
```

```
mapped_HR5109_features_UTR_counts_c_curve = read.table(file="mapped_HR5109_features_UTR_counts_c_curve.
tail(mapped_HR5109_features_UTR_counts_c_curve)
```

```
##    total_reads distinct_reads
## 72     7100000          31844
## 73     7200000          31927
## 74     7300000          32009
## 75     7400000          32088
## 76     7500000          32168
## 77     7600000          32246
```

```
scaling_val = 96070080/7609716
plot(scaling_val*mapped_HR5109_features_UTR_counts_c_curve$total_reads, mapped_HR5109_features_UTR_coun
polygon(c(scaling_val*mapped_HR5109_features_UTR_counts_lc_extrap$TOTAL_READS, rev(scaling_val*mapped_HR
lines(scaling_val*mapped_HR5109_features_UTR_counts_c_curve$total_reads, mapped_HR5109_features_UTR_coun
lines(scaling_val*mapped_HR5109_features_UTR_counts_lc_extrap$TOTAL_READS, mapped_HR5109_features_UTR_co
abline(v = 0, lty=3, lwd=2)
abline(v = 96070080, lty=3, lwd=2)
abline(h = tail(mapped_HR5109_features_UTR_counts_lc_extrap$EXPECTED_DISTINCT, 1), lty=3, lwd=2, col="da
legend("bottomright", legend=c("observed", "predicted", "95% CI", "asymptote"), lty=c(1, 2, NA, 3), pch=
```



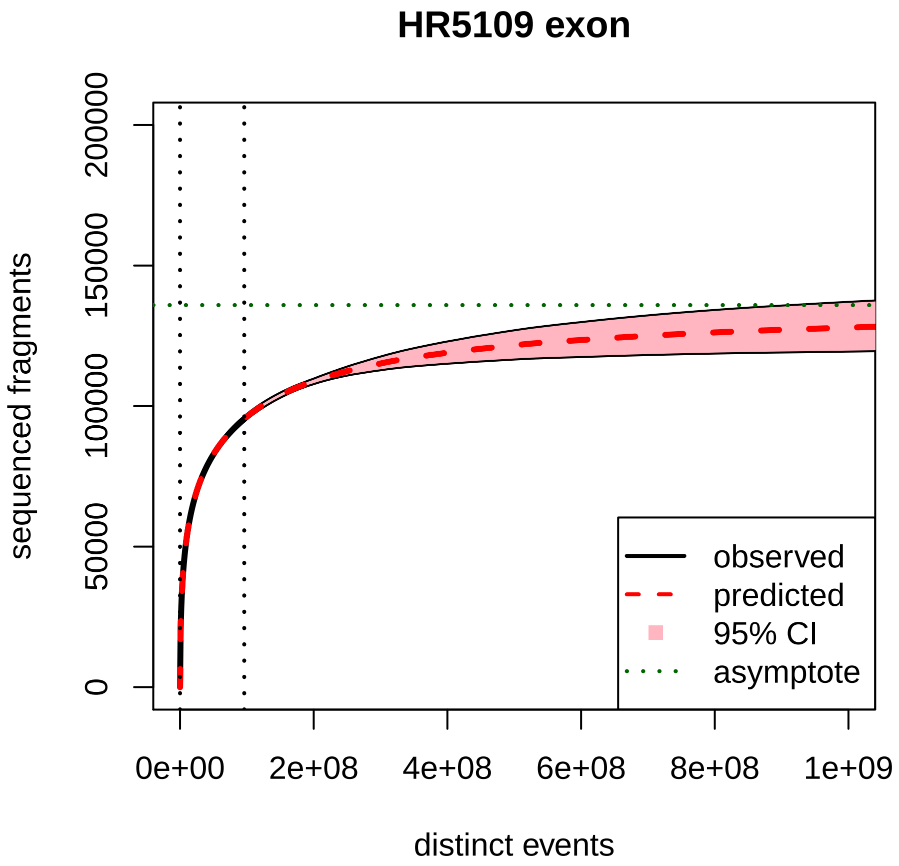
```
# exons
mapped_HR5109_features_exon_counts_lc_extrap = read.table(file="mapped_HR5109_features_exon_counts_lc_ex
tail(mapped_HR5109_features_exon_counts_lc_extrap)
```

```
##        TOTAL_READS EXPECTED_DISTINCT LOWER_0.95CI UPPER_0.95CI
## 99995   9999400000          135924.1     121681.9     151833.2
## 99996   9999500000          135924.1     121681.9     151833.2
## 99997   9999600000          135924.1     121681.9     151833.2
## 99998   9999700000          135924.1     121681.9     151833.2
## 99999   9999800000          135924.1     121681.9     151833.2
## 100000  9999900000          135924.1     121681.9     151833.2
```

```
mapped_HR5109_features_exon_counts_c_curve = read.table(file="mapped_HR5109_features_exon_counts_c_curve
tail(mapped_HR5109_features_exon_counts_c_curve)
```

```
##     total_reads distinct_reads
## 98      9700000          94359
## 99      9800000          94555
## 100     9900000          94750
## 101    10000000          94943
## 102    10100000          95134
## 103    10200000          95323
```

```
scaling_val = 96070080/10220484
plot(scaling_val*mapped_HR5109_features_exon_counts_c_curve$total_reads, mapped_HR5109_features_exon_cou
polygon(c(scaling_val*mapped_HR5109_features_exon_counts_lc_extrap$TOTAL_READS, rev(scaling_val*mapped_H
lines(scaling_val*mapped_HR5109_features_exon_counts_c_curve$total_reads, mapped_HR5109_features_exon_c
lines(scaling_val*mapped_HR5109_features_exon_counts_lc_extrap$TOTAL_READS, mapped_HR5109_features_exon_
abline(v = 0, lty=3, lwd=2)
abline(v = 96070080, lty=3, lwd=2)
abline(h = tail(mapped_HR5109_features_exon_counts_lc_extrap$EXPECTED_DISTINCT, 1), lty=3, lwd=2, col="
legend("bottomright", legend=c("observed", "predicted", "95% CI", "asymptote"), lty=c(1, 2, NA, 3), pch
```



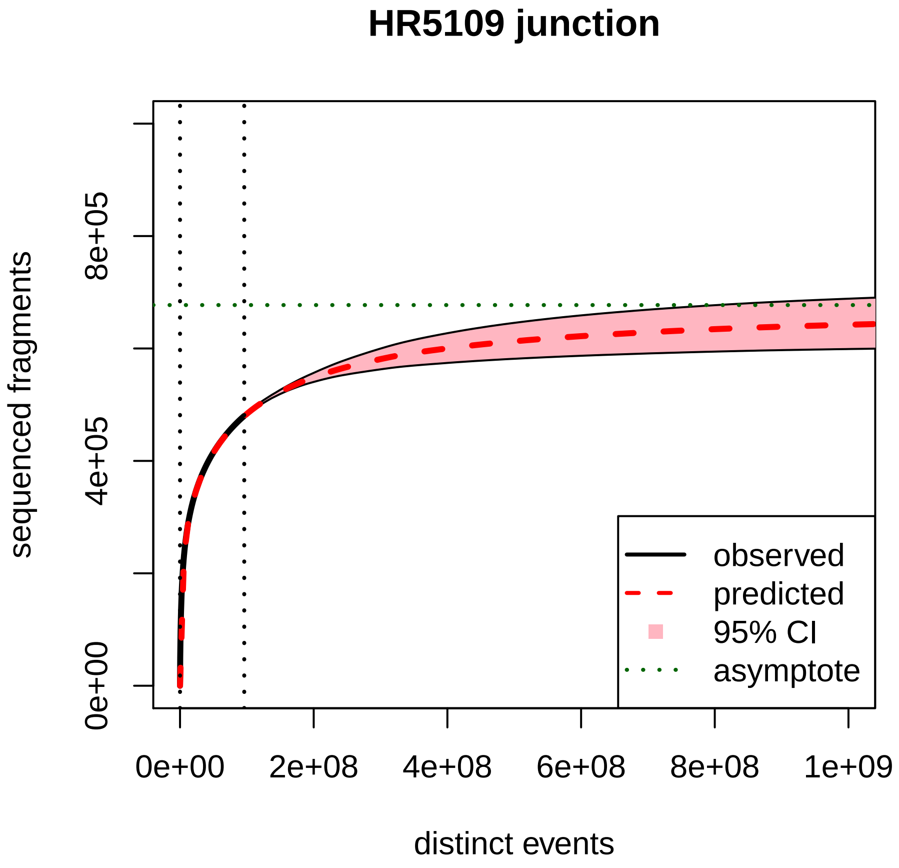
```
# junctions
mapped_HR5109_features_junction_counts_lc_extrap = read.table(file="mapped_HR5109_features_junction_cou
tail(mapped_HR5109_features_junction_counts_lc_extrap)
```

```
##      TOTAL_READS EXPECTED_DISTINCT LOWER_0.95CI UPPER_0.95CI
## 4995   9.988e+09          677253.6     618547.8     741531.2
## 4996   9.990e+09          677253.9     618547.9     741531.6
## 4997   9.992e+09          677254.2     618548.0     741532.0
## 4998   9.994e+09          677254.4     618548.2     741532.5
## 4999   9.996e+09          677254.7     618548.3     741532.9
## 5000   9.998e+09          677255.0     618548.5     741533.3
```

```
mapped_HR5109_features_junction_counts_c_curve = read.table(file="mapped_HR5109_features_junction_count
tail(mapped_HR5109_features_junction_counts_c_curve)
```

```
##     total_reads distinct_reads
## 317    31600000         478229
## 318    31700000         478544
## 319    31800000         478858
## 320    31900000         479171
## 321    32000000         479482
## 322    32100000         479794
```

```
scaling_val = 96070080/32199494
plot(scaling_val*mapped_HR5109_features_junction_counts_c_curve$total_reads, mapped_HR5109_features_jun
polygon(c(scaling_val*mapped_HR5109_features_junction_counts_lc_extrap$TOTAL_READS, rev(scaling_val*mapp
lines(scaling_val*mapped_HR5109_features_junction_counts_c_curve$total_reads, mapped_HR5109_features_ju
lines(scaling_val*mapped_HR5109_features_junction_counts_lc_extrap$TOTAL_READS, mapped_HR5109_features_
abline(v = 0, lty=3, lwd=2)
abline(v = 96070080, lty=3, lwd=2)
abline(h = tail(mapped_HR5109_features_junction_counts_lc_extrap$EXPECTED_DISTINCT, 1), lty=3, lwd=2, c
legend("bottomright", legend=c("observed", "predicted", "95% CI", "asymptote"), lty=c(1, 2, NA, 3), pch
```



We compare this to the read complexity.

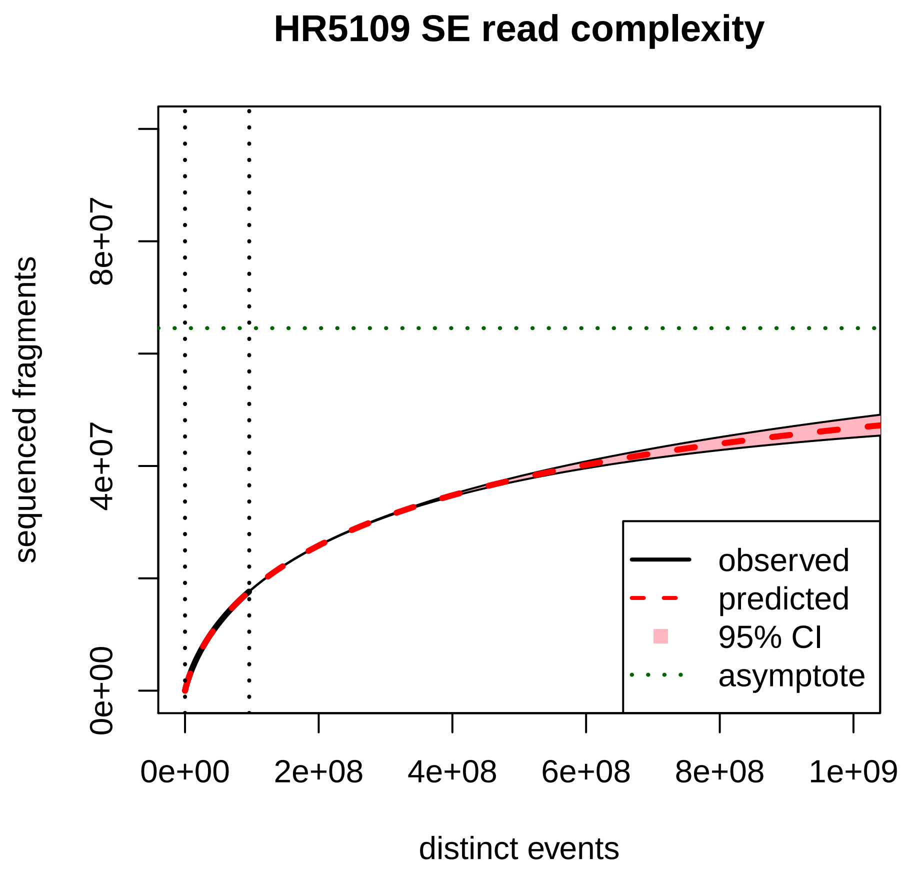
```
# SE read complexity
mapped_HR5109_se_read_dup_lc_extrap = read.table(file="mapped_HR5109_se_read_dup_lc_extrap.txt", header
tail(mapped_HR5109_se_read_dup_lc_extrap)
```

```
##       TOTAL_READS EXPECTED_DISTINCT LOWER_0.95CI UPPER_0.95CI
## 9995    9.994e+09          64533552     46869186     88855381
## 9996    9.995e+09          64533734     46868796     88856621
## 9997    9.996e+09          64533917     46868407     88857861
## 9998    9.997e+09          64534099     46868018     88859101
## 9999    9.998e+09          64534281     46867628     88860341
## 10000   9.999e+09          64534463     46867239     88861581
```

```
mapped_HR5109_se_read_dup_c_curve = read.table(file="mapped_HR5109_se_read_dup_c_curve.txt", header=T)
tail(mapped_HR5109_se_read_dup_c_curve)
```

```
##    total_reads distinct_reads
## 54    53000000       16758100
## 55    54000000       16938600
## 56    55000000       17117000
## 57    56000000       17293300
## 58    57000000       17467600
## 59    58000000       17640000
```

```
scaling_val = 96070080/58196054
plot(scaling_val*mapped_HR5109_se_read_dup_c_curve$total_reads, mapped_HR5109_se_read_dup_c_curve$disti
polygon(c(scaling_val*mapped_HR5109_se_read_dup_lc_extrap$TOTAL_READS, rev(scaling_val*mapped_HR5109_se
lines(scaling_val*mapped_HR5109_se_read_dup_c_curve$total_reads, mapped_HR5109_se_read_dup_c_curve$dist
lines(scaling_val*mapped_HR5109_se_read_dup_lc_extrap$TOTAL_READS, mapped_HR5109_se_read_dup_lc_extrap$
abline(v = 0, lty=3, lwd=2)
abline(v = 96070080, lty=3, lwd=2)
abline(h = tail(mapped_HR5109_se_read_dup_lc_extrap$EXPECTED_DISTINCT, 1), lty=3, lwd=2, col="darkgreen
legend("bottomright", legend=c("observed", "predicted", "95% CI", "asymptote"), lty=c(1, 2, NA, 3), pch
```



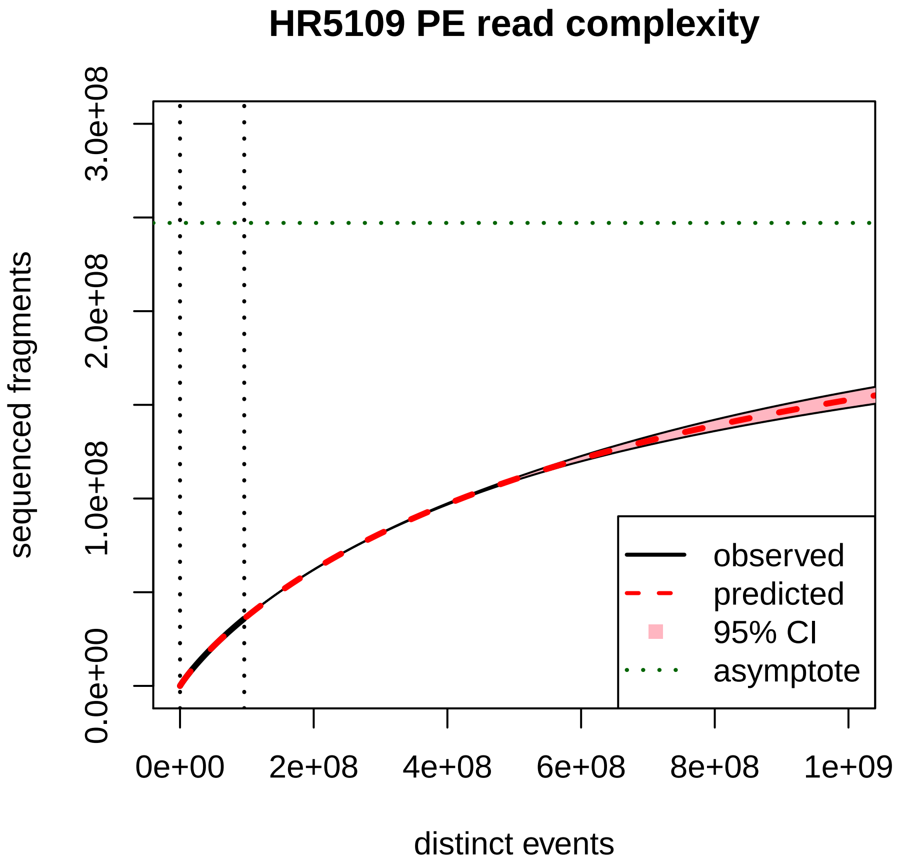
```
# PE read complexity
mapped_HR5109_pe_read_dup_lc_extrap = read.table(file="mapped_HR5109_pe_read_dup_lc_extrap.txt", header=
tail(mapped_HR5109_pe_read_dup_lc_extrap)
```

```
##       TOTAL_READS EXPECTED_DISTINCT LOWER_0.95CI UPPER_0.95CI
## 9995    9.994e+09         247093250    231959457    263214421
## 9996    9.995e+09         247094269    231960302    263215632
## 9997    9.996e+09         247095288    231961148    263216844
## 9998    9.997e+09         247096306    231961993    263218055
## 9999    9.998e+09         247097324    231962837    263219266
## 10000   9.999e+09         247098343    231963682    263220476
```

```
mapped_HR5109_pe_read_dup_c_curve = read.table(file="mapped_HR5109_pe_read_dup_c_curve.txt", header=T)
tail(mapped_HR5109_pe_read_dup_c_curve)
```

```
##    total_reads distinct_reads
## 54    53000000       33357000
## 55    54000000       33854300
## 56    55000000       34348600
## 57    56000000       34840000
## 58    57000000       35328600
## 59    58000000       35814300
```

```
scaling_val = 96070080/58196053
plot(scaling_val*mapped_HR5109_pe_read_dup_c_curve$total_reads, mapped_HR5109_pe_read_dup_c_curve$distin
polygon(c(scaling_val*mapped_HR5109_pe_read_dup_lc_extrap$TOTAL_READS, rev(scaling_val*mapped_HR5109_pe_
lines(scaling_val*mapped_HR5109_pe_read_dup_c_curve$total_reads, mapped_HR5109_pe_read_dup_c_curve$disti
lines(scaling_val*mapped_HR5109_pe_read_dup_lc_extrap$TOTAL_READS, mapped_HR5109_pe_read_dup_lc_extrap$
abline(v = 0, lty=3, lwd=2)
abline(v = 96070080, lty=3, lwd=2)
abline(h = tail(mapped_HR5109_pe_read_dup_lc_extrap$EXPECTED_DISTINCT, 1), lty=3, lwd=2, col="darkgreen
legend("bottomright", legend=c("observed", "predicted", "95% CI", "asymptote"), lty=c(1, 2, NA, 3), pch=
```



Additionally we can estimate lower bounds using a method still in development (see https://www.dropbox.com/s/95lievz7n744851/better_lower_bounds.pdf?dl=0 for a draft of the paper or https://github.com/timydaley/preseq_dev/blob/master/test_quadrature.cpp for the code).

```
for feature in exon junction CDS UTR; do echo $feature; ~/panfs/programs/test_quadrature -p 10 -o mappe
```

This gives the following lower bounds on the total number of features in the library HR5109 along with the number of unobserved and the calculated asymptotes (that can serve as an estimate for the total number of events in the library):

feature	observed	asymptote	lower bound
CDS	98791	134572.5	122514.1
UTR	32253	44530.9	41299.1
exon	95361	135924.1	121753.7
junction	480101	677255.0	576923.0

feature	observed	asymptote	lower bound
SE reads	17673548	64534463	47314599.1
PE reads	35909163	247098343	176951859.5

The fact that most of the features have already been observed indicates that the library is nearly saturated at the current sequencing depth. To investigate the current saturation we use the Good-Turing estimate for the mathematical coverage, aka sample coverage or one minus the discovery probability Good, Biometrika, 1953. The mathematical coverage of a sample from a population is defined as the sum of the probabilities of the observed class, i.e. if x_i is the number of observed individuals from class i then $C = \sum_{i=1}^{\infty} p_i 1\big(x_i > 0\big)$. This represents the relative proportion of the events that have been observed. The Good-Turing estimate for the mathematical coverage is one minus the number of singletons divided by the number of samples.

feature	total	singletons	C
CDS	3194397	20307	0.9936429
UTR	7609716	5920	0.999222
exon	10220484	19204	0.998121
junction	32199494	99616	0.9969063
SE reads	58196054	9951442	0.8290014
PE reads	58196053	28151457	0.5162652

This indicates that though a large number of the molecules in the library have not be sequenced, a large proportion of the events (CDS, UTR, exon, and junction) have been observed. The ones that remain are very low probability events and will take significant sequencing resource to observe and even more to quantify.