The bootstrap

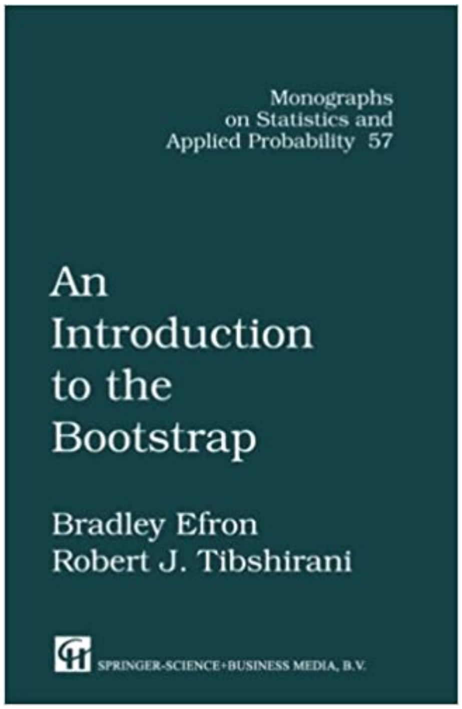
Confidence intervals for QTL location

Karl Broman

Biostatistics & Medical Informatics, UW–Madison

kbroman.org
github.com/kbroman
@kwbroman
Course web: kbroman.org/AdvData

In this lecture, we'll look at the bootstrap; a method to get standard errors and confidence intervals by resampling one's own data. But then we'll proceed to an example where the bootstrap performs terribly.



Monographs
on Statistics and
Applied Probability 57

An
Introduction
to the
Bootstrap

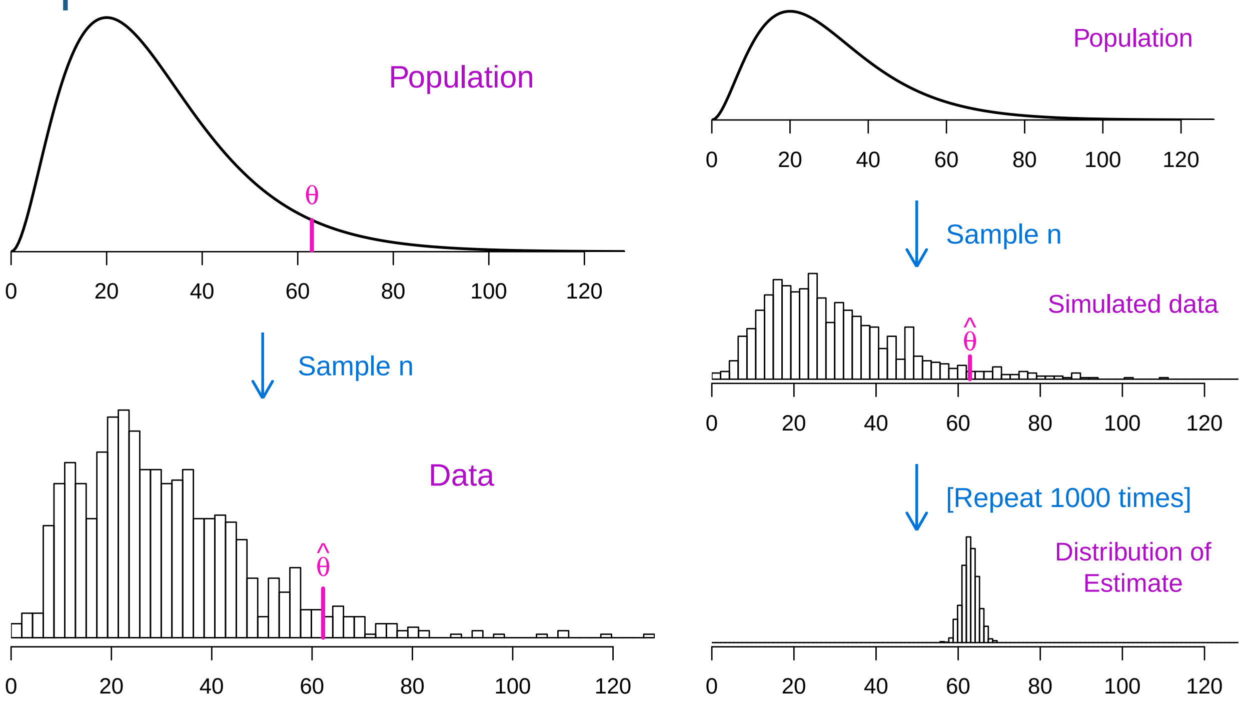
Bradley Efron
Robert J. Tibshirani

SPRINGER-SCIENCE+BUSINESS MEDIA, B.V.

The bootstrap is crazy useful. Brad Efron has a big book about it, but I'd start with this shorter one with Rob Tibshirani.

As I've emphasized before, computer simulation is really useful. The bootstrap can be seen as an extension of that point.

Example

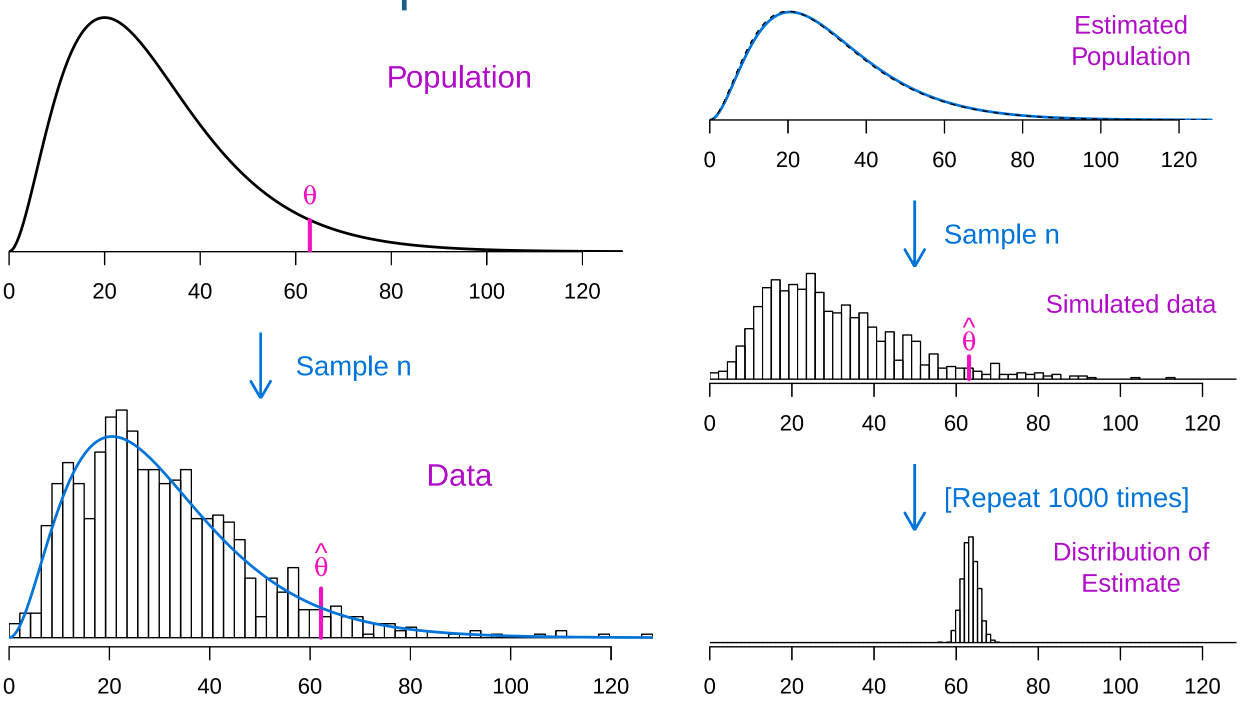


Consider the case that you are seeking to estimate the 95th percentile of some population.

You have data on a random sample with n=1000. What can you say about the standard error of the sample estimate? How could we figure this out?

Well, if we knew the population distribution, we could simulate from it. Simulate a sample of size n=1000, calculate the 95th percentile, and then repeat that a bunch of times. The distribution of the estimates is what we're interested it; the SD is the standard error we're looking for.

Parametric bootstrap

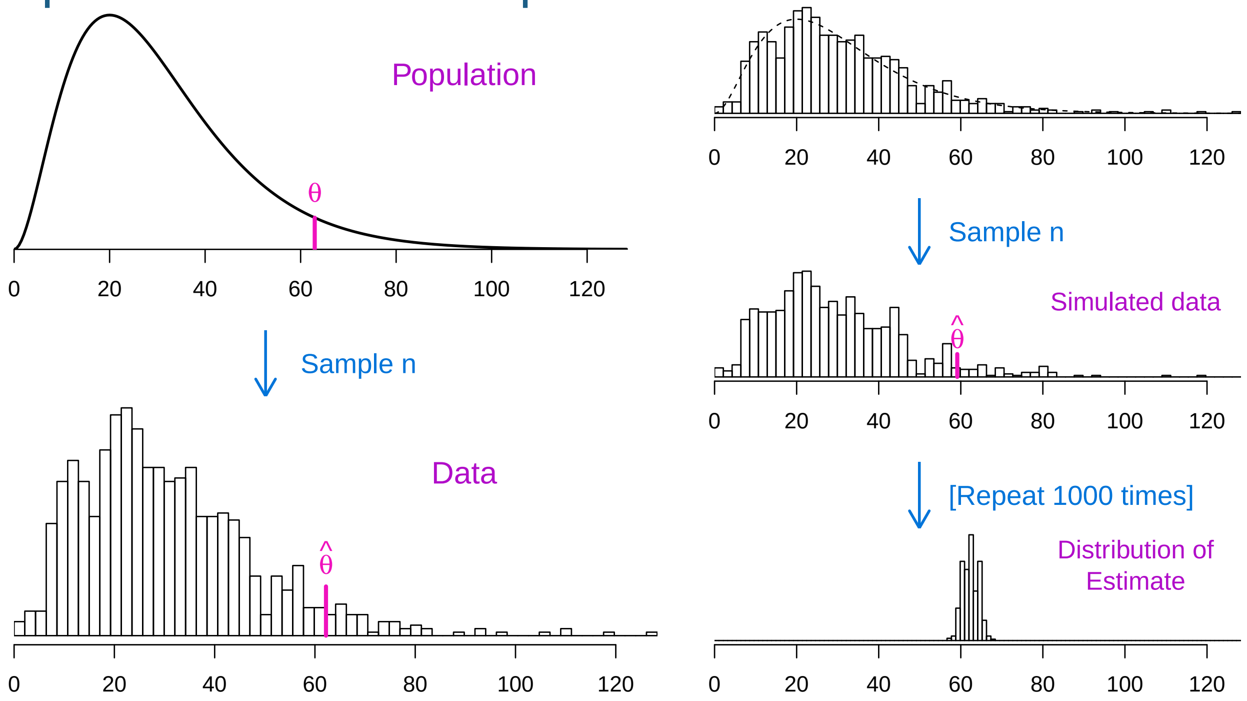


But of course we don't know the population distribution. But we might have a model for it (for example, that it follows a scaled chi-square distribution). We could use our data to estimate the population distribution.

We could then simulate from the estimated population distribution, and determine the distribution (and so the SE) of our estimate in the case that the population follows that estimated distribution.

This is called the parametric bootstrap: you have a model for the underlying population, you use your data to estimate that model, and then you simulate from the model in order to get an understanding of your target estimator.

Non-parametric bootstrap



But if we have enough data, we could use the data itself (the empirical distribution) as an estimate of the population distribution. We could then simulate from that.

In other words, we make draws with replacement from our observed data, to create a new sample of the same size. We calculate our estimate with this re-sampled data, and then repeat many times, to get an estimate of the distribution of the estimator.

This is called the non-parametric bootstrap: skip modeling the underlying population and just use your data to approximate the population, then simulate (re-sample) from that empirical distribution in order to get an understanding of your target estimator.

Don’t think “re-sampling”

Think “simulate from an estimate of the population”

Folks often focus on the re-sampling aspect of the bootstrap. But I think it’s better to focus on estimating the population distribution and then simulating from that estimate.

The key idea in the bootstrap is not resampling, but estimating the population distribution with the empirical distribution.

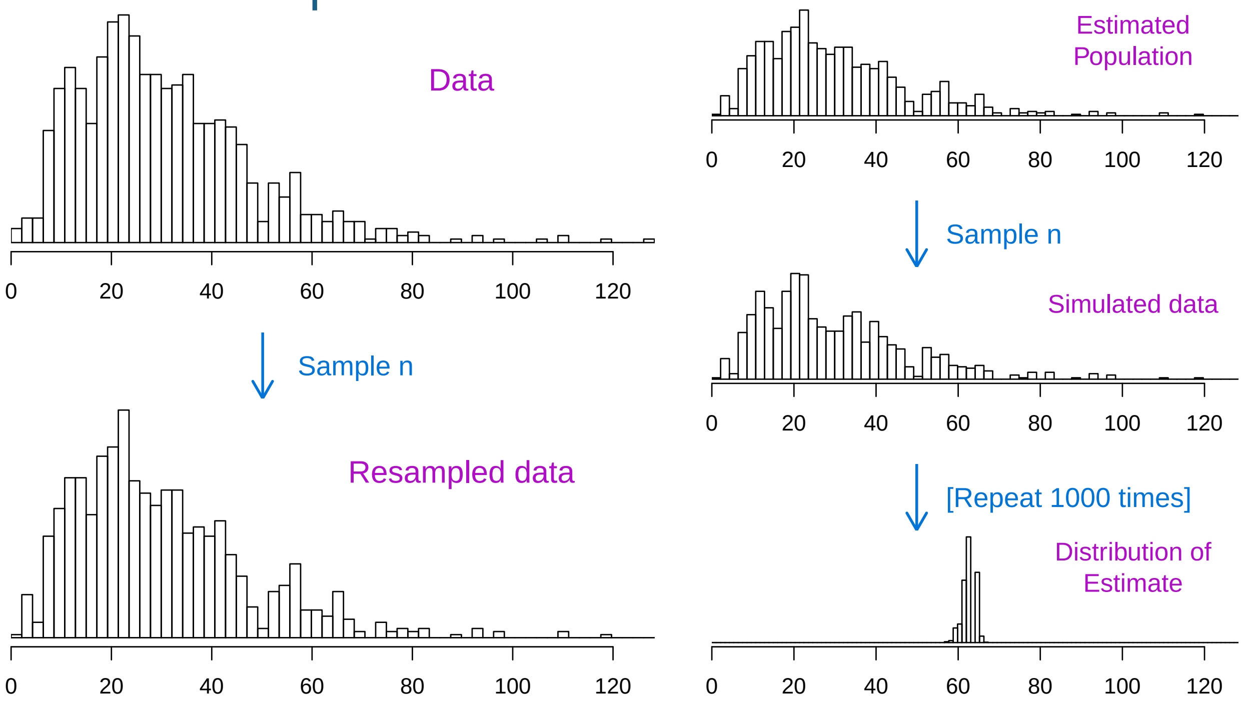
How can we tell if the bootstrap works?

Simulate!

The bootstrap works well in lots of cases. It's dependent on having a reasonably large dataset. And its performance depends on the nature of the estimate.

How can we tell if it works? Well, we could simulate. Basically, we could create a nested bootstrap.

Nested bootstrap

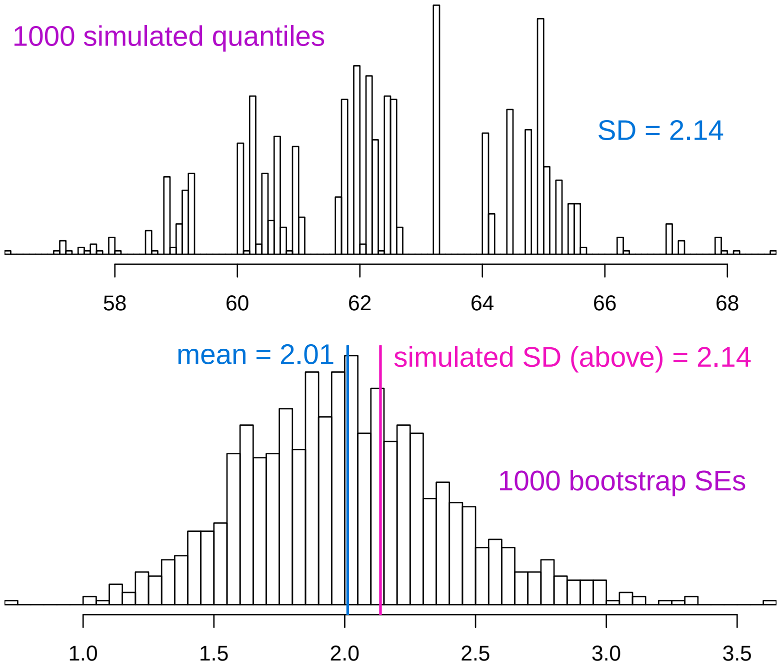


The idea is to treat your data as the true population distribution. Sample from it and you have some data.

Then take that data and apply the bootstrap method to it: treat it as the population distribution (or fit a model to get an estimated population distribution) and then simulate from it, calculate your estimate, and repeat many times. So that gives you one bootstrap-based SE.

You then go back and get a new sample of data from your observed data, and then apply the bootstrap again, and then repeat many times.

Nested bootstrap results



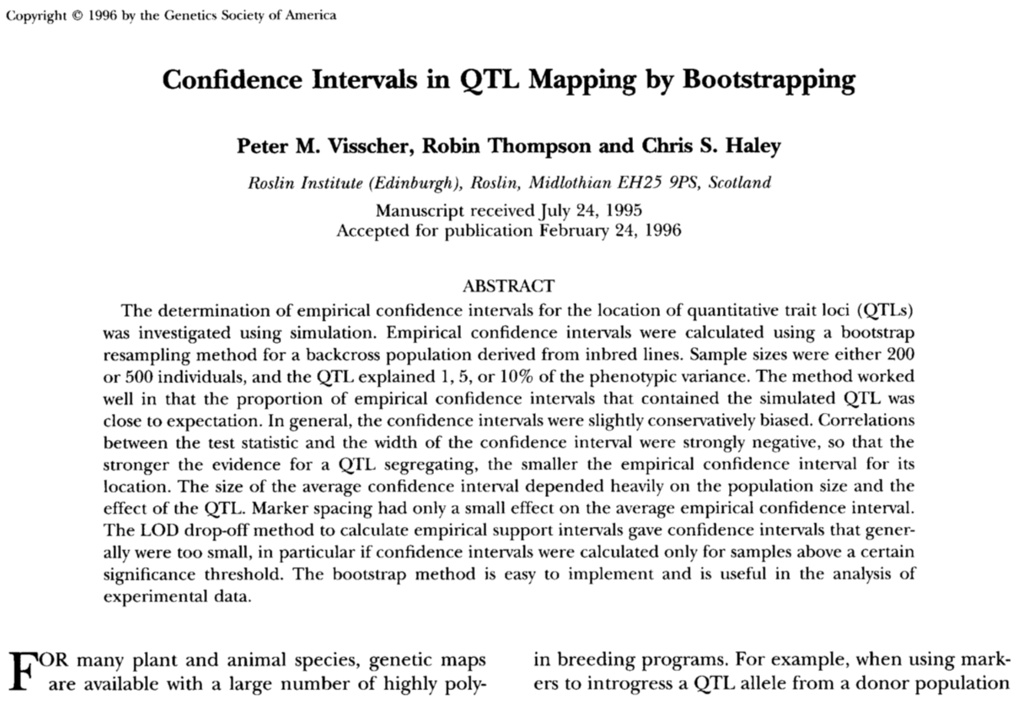
The result of this is you have a bunch of estimates of your parameter, for different samples from your data, plus you have a bootstrap-based SE for each.

On the top panel here, I show the estimated quantiles from 1000 samples from my data. The odd multimodal features are due to the discrete nature of my original data. The SD of these estimates is 2.14. That's sort of the "real" standard error of my estimate.

In the lower panel, I show the estimated SE (by the bootstrap) from each of 1000 samples from my original data. The average SE is 2.01, which is a bit smaller than the SD of the top histogram.

So the bootstrap is slightly under-estimating the SE, and it's not particularly well estimated (with an SD of like 0.5). But we can see that it is not too badly behaved.

Note that you could do all of this, looking at confidence intervals. And you might measure something like their coverage, and maybe also their width.



Confidence Intervals in QTL Mapping by Bootstrapping

Peter M. Visscher, Robin Thompson and Chris S. Haley

Roslin Institute (Edinburgh), Roslin, Midlothian EH25 9PS, Scotland

Manuscript received July 24, 1995
Accepted for publication February 24, 1996

ABSTRACT

The determination of empirical confidence intervals for the location of quantitative trait loci (QTLs) was investigated using simulation. Empirical confidence intervals were calculated using a bootstrap resampling method for a backcross population derived from inbred lines. Sample sizes were either 200 or 500 individuals, and the QTL explained 1, 5, or 10% of the phenotypic variance. The method worked well in that the proportion of empirical confidence intervals that contained the simulated QTL was close to expectation. In general, the confidence intervals were slightly conservatively biased. Correlations between the test statistic and the width of the confidence interval were strongly negative, so that the stronger the evidence for a QTL segregating, the smaller the empirical confidence interval for its location. The size of the average confidence interval depended heavily on the population size and the effect of the QTL. Marker spacing had only a small effect on the average empirical confidence interval. The LOD drop-off method to calculate empirical support intervals gave confidence intervals that generally were too small, in particular if confidence intervals were calculated only for samples above a certain significance threshold. The bootstrap method is easy to implement and is useful in the analysis of experimental data.

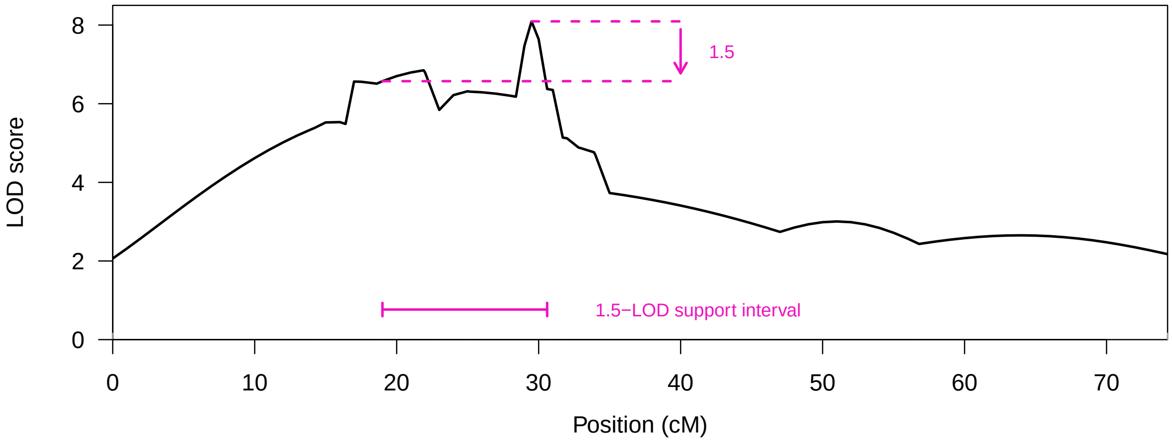
FOR many plant and animal species, genetic maps are available with a large number of highly poly-

in breeding programs. For example, when using markers to introgress a QTL allele from a donor population

When you have mapped a QTL (a genetic locus that affects a quantitative trait), it is important to establish a confidence interval for the location of the QTL. This can guide further experiments: how precisely have you mapped the locus, and are there any reasonable candidate genes in the region.

Peter Visscher and colleagues suggested using the bootstrap to get such a confidence interval. Lots of people were using it.

LOD support interval



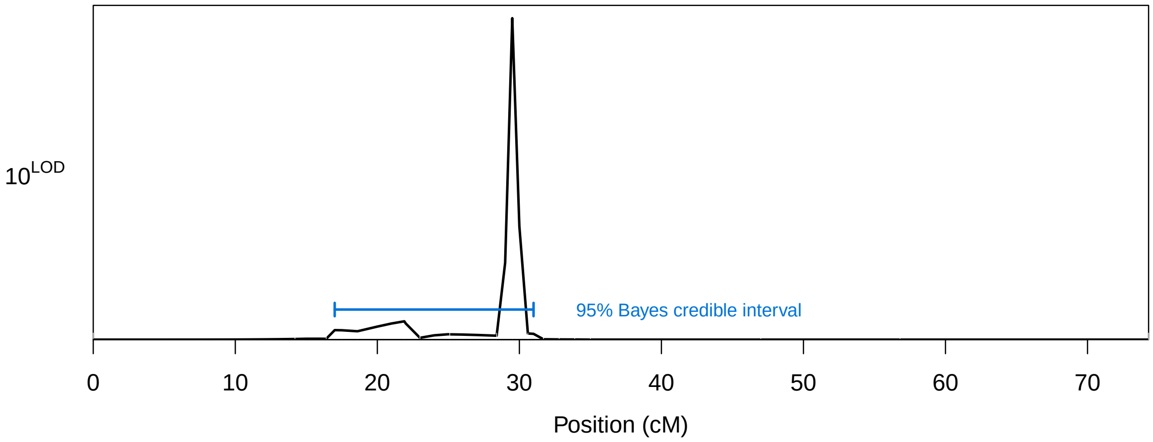
Data from Sugiyama et al. (2001) Genomics 71:70–77

The usual method of getting a confidence interval for QTL location was the LOD support interval: drop down from the maximum likelihood some fixed amount. In other words, look at the region where the LOD score is within some amount of its maximum.

This ends up being the traditional sort of “invert the hypothesis test” approach for constructing a confidence interval, as the difference in LOD scores is a $\log_{10}$ likelihood ratio.

But various simulation studies showed that this doesn’t behave as a real confidence interval. Coverage depends on a bunch of things, including the strength of the QTL effect, but also the type of cross and the relative locations of the markers to the QTL.

Approximate Bayes interval

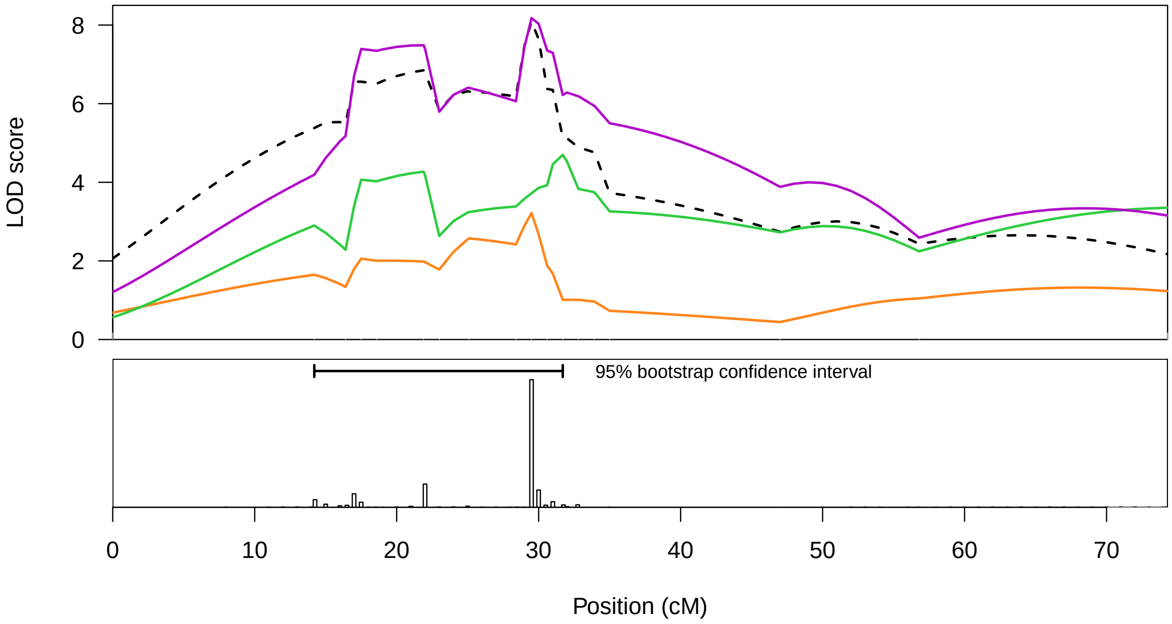


Data from Sugiyama et al. (2001) Genomics 71:70–77

Another approach is an approximate Bayes interval. Basically take 10^{LOD} and treat it as if it were a real likelihood. Assume that the QTL location is uniform along the chromosome (as the prior distribution), and so re-scale the likelihood so that it integrates to 1, and you have the posterior. Drop down some amount from the maximum until you get a region that covers 95% of the area, and take that as a 95% Bayes credible interval.

This hadn't been much studied, but it was one of the other approaches that I had considered using.

Bootstrap CI



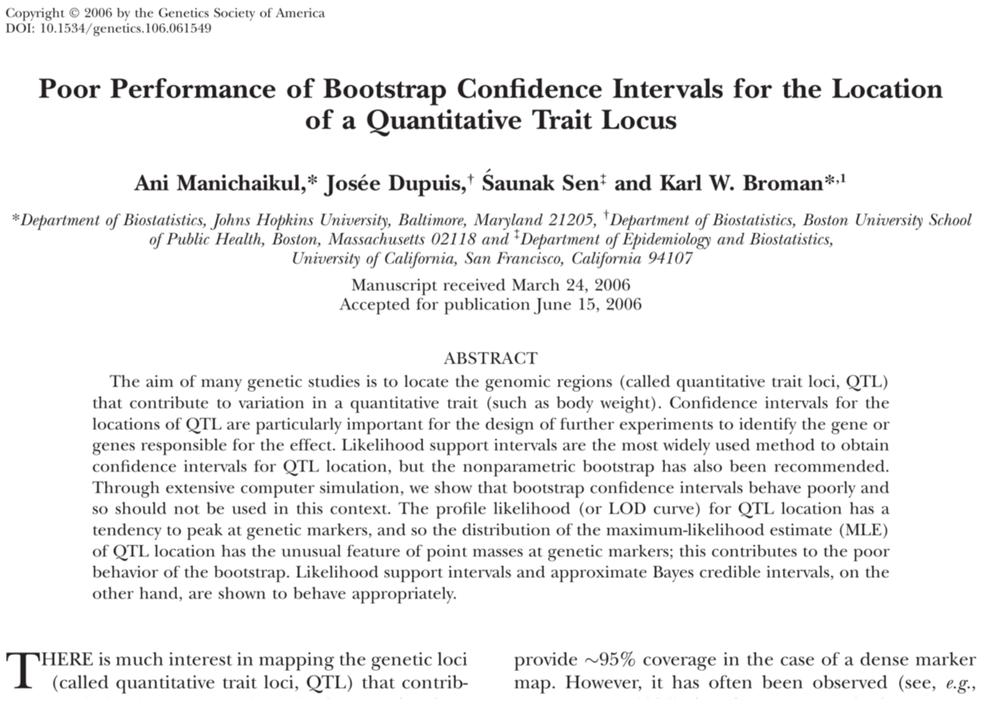
Data from Sugiyama et al. (2001) Genomics 71:70–77

Well, the thing is that the bootstrap in this context tended to behave rather strangely. Here are the bootstrap results for this dataset. They don't look quite right. I mean, they don't really reflect what you might perceive to be the uncertainty in QTL location.

(Note that in the top graph, the dashed curve is the LOD curve with the original data. The three colored curves are results with different bootstrap samples.)

The fact is, these LOD curves tend to achieve their maximum at the markers, and as a result, the bootstrap results have spikes at the marker positions, which makes the whole thing behave a bit wonky.

I looked at these results and thought, "We really should investigate this bootstrap thing. Does it really work here?"



DOI: 10.1534/genetics.106.061549

Poor Performance of Bootstrap Confidence Intervals for the Location of a Quantitative Trait Locus

Ani Manichaikul,* Josée Dupuis,† Śaunak Sen‡ and Karl W. Broman*,1

**Department of Biostatistics, Johns Hopkins University, Baltimore, Maryland 21205, †Department of Biostatistics, Boston University School of Public Health, Boston, Massachusetts 02118 and ‡Department of Epidemiology and Biostatistics, University of California, San Francisco, California 94107*

Manuscript received March 24, 2006
Accepted for publication June 15, 2006

ABSTRACT

The aim of many genetic studies is to locate the genomic regions (called quantitative trait loci, QTL) that contribute to variation in a quantitative trait (such as body weight). Confidence intervals for the locations of QTL are particularly important for the design of further experiments to identify the gene or genes responsible for the effect. Likelihood support intervals are the most widely used method to obtain confidence intervals for QTL location, but the nonparametric bootstrap has also been recommended. Through extensive computer simulation, we show that bootstrap confidence intervals behave poorly and so should not be used in this context. The profile likelihood (or LOD curve) for QTL location has a tendency to peak at genetic markers, and so the distribution of the maximum-likelihood estimate (MLE) of QTL location has the unusual feature of point masses at genetic markers; this contributes to the poor behavior of the bootstrap. Likelihood support intervals and approximate Bayes credible intervals, on the other hand, are shown to behave appropriately.

THERE is much interest in mapping the genetic loci (called quantitative trait loci, QTL) that contrib-

provide $\sim$95% coverage in the case of a dense marker map. However, it has often been observed (see, *e.g.*,

We ended up writing this paper. The results are well summarized in the title. Nevertheless, Visscher's paper has $> 3\times$ as many citations as mine. ;)

Simulation study

- Backcross, 200 individuals
- One chromosome of length 100 cM
- Markers at 10 cM spacing
- Single QTL responsible for 10% of phenotypic variance
- Normally distributed residual variation
- Varied location of QTL, at positions 0, 1, …, 100 cM
- Analysis by standard interval mapping; calculations every 1 cM
- 10,000 simulations for each QTL position
- Bootstrap used 1000 replicates

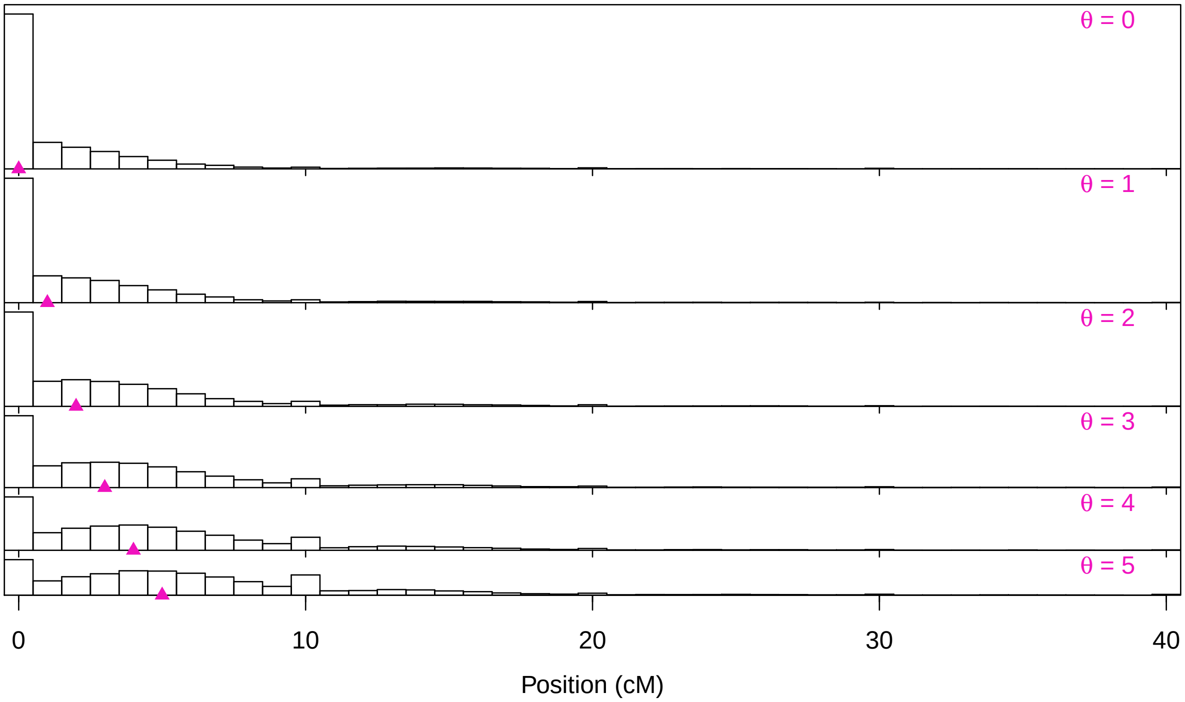
So what we did was a simulation study to see how well the bootstrap worked in this case.

We considered a backcross (so there are two possible genotypes) and a single chromosome with equally-spaced markers.

We simulated a single QTL and varied its location along the chromosome.

For each QTL location, we did 10,000 simulation replicates. For each replicate, we did the QTL analysis and calculated three possible QTL intervals, including a bootstrap with 1000 replicates.

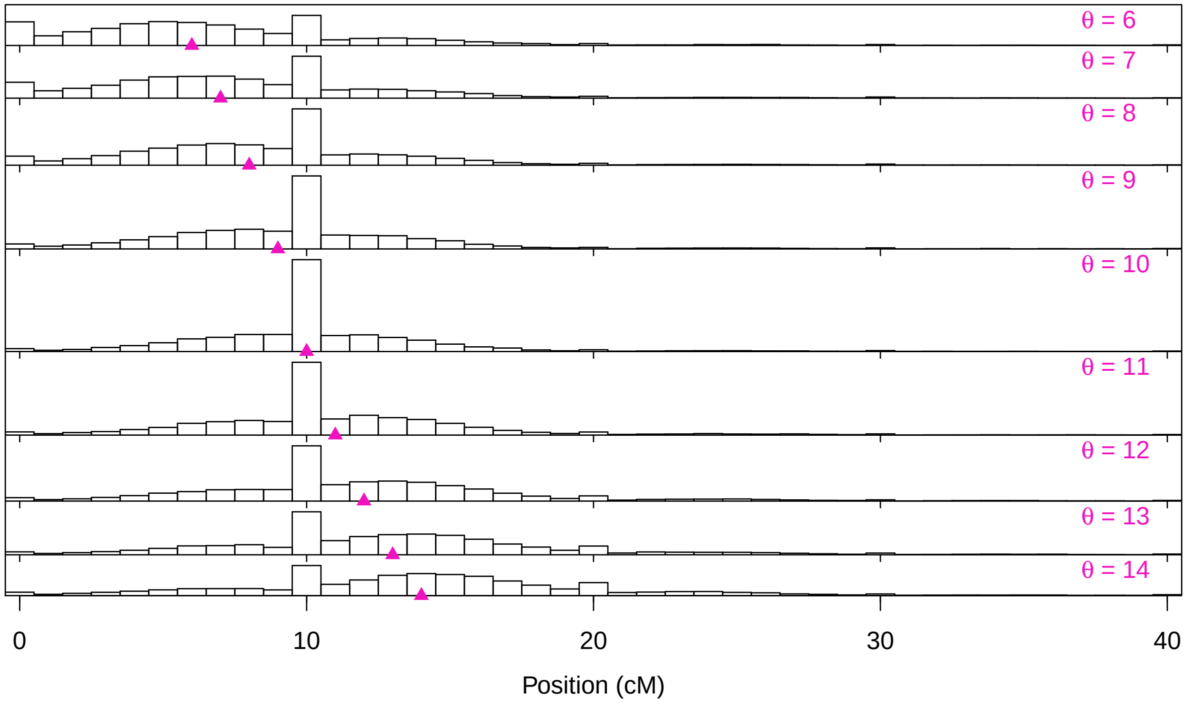
Distribution of estimated location



The first thing to look at is just the distribution of the estimated QTL locations. In these panels, the triangle is indicating the location of the QTL, and the histogram shows the distribution of its estimated location. Remember that the markers are placed every 10 cM.

There's a clear tendency for the estimated location to be at a marker. When the QTL is right next to a marker, still the estimate ends up being mostly at the marker.

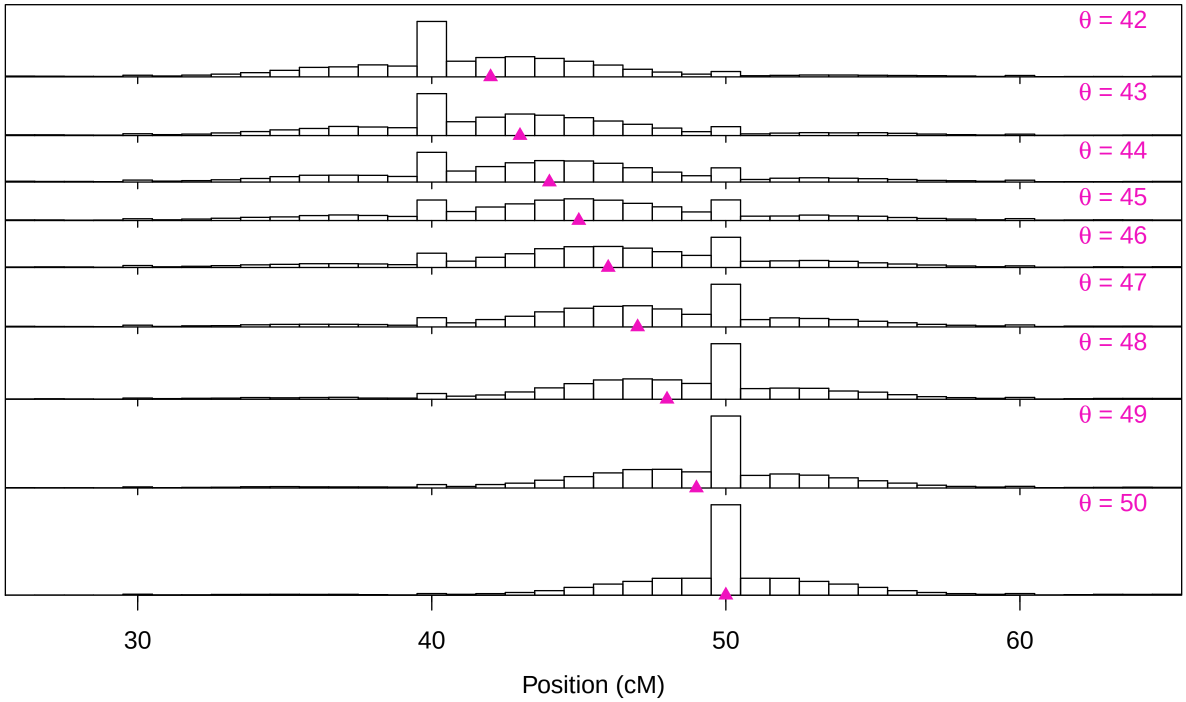
Distribution of estimated location



Here we move the QTL to the right a bit, at 6 - 14 cM. You again see the very strong tendency for the QTL to be at a marker.

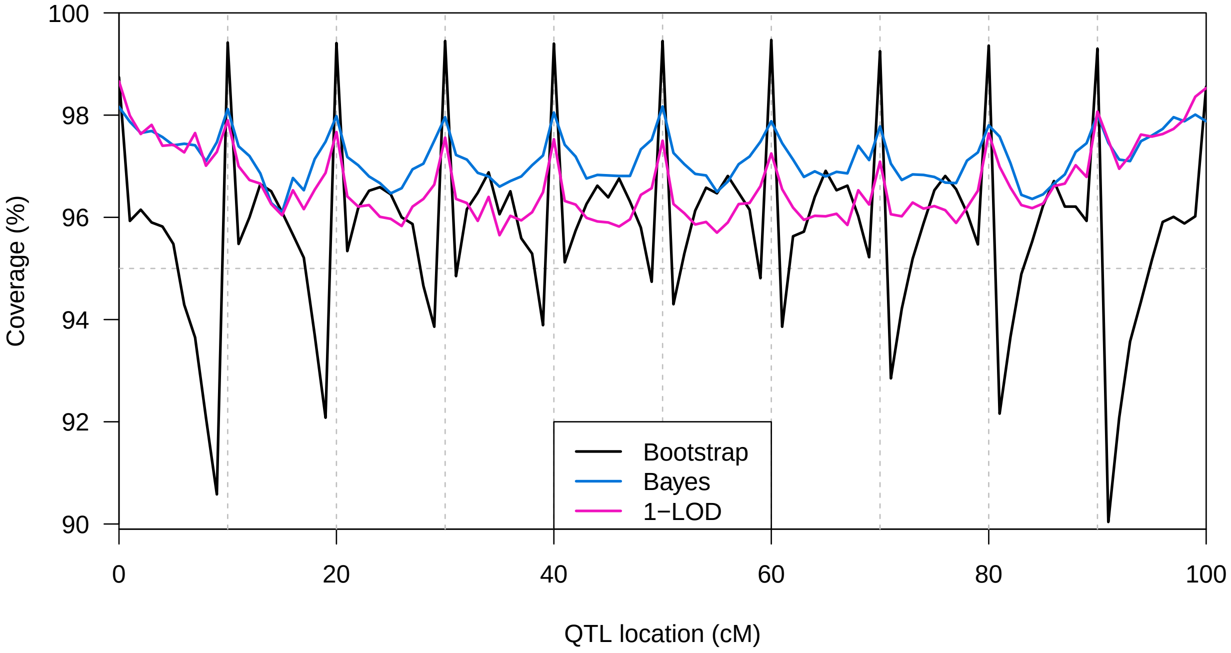
It's not that the estimated QTL position is biased, though that is sort of a problem here at the end of the chromosome. Rather, it's just an odd spike in the sampling distribution that is due to cusps in the likelihood function at the markers.

Distribution of estimated location



Here the QTL is near the center of the chromosome, but in the interval to the left. When the QTL is in an interval between markers, the estimate is somewhat more likely to be at one of the two flanking markers.

Coverage vs. true location



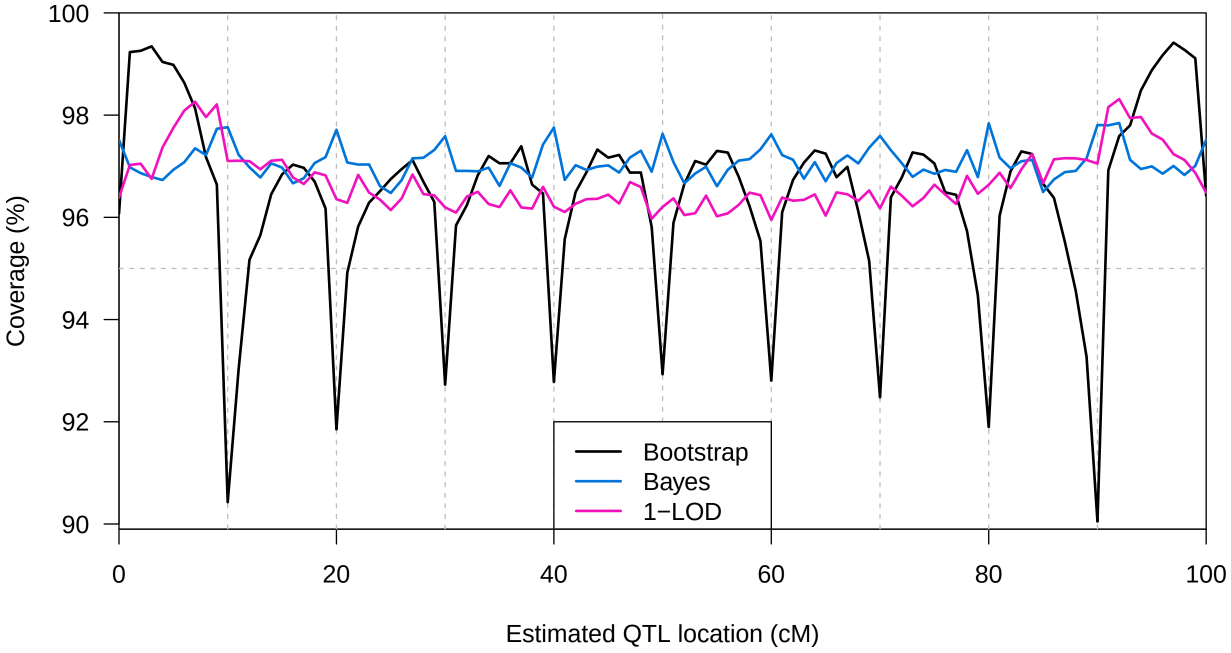
Here now is the plot of the coverage of the three different types of intervals as a function of the location of the QTL.

For each interval type, there is some variation in coverage depending on the true parameter value, with coverage being higher if the QTL is at a marker and lower if it is between markers.

The bootstrap shows the greatest variation in coverage, with coverage being particularly high when the QTL is right at a marker, but being particularly low when it is just adjacent to a marker.

To me, this variation in coverage is a killer for the bootstrap.

Coverage vs. estimated location



We had this idea to also look at coverage relative to the estimated location of the QTL, pooling results with different true QTL positions.

The idea is that, for each simulation replicate, we have some QTL location and some estimated QTL location and an indicator of whether the interval covered the truth or not. In the previous graph, we binned replicates by the true QTL location and found the coverage rate in each bin.

Here, though, we are binning the replicates by the estimated location. We are looking at the coverage in bins defined by the estimated location.

This is rather unorthodox, but is super revealing, as you see that the coverage of the bootstrap intervals depends most critically on the estimated QTL location: if the QTL is estimated to be at a marker, the interval coverage is particularly low. The other two types of intervals show much more stable coverage here.

Summary

- The bootstrap can be super useful
- But it can also behave badly

  You need the distribution of $\hat{\theta} - \theta$ to not depend on θ

- If results look wonky, maybe you shouldn't trust them
- How to tell if the bootstrap works? Simulate!
- The odd tendency for the estimated QTL location to be at a marker messes up the bootstrap.

I like summaries.