In this note we try to figure out why the BIC estimate of the model evidence looks different for simulated data than for real data.

Our real data is from `/tier2/deweylab/nathanae/rnaseq-eval/data-sep6.single-0.001`. Our simulated data is from `/tier2/deweylab/nathanae/rnaseq-eval/data-sep7.single-0.001-sim`.

```
> require("plyr")
> require("ggplot2")
> sim <- read.csv("summ-sim.csv")[1:76, ]
> sample <- read.csv("summ-sample.csv")[1:76, ]
> summ <- rbind(data.frame(sim, dataset = "sim"), data.frame(sample,
+     dataset = "sample"))
```

By looking at the EM.cpp, we see that

1. `rsem.approx.loglikelihood` is the unpenalized log likelihood plus `(N0 > 0 ? N0 * log(theta[0]) : 0.0)`, where `ec[0] = N0`, i.e., $N0$ is the expected number of noise reads.
2. `rsem.eval.loglikelihood` doesn't appear to be the unpenalized log likelihood (at least as calculated by `model.getLogP()`), either.
3. However, the unpenalized log likelihood is available from `expression.temp/expression.ns`.
4. It might be interesting to know what $N0$ above is, also. We can get this from `expression.temp/expression.ofg`.
5. `rsem.approx.bic` is `rsem.approx.loglikelihood` plus the BIC penalty.

```
> if (FALSE) {
+     summ <- adply(summ, 1, function(r) {
+         base <- sprintf("%s.rsem-eval-d/expression.temp", r$summary)
+         fp <- file(sprintf("%s/expression.ofg", base))
+         lines <- readLines(fp)
+         close(fp)
+         N0 <- as.double(strsplit(lines[1], split = " ")[[1]][2])
+         fp <- file(sprintf("%s/expression.ns", base))
+         lines <- readLines(fp)
+         close(fp)
+         loglik <- as.double(lines[1])
+         fp <- file(sprintf("%s/expression.pme_theta", base))
+         lines <- readLines(fp)
+         close(fp)
+         th0 <- as.double(strsplit(lines[1], split = " ")[[1]][1])
+         c(rsem.ofg.N0 = N0, rsem.ns.loglik = loglik, rsem.pme_theta.0 = th0)
+     })
+     write.csv(summ, file = "summ.updated.csv")
+ } else {
+     summ <- read.csv("summ.updated.csv")
+ }
```

We add some convenience columns to our data frame:

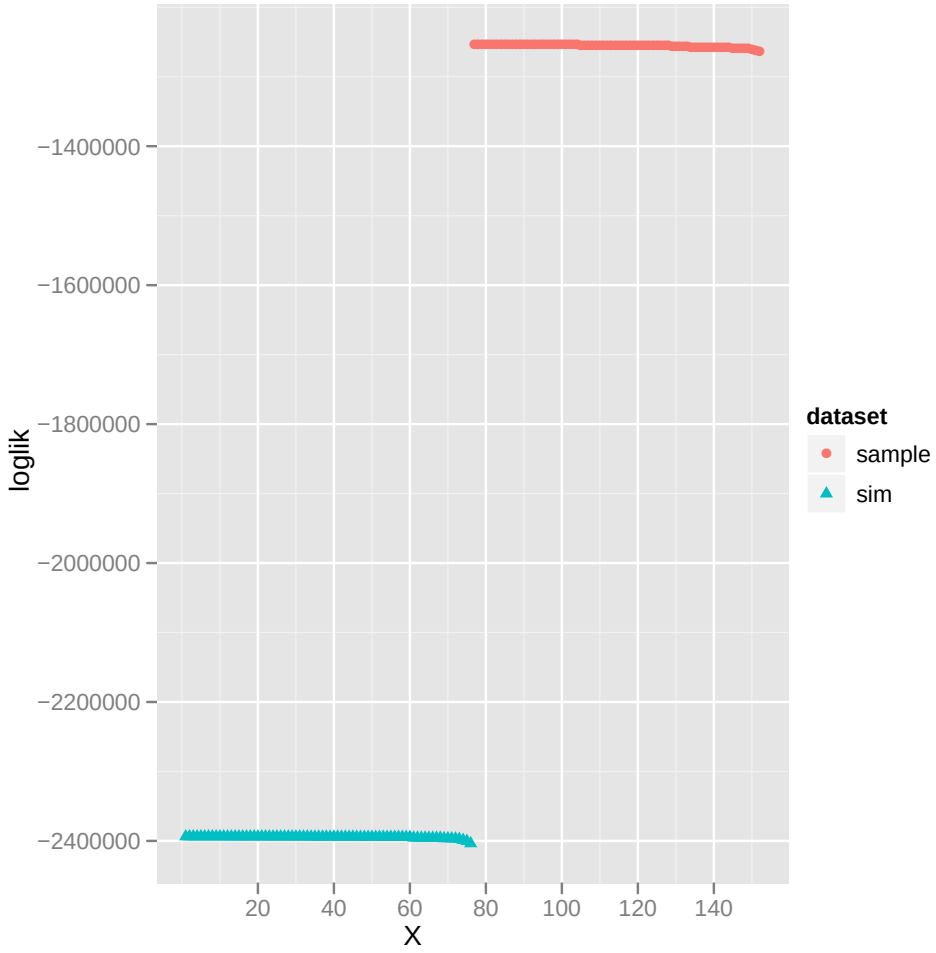
1. `loglik` is the log likelihood, unpenalized
2. `bic` is the BIC penalty

3. `loglik.minus.bic` is the log likelihood minus the BIC penalty

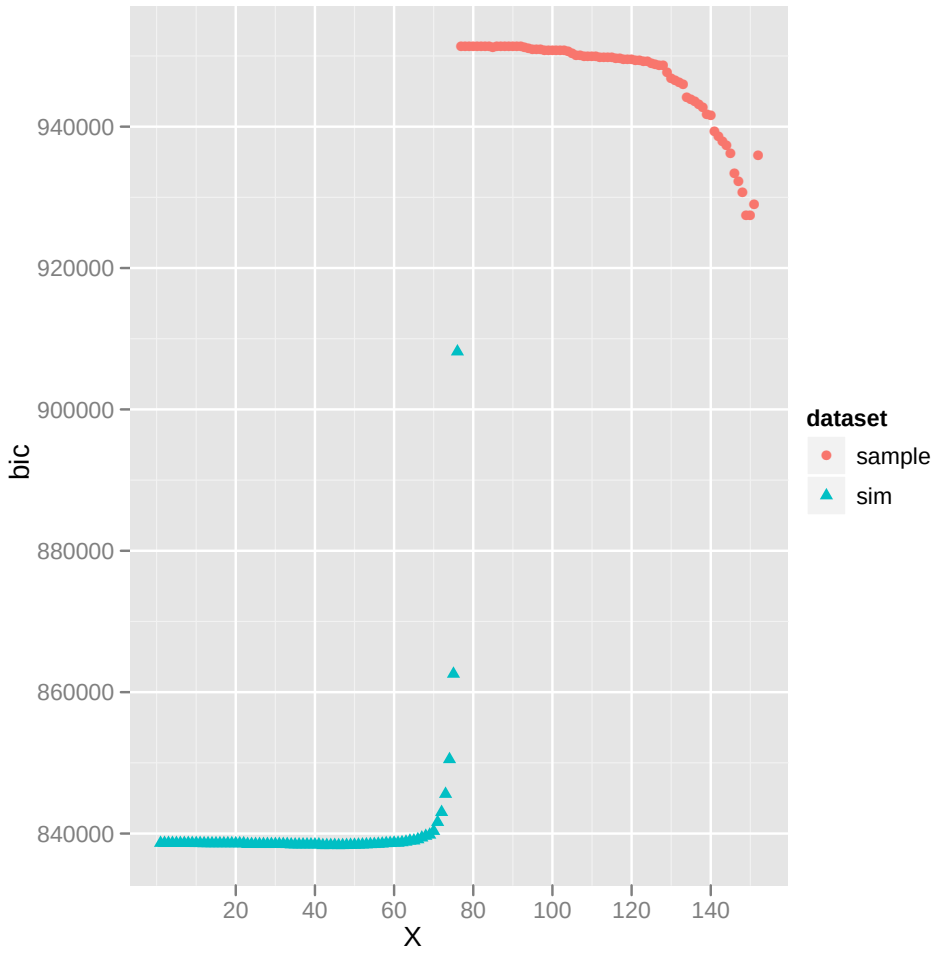
```
> summ <- data.frame(summ, loglik = summ$rsem.ns.loglik, bic = -(summ$rsem.approx.bic -
+     summ$rsem.ns.loglik), loglik.minus.bic = summ$rsem.approx.bic)
> stopifnot(sum(abs(summ$loglik.minus.bic - (summ$loglik - summ$bic))) <
+     1e-05)
```

Now let's plot each of these:

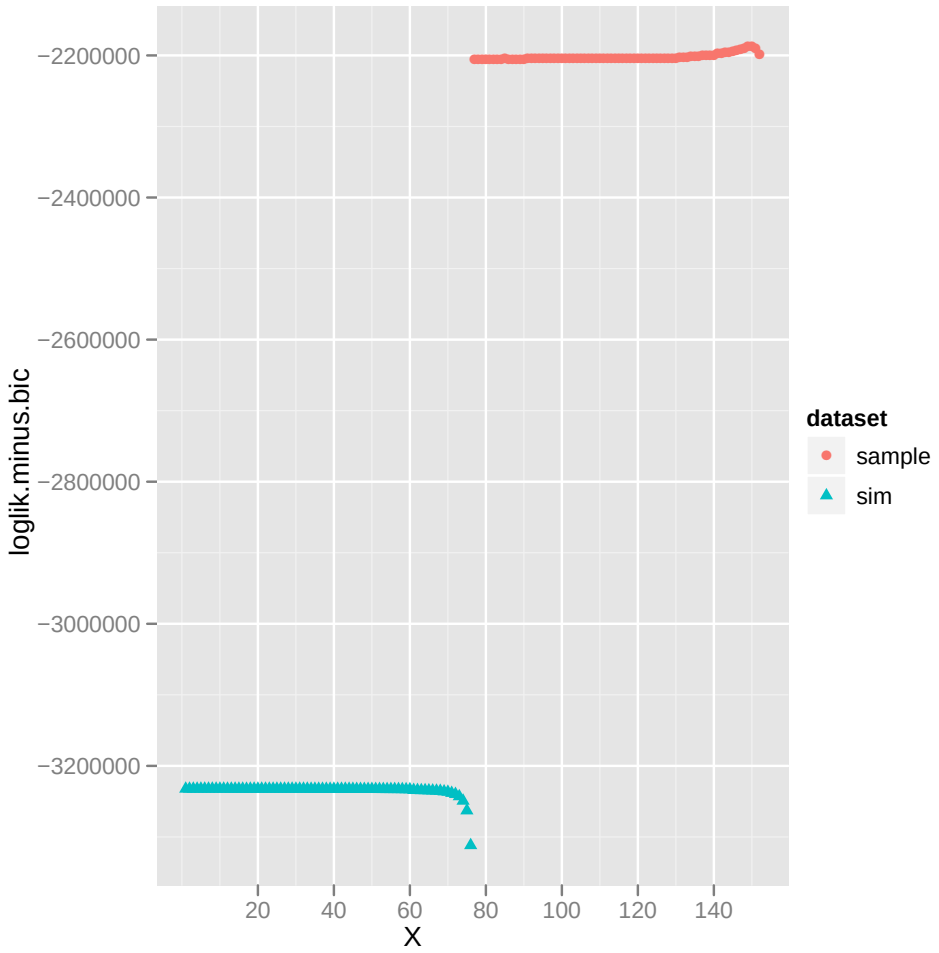
```
> print(ggplot(summ, aes(X, loglik, shape = dataset, color = dataset)) +
+     geom_point())
```




```
> print(ggplot(summ, aes(X, bic, shape = dataset, color = dataset)) +
+     geom_point())
```

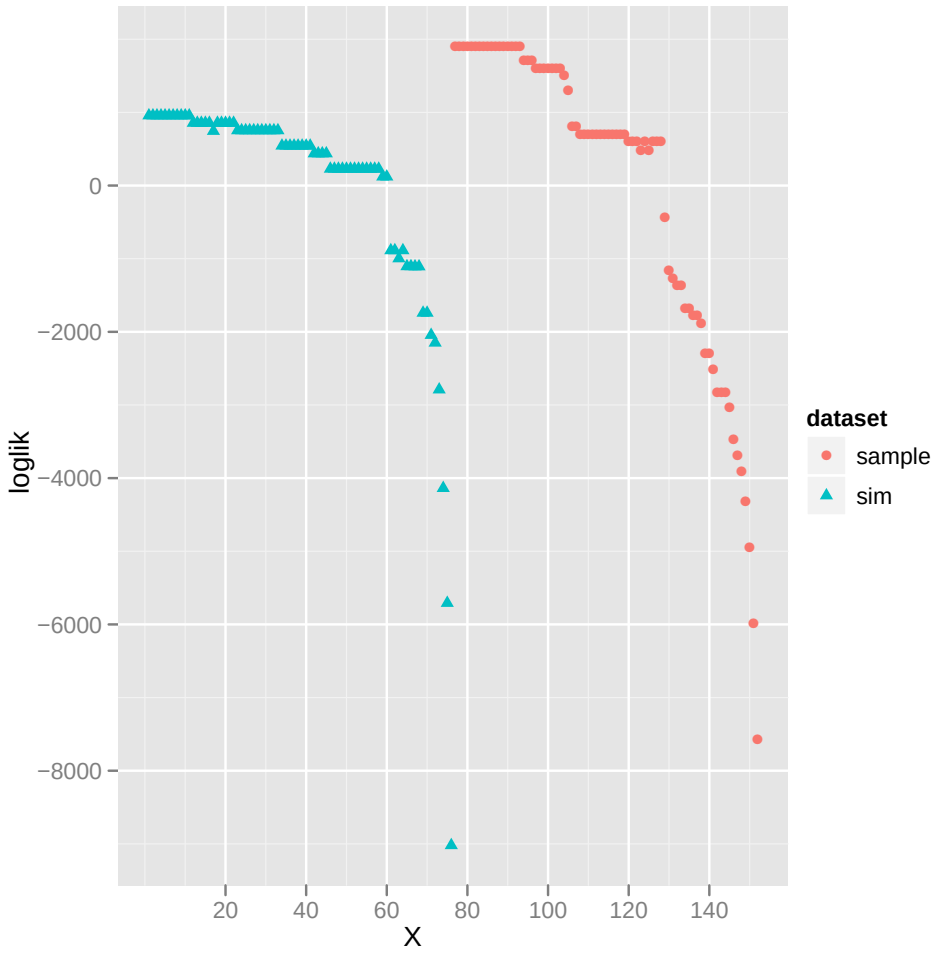



```
> print(ggplot(summ, aes(X, loglik.minus.bic, shape = dataset,
+      color = dataset)) + geom_point())
```

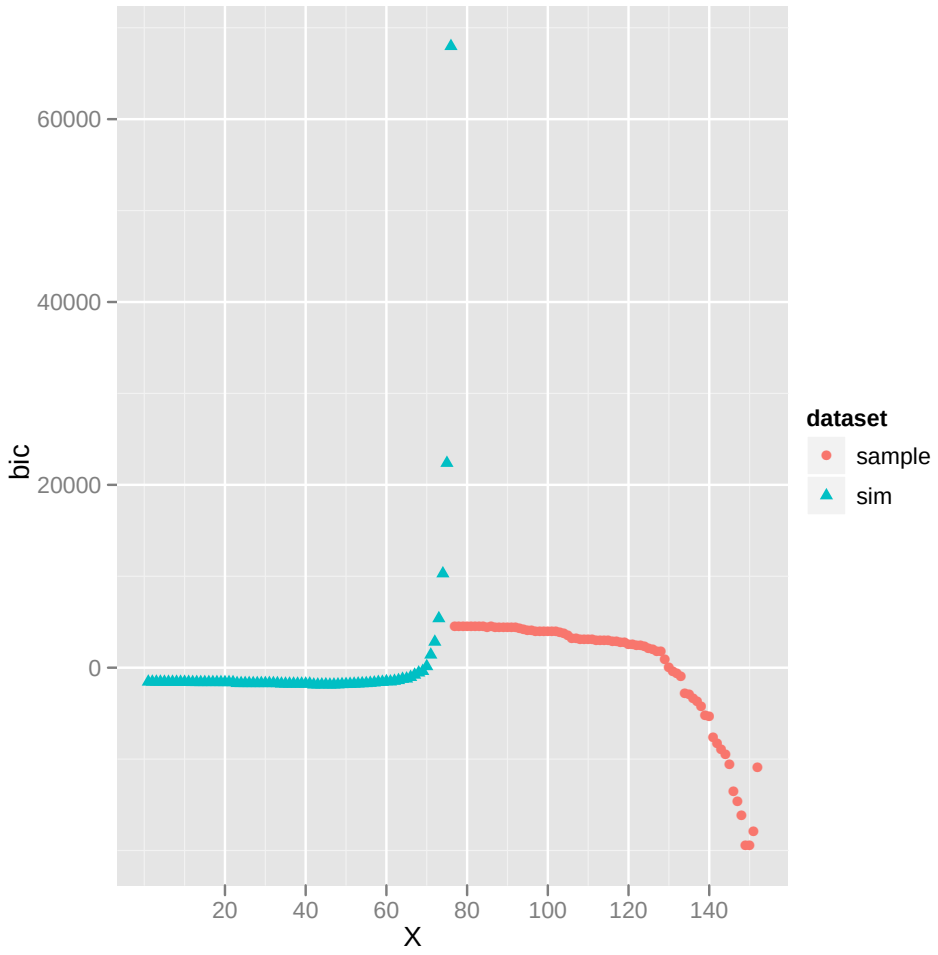


It's sort of hard to see anything due to the large overall differences. Let's center each dataset's scores at 0:

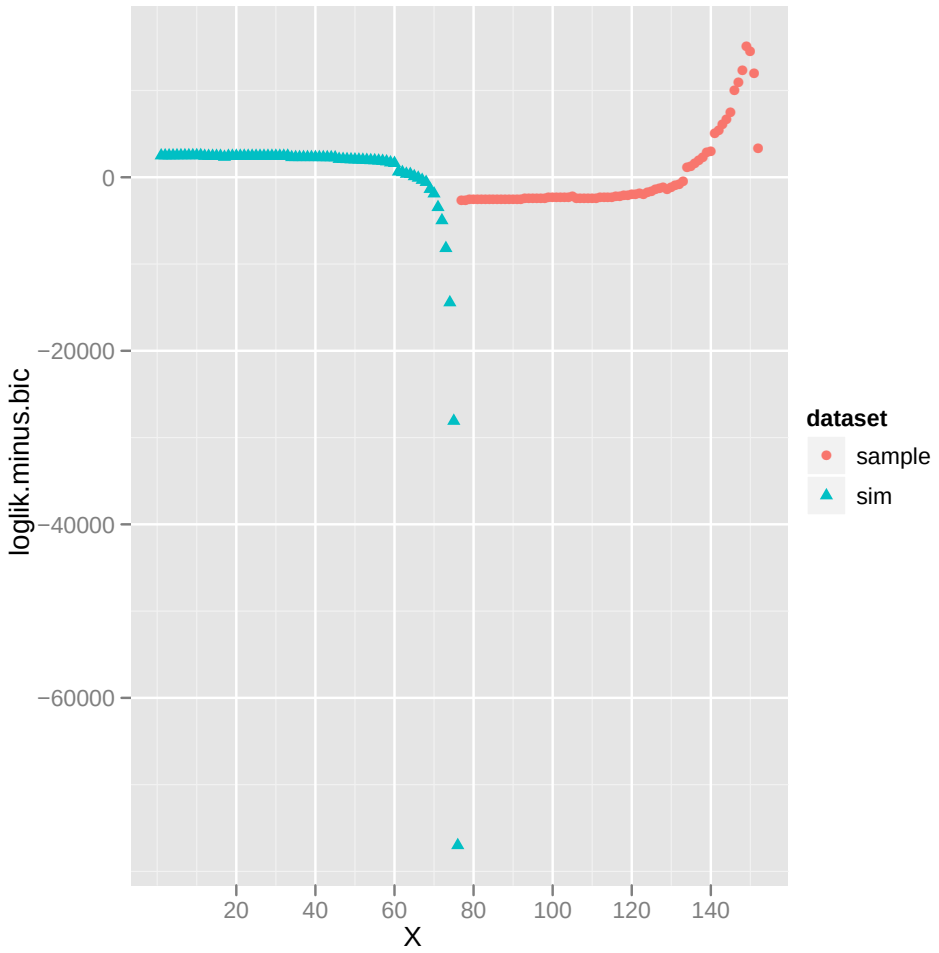
```
> s <- ddply(summ, .(dataset), function(s) {
+     s$loglik <- s$loglik - mean(s$loglik)
+     s
+ })
> print(ggplot(s, aes(X, loglik, shape = dataset, color = dataset)) +
+     geom_point())
```




```
> s <- ddply(summ, .(dataset), function(s) {
+     s$bic <- s$bic - mean(s$bic)
+     s
+ })
> print(ggplot(s, aes(X, bic, shape = dataset, color = dataset)) +
+     geom_point())
```



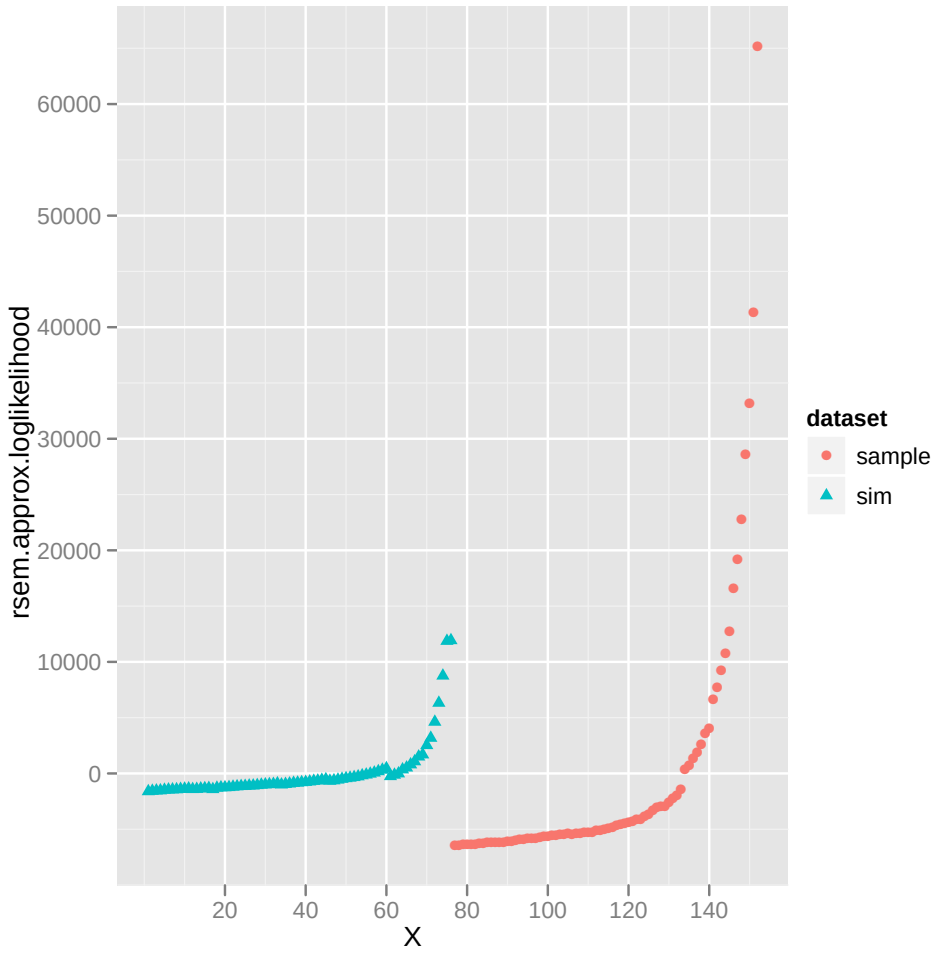

```
> s <- ddply(summ, .(dataset), function(s) {
+     s$loglik.minus.bic <- s$loglik.minus.bic - mean(s$loglik.minus.bic)
+     s
+ })
> print(ggplot(s, aes(X, loglik.minus.bic, shape = dataset, color = dataset)) +
+     geom_point())
```



We can see clearly from these plots that the BIC is different between the two datasets, and this is mainly what causes the difference in penalized likelihood.

Why are the BIC's different? Let's plot the `rsem.approx.loglikelihood`.

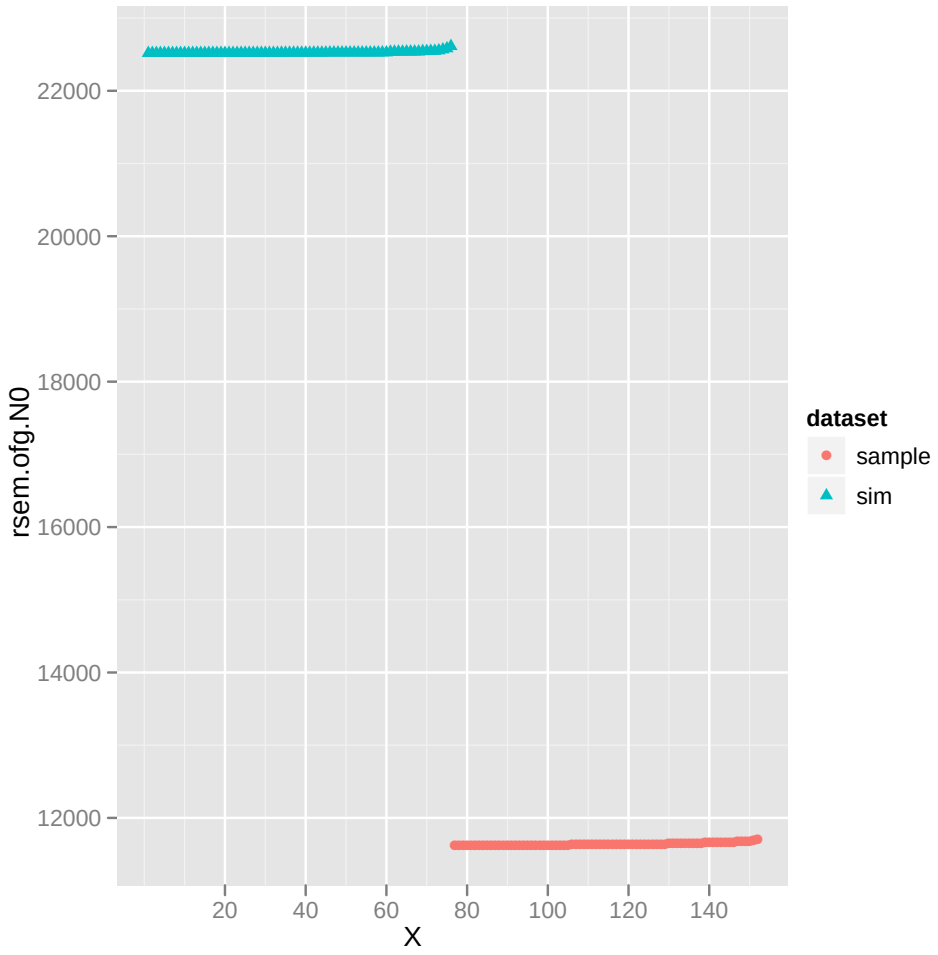
```
> s <- ddply(summ, .(dataset), function(s) {
+     s$rsem.approx.loglikelihood <- s$rsem.approx.loglikelihood -
+         mean(s$rsem.approx.loglikelihood)
+     s
+ })
> print(ggplot(s, aes(X, rsem.approx.loglikelihood, shape = dataset,
+     color = dataset)) + geom_point())
```



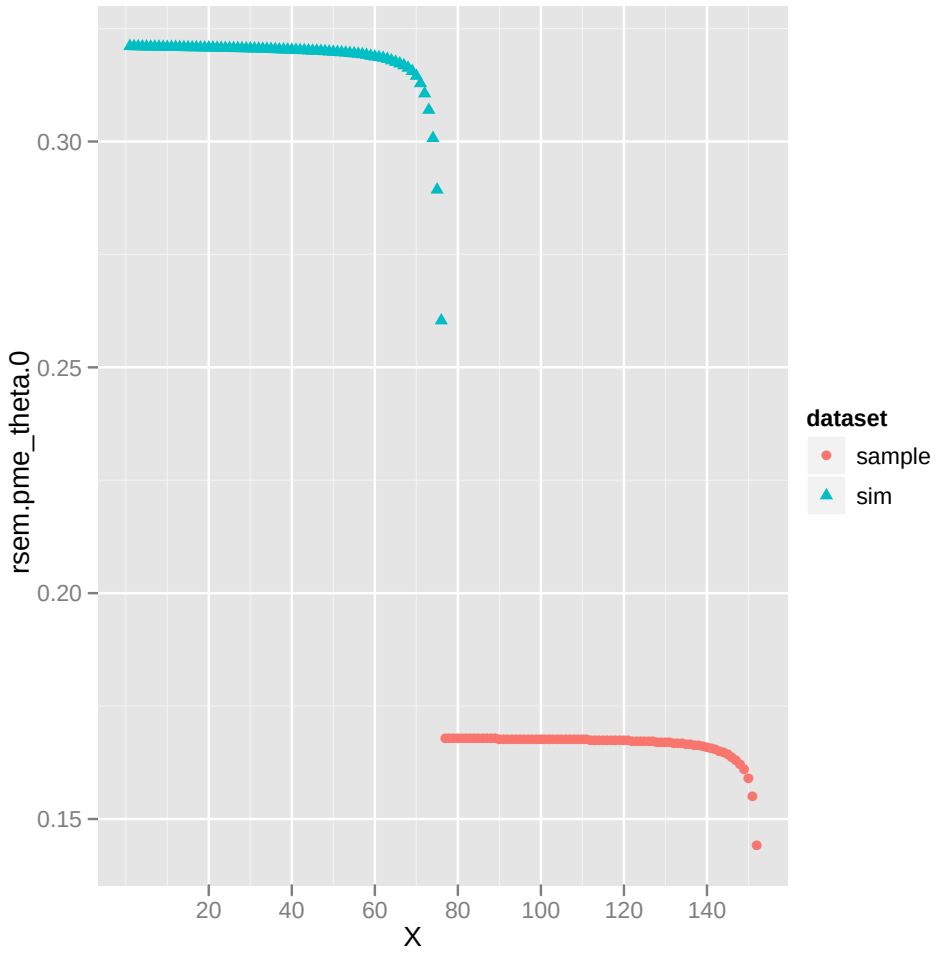
So it's not due to the noise term, but due to differences in the non-noise reads.

Let's also plot N_0 and θ_0:

```
> print(ggplot(summ, aes(X, rsem.ofg.N0, shape = dataset, color = dataset)) +
+     geom_point())
```

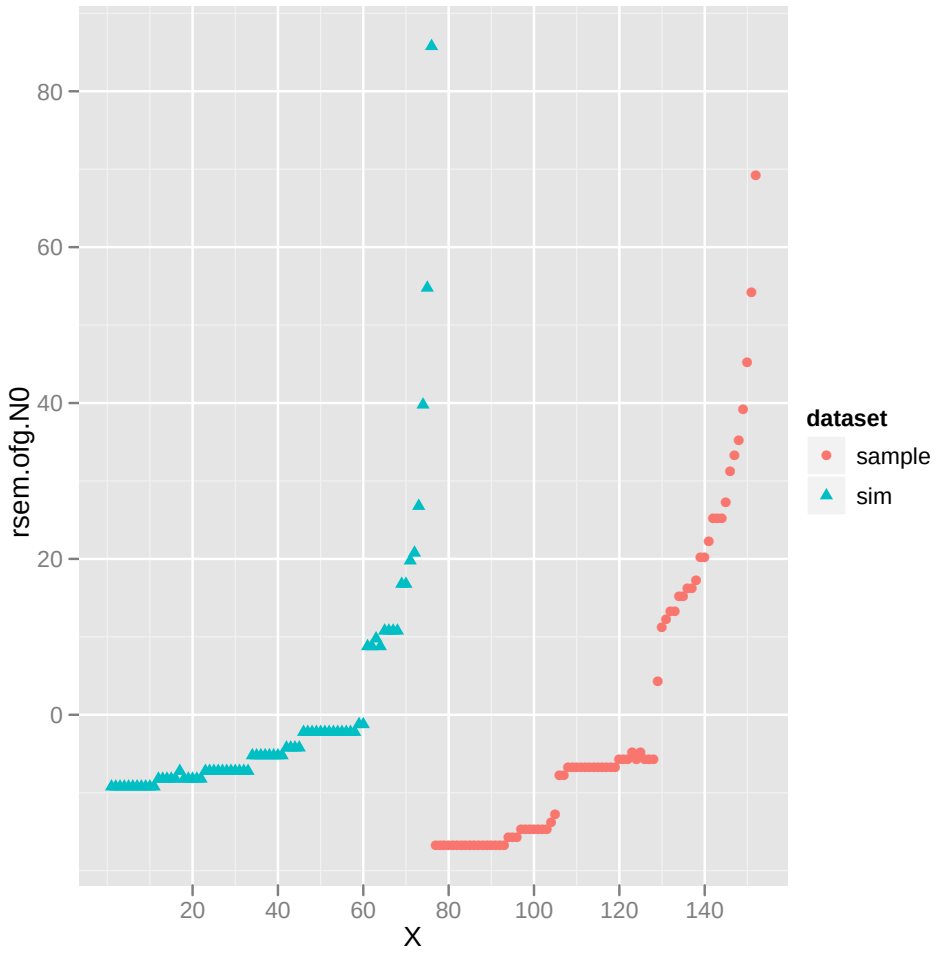



```
> print(ggplot(summ, aes(X, rsem.pme_theta.0, shape = dataset,
+     color = dataset)) + geom_point())
```

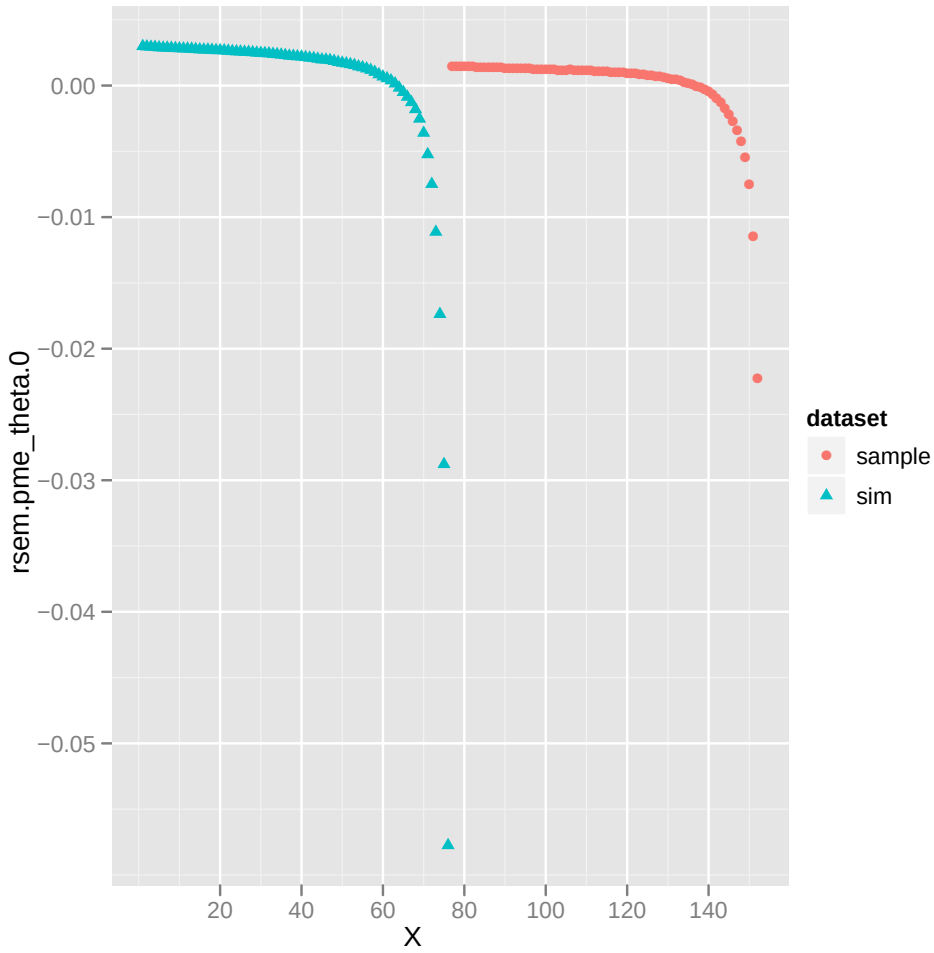


Centered versions:

```
> s <- ddply(summ, .(dataset), function(s) {
+     s$rsem.ofg.N0 <- s$rsem.ofg.N0 - mean(s$rsem.ofg.N0)
+     s
+ })
> print(ggplot(s, aes(X, rsem.ofg.N0, shape = dataset, color = dataset)) +
+     geom_point())
```




```
> s <- ddply(summ, .(dataset), function(s) {
+     s$rsem.pme_theta.0 <- s$rsem.pme_theta.0 - mean(s$rsem.pme_theta.0)
+     s
+ })
> print(ggplot(s, aes(X, rsem.pme_theta.0, shape = dataset, color = dataset)) +
+     geom_point())
```


rsem.pme_theta.0
0.00
−0.01
−0.02
−0.03
−0.04
−0.05
20
40
60
80
100
120
140
X
dataset
sample
sim