Bayesian analysis

Identifying essential genes by mutagenesis

Karl Broman

Biostatistics & Medical Informatics, UW–Madison

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

In this lecture, I'll present a case study on random transposon mutagenesis in Mycobacterium tuberculosis. Traditional frequentist methods are not suitable for this problem, and so I resorted to using Bayesian statistics, with Markov chain Monte Carlo.

Mycobacterium tuberculosis

- The organism that causes tuberculosis.
  - Cost for treatment: ∼$15,000
  - Other bacterial pneumonias: ∼$35
- 4.4 Mbp circular genome, completely sequenced
- 4250 known or inferred genes

Tuberculosis can be surprisingly difficult and expensive to treat. So there's good reason to try to better understand its genome, to identify potential targets for new drugs.

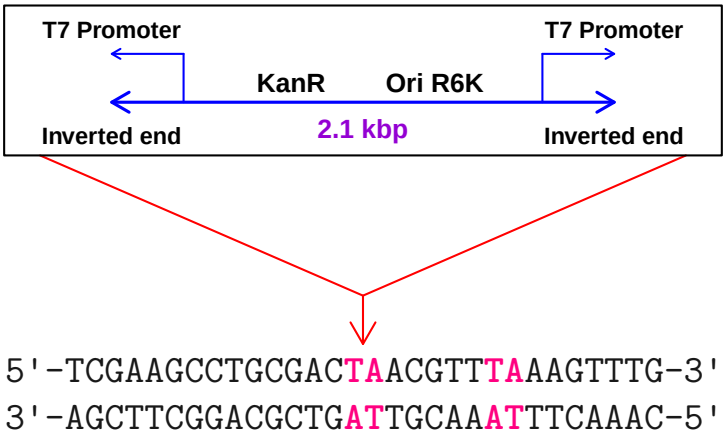
Goal: identify the essential genes

Method: random transposon mutagenesis

The current project seeks to identify the essential genes in the M. tuberculosis genome; the ones that if you knock them out, you get a non-viable mutant.

This is done by random transposon mutageneis. A transposon is a bit of DNA that likes to insert itself into other DNA. So we'll randomly disrupt genes to find out which ones matter and which ones don't so much.

Himar1 transposon



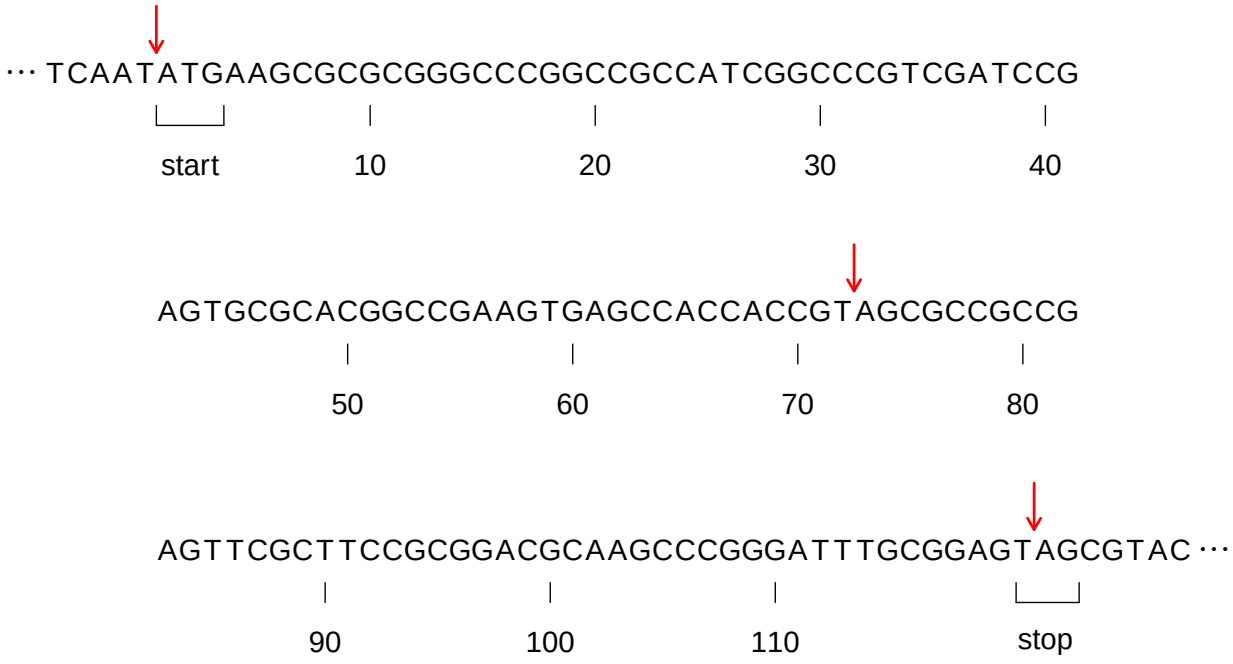
5'-TCGAAGCCTGCGACTAACGTTTAAAGTTTG-3'
3'-AGCTTCGGACGCTGATTGCAAATTTCAAAC-5'

Note: $\geq$ 30 stop codons in each reading frame

They used the Himar1 transposon which randomly inserts itself into a site TA.

"Stop codons" are codes for where a gene stops. The point is that if this thing gets inserted into a gene, it will end up with a truncated product.

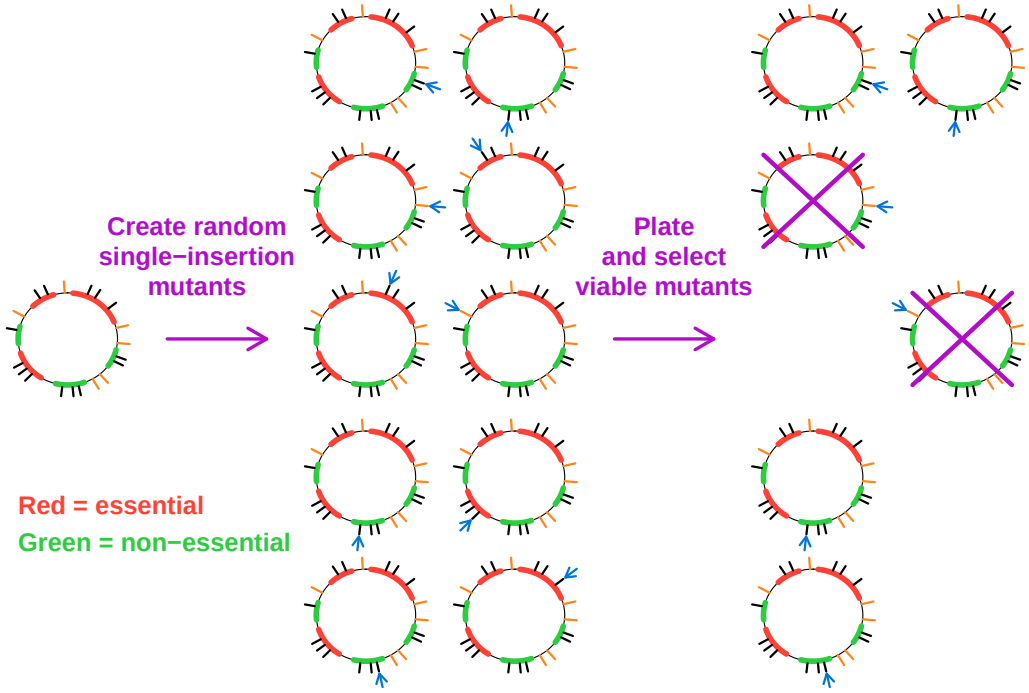
Sequence of the gene MT598



Here's the sequence of one particular gene. It contains 3 transposon insertion sites, one just before the start of the gene, one at around 72 bp, and one in the stop codon.

Transposon insertion at middle site will surely disrupt the gene. But insertion at the site in the stop codon would cause a small increase in the protein which might not actually matter. Insertion at the first site may not affect the protein product but would separate the gene from its promotor.

Random transposon mutagenesis



Here's a schematic for how random transposon mutagenesis works. Imagine a circular genome with just six genes, three essential (in red) and three non-essential (in green). The tick marks are the transposon insertion sites; yellow ticks are sites in-between genes.

We first create a set of mutants with a single transposon insertion and select for viable mutants. Insertions in essential genes (red) will be non-viable and won't grow. We then sequence the mutants to identify the insertion sites. We'll ignore mutants where the insertion was in-between genes.

Genes with viable insertions are then known to be non-essential. The essential genes are the ones where, as you increase the size of the mutant library, you never see a viable mutant.

Random transposon mutagenesis

- Location of transposon insertion determined by sequencing across junctions
- Viable insertion within a gene $\Longrightarrow$ gene is non-essential
- Essential genes: we will never see a viable insertion
- Complication: Insertions in the very distal portion of an essential gene may not be sufficiently disruptive.
  Thus, we omit from consideration insertions sites within the last 20% and last 100 bp of a gene.

So again, we identify the location of the insertion. A gene with a viable insertion mutant is non-essential; we are interested in trying to predict the essential ones.

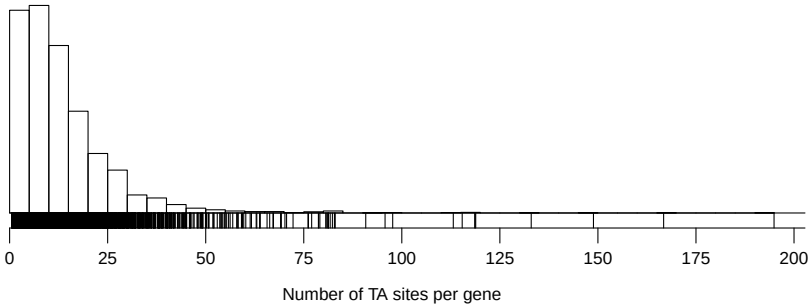
There's this complication that insertion in the very distal portion of an essential gene may not be sufficiently disruptive. So we used a rule where we just ignore sites and mutants in the last portion of the gene.

The data

- Number, locations of genes
- Number of insertion sites in each gene
- n viable mutants with exactly one transposon insertion
- Location of the transposon insertion in each mutant

Our data consists of the number and locations of genes, the number and locations of the insertion sites in each gene, and then we have n viable mutants with exactly one traponson insertion, for which we know the insertion site.

TA sites in M. tuberculosis



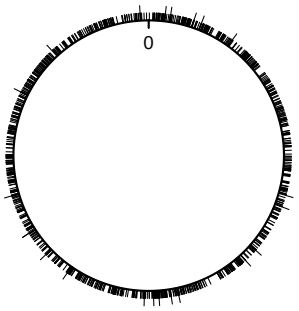
- 74,403 sites
- 65,649 sites within a gene
- 57,934 sites within proximal portion of a gene
- 4204/4250 genes with at least one TA site

For the Mycobacterium tuberculosis genome, there are 74k insertion sites: 66k within a gene, and 58k within the proximal portion of the gene that we'll consider.

Of the 4250 genes, 4204 have at least one insertion site.

The histogram shows the number of insertion sites. Typically about 10; max near 200.

1425 insertion mutants



- 1425 insertion mutants
- 1025 within proximal portion of a gene
- 21 double-hits
- 770 unique genes hit

Questions: Proportion of essential genes in M. tuberculosis?

Which genes are likely essential?

The transposon mutagenesis data includes 1425 insertion mutants, of which 1025 are within the proximal portion of a gene. There were 770 unique genes hit. 21 specific sites were hit twice.

The questions are: what is the proportion of essential genes? Which genes are likely essential?

Model

Transposon inserts completely at random

- Each TA site equally likely
- Genes are either completely essential or completely non-essential

We first form a model for the process: that the transposon inserts completely at random, with each TA site equally likely. And that genes are completely essential or completely non-essential.

Model

N genes x_i = no. TA sites in gene i

n mutants y_i = no. mutants with insertion in gene i

$$\theta_i = \begin{cases} 1 \\ 0 \end{cases} \text{ if gene } i \text{ is } \begin{matrix} \text{non-essential} \\ \text{essential} \end{matrix}$$

Model: $\boldsymbol{y} \sim \text{multinomial}(n, \boldsymbol{p})$ where $p_i = x_i\theta_i / \sum_j x_j\theta_j$

Goal: Estimate $\theta_+ = \sum_i \theta_i$ or $1 - \theta_+/N$

Let x_i be the number of insertion sites in gene i (of N) and y_i be the number of mutants with insertion in gene i (of n). That's our data.

Let $\theta_i = 1$ if the gene is non-essential and 0 if essential. Those are our parameters. Then $\boldsymbol{y}$ is multinomial(n,$\boldsymbol{p}$) where p_i is the proportion of TA sites in gene i, among TA sites in non-essential genes.

We want to estimate $\theta_+ = \sum_i \theta_i$

The likelihood

$$L(\boldsymbol{\theta} \mid \boldsymbol{y}) = \binom{n}{y} \prod_i (x_i\theta_i)^{y_i} / \sum_j (x_j\theta_j)^n$$

$$\propto \begin{cases} (\sum_i x_i\theta_i)^{-n} & \text{if } \theta_i = 1 \text{ whenever } y_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

Notes:

- Depends only on which $y_i > 0$ and not on the specific values
- The MLE is $\hat{\theta}_i = 1\{y_i > 0\}$

The likelihood ends up depending only on which $y_i > 0$ and not on the specific values.

Maximum likelihood works particularly badly in this sort of situation. The MLE of θ_+ is just the minimum possible value that is consistent with the data. (That the genes with observed mutants are non-essential and all other genes are essential.)

It took a while to understand the mutagenesis data, but once we did we came almost directly to this model. It is similar to species counting sorts of problems. Gather and classify a random sample of organisms from some environment and try to infer something about the diversity of the population.

The prior

$\theta_+ \sim$ uniform on { 0, 1, ..., N }

$\boldsymbol{\theta} \mid \theta_+ \sim$ uniform over all sequences of 0's and 1's with θ_+ 1's

Notes:

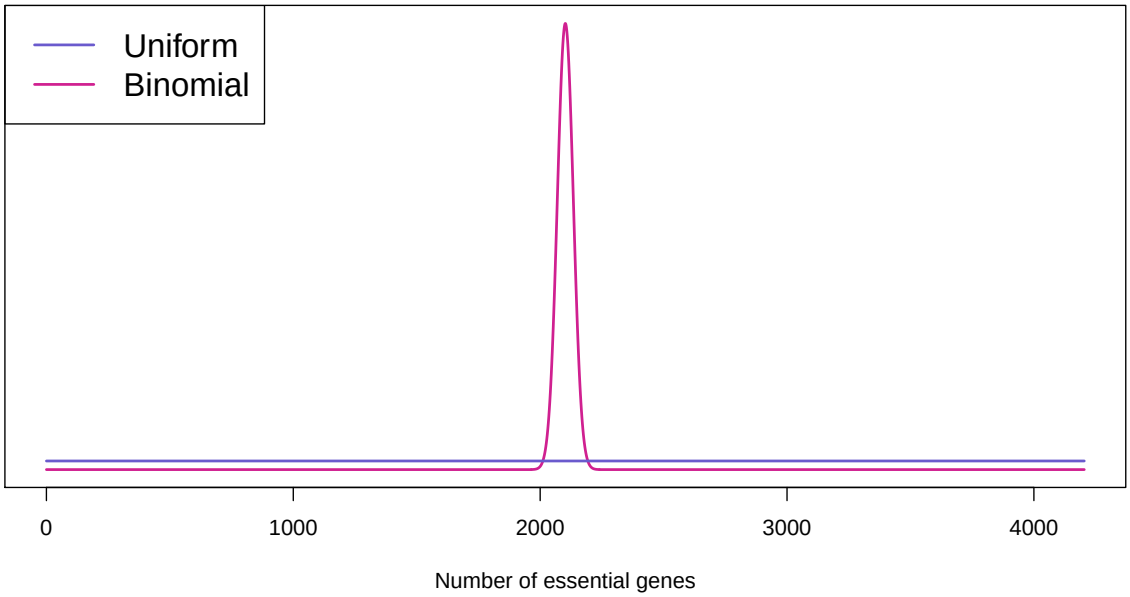
- We are assuming that $\Pr(\theta_i = 1) = 1/2$
- This is quite different from taking θ_i iid Bernoulli(1/2)
- We are assuming that θ_i is independent of x_i and the length of the gene
- We could make use of information about the essential status of particular genes (e.g. known viable knock-outs)

There seemed no real hope but to bite the bullet and do Bayes. That is, take our parameters to be random, make some assumption about their prior distribution, and then calculate the posterior distribution given the observed data.

You might naturally say something like $\theta_i \sim$ iid coin tosses, but this leads to a very concentrated prior on the number of essential genes.

So I instead thought: put a flat prior on the number of essential genes, and then given the number, take the identity of the particular genes that are essential to be random.

This means the θ_i are dependent, but seems to correspond reasonably well to what we know. But note that were assuming that each gene is equally likely to be essential.



Just to hammer this home: here's a comparison on the marginal distribution of θ_+ under the θ_i iid prior vs the prior we used.

A Gibbs sampler

Goal: Estimate $\Pr(\boldsymbol{\theta} \mid \boldsymbol{y})$

Gibbs sampler:

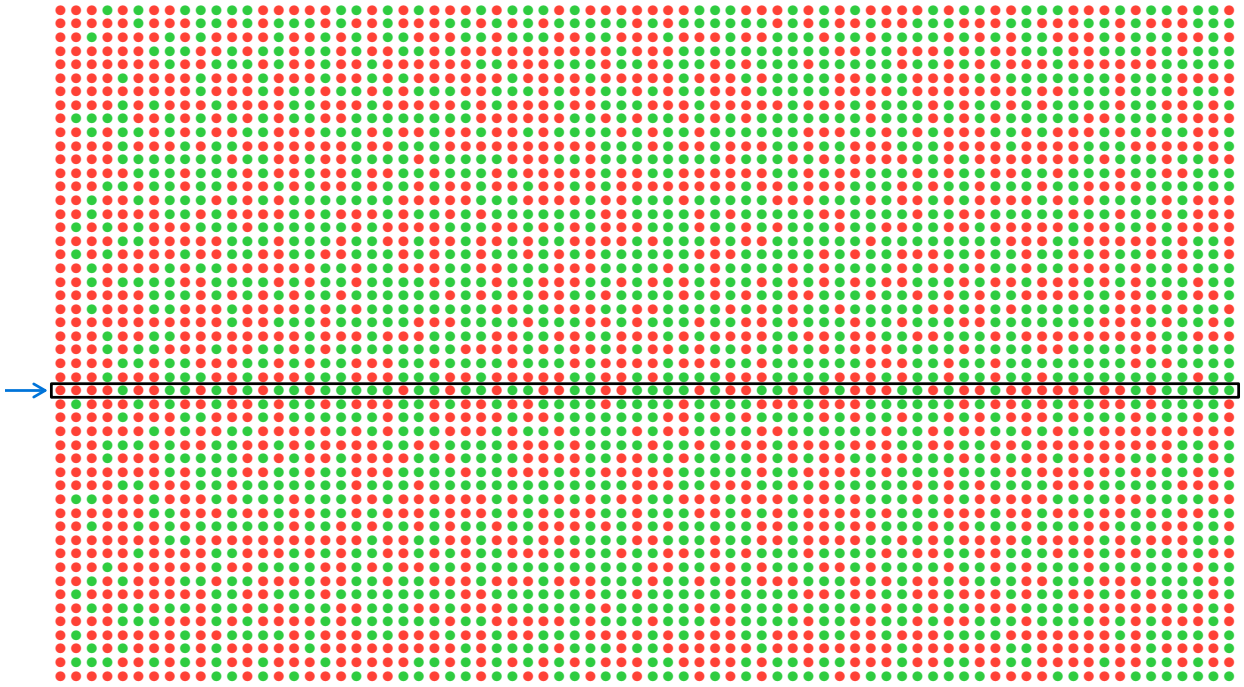
- Begin with some initial assignment $\boldsymbol{\theta}^{(0)}$
- For iteration s, consider each gene one at a time
  - Let $\boldsymbol{\theta}_{-i}^{(s)} = (\theta_1^{(s+1)}, ..., \theta_{i-1}^{(s+1)}, \theta_{i+1}^{(s)}, ..., \theta_N^{(s)})$
  - Calculate $\Pr(\theta_i = 1 \mid \boldsymbol{\theta}_{-i}^{(s)}, \boldsymbol{y})$
  - Assign $\theta_i^{(s)} = 1$ at random with that probability
- Repeat many times

This is an example of Markov chain Monte Carlo (MCMC).

Our goal is to estimate the posterior distribution $\Pr(\boldsymbol{\theta} \mid \boldsymbol{y})$. The Gibbs simpler is a natural way to do so, it's an example of Markov chain Monte Carlo.

You form a Markov chain whose limiting distribution is the target distribution.

MCMC in action



Here's an illustration of the Gibbs sampler for a portion of the data. Each dot is a parameter (for a gene): red is essential and green is non-essential. We start by assuming that all of the genes without viable mutants are essential (red). Each column is one iteration of the Gibbs sampler, and each row is one gene.

We can estimate the probability that a gene is essential by looking at the proportion of red dots in its row.

The conditional probabilities

If $y_i > 0$, then $\Pr(\theta_i = 1 \mid \boldsymbol{y}, \boldsymbol{\theta}_{-i}^{(s)}) = 1$

If $y_i = 0$,

$$\begin{aligned} \text{Let A} &= \textstyle\sum_{j<i} \theta_j^{(s+1)} + \sum_{j>i} \theta_j^{(s)} \\ \text{B} &= \textstyle\sum_{j<i} x_j\, \theta_j^{(s+1)} + \sum_{j>i} x_j\, \theta_j^{(s)} \end{aligned}$$

$$\begin{aligned} \text{Then } \Pr(\boldsymbol{\theta}_{-i}^{(s)}, \theta_i = k) &\propto \binom{n}{A+k} / n \\ \Pr(\boldsymbol{y} \mid \boldsymbol{\theta}_{-i}^{(s)}, \theta_i = k) &\propto (B + k\, x_i)^{-n} \end{aligned}$$

$$\begin{aligned} \text{And so } \Pr(\theta_i = 1 \mid \boldsymbol{y}, \boldsymbol{\theta}_{-i}^{(s)}) &= \ldots \\ &= \frac{(1 + x_i/B)^{-n}}{(1 + x_i/B)^{-n} + (n - A)/(A + 1)} \end{aligned}$$

Here are the details on the conditional probabilities. It's not really worth digging into them, really.

The key thing is to take A to be the number of non-essential genes (ignoring gene i) and B be the number of insertion sites in non-essential genes (again ignoring gene k). A straightforward application of Bayes's rule plus a bit of algebra leads to a reasonably simple equation.

Estimators

The Gibbs sampler produces $\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(S)}$

We discard the first 200 or so samples ("burn-in").

Estimated number of non-essential genes: $\mathsf{E}(\theta_+ \mid \mathbf{y})$

$$\theta_+^{(s)} = \sum_i \theta_i^{(s)} \qquad \longrightarrow \qquad \hat{\theta}_+ = \frac{1}{S-200} \sum_{s=201}^{S} \theta_+^{(s)}$$

Probability that gene i is non-essential: $\mathsf{E}(\theta_i \mid \mathbf{y}) = \mathsf{Pr}(\theta_i = 1 \mid \mathbf{y})$

$$\hat{\theta}_i = \frac{1}{S-200} \sum_{s=201}^{S} \theta_i^{(s)}$$

or Rao-Blackwellize:

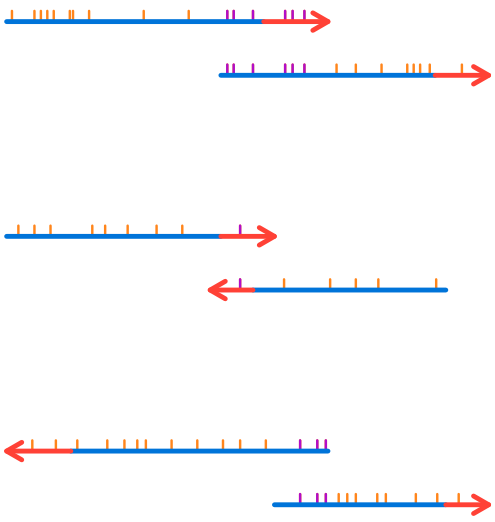
$$\hat{\theta}_i^{\star} = \frac{1}{S-200} \sum_{s=201}^{S} \mathsf{Pr}(\theta_i = 1 \mid \mathbf{y}, \boldsymbol{\theta}_{-i}^{(s)})$$

How do you use the MCMC results? Typically you'll discard the first bunch of samples and then take like the average of the rest. Rather than average the observed values, you can average the probabilities that you're sampling from. That ends up giving more precise estimates.

A further complication

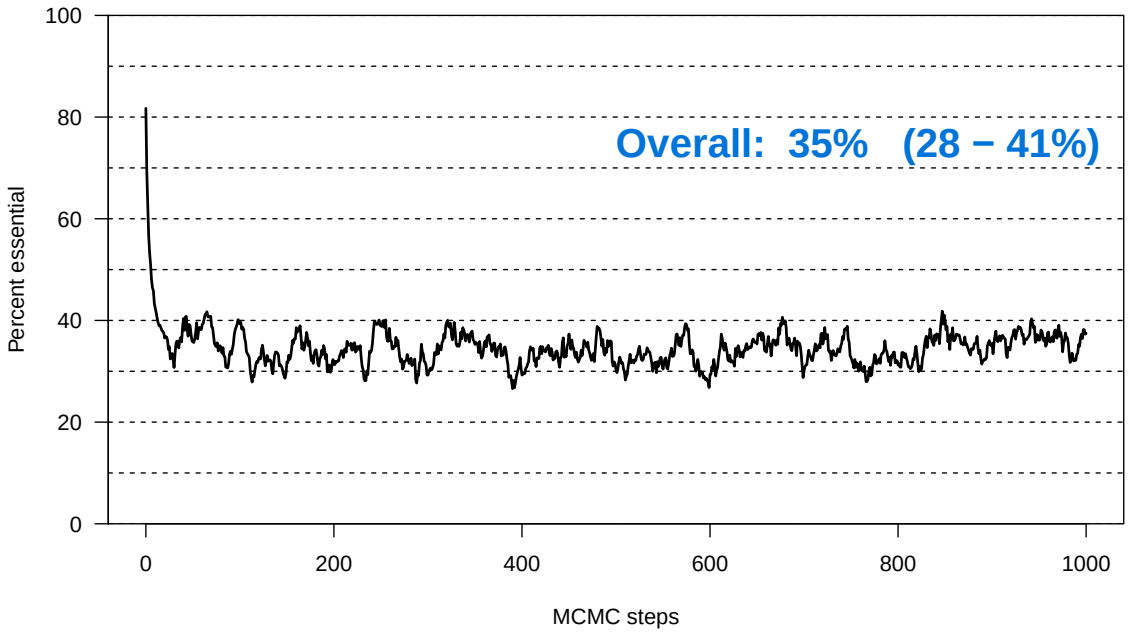
Many genes overlap

- Of 4250 genes, 1005 pairs overlap (mostly by exactly 4 bp).
- The overlapping regions contain 547 insertion sites.
- Omit TA sites in overlapping regions, unless in the proximal portion of *both* genes.
- The algebra gets a bit more complicated.



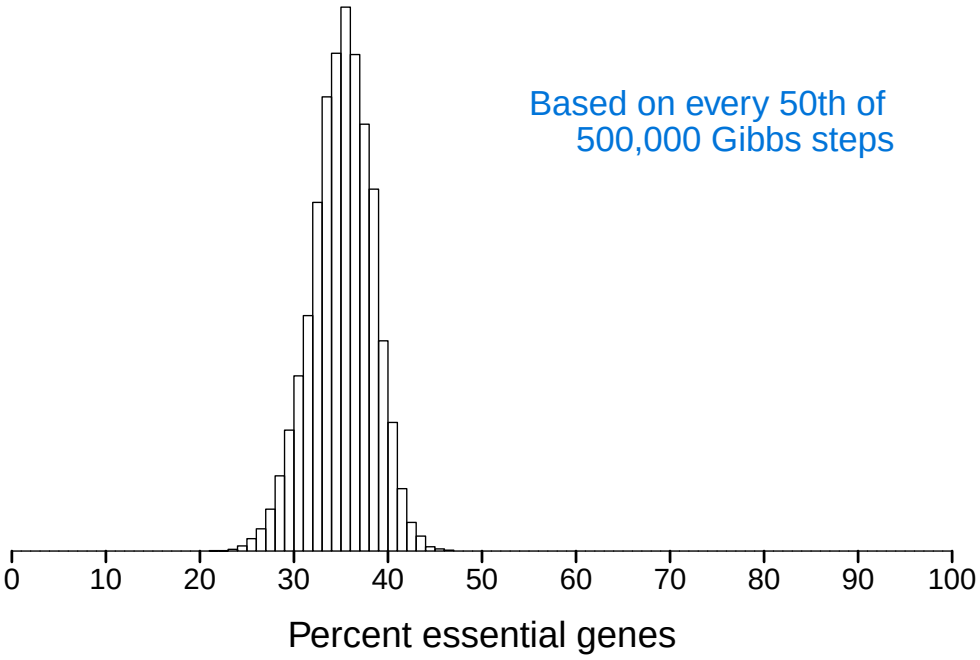
A further complication that I hadn't mentioned above: some of the genes overlap. And those regions of overlap include transposon insertion sites. This makes things a bit more complicated, but basically we consider sites in the overlap regions only if they are in the early bit of both genes. And insertion in such a site would indicate that both of the genes are non-essential.

Percent essential genes in M. tb.



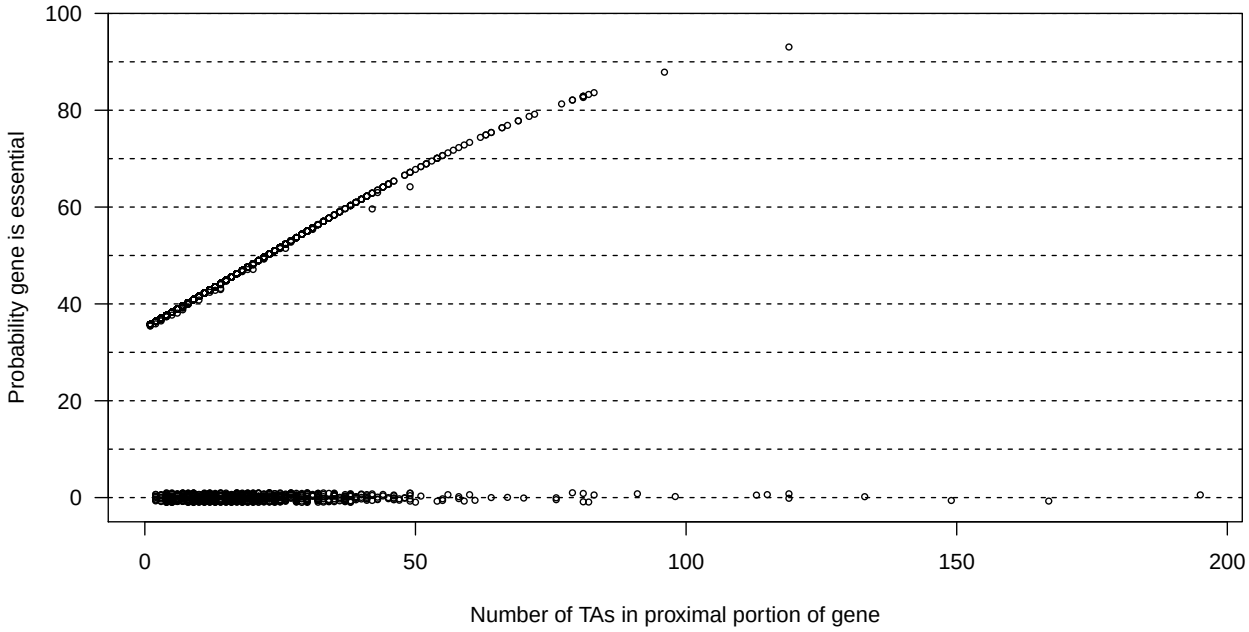
Here are the results of the first 1000 samples, showing the percent essential genes in each iteration. We start assuming that all genes without a mutation are essential, but it shoots immediately towards $\sim$35%, where it then stays.

Percent essential genes in M. tb.



Here are the results for every 50th of 500k steps. This is the estimated posterior distribution of the number of essential genes. We can use it to get an estimate and a "credible" interval.

Probability each gene is essential



We also get, for each gene, the estimated probability that it is essential. The genes showing a viable mutant are all down at 0 (jittered vertically to better see the individual points).

For genes without a viable insertion, the chance they are essential depends on the number of insertion sites. There is basically a functional relationship here; the points below the main curve are genes that share insertion sites with another gene. Shared insertion sites that haven't been hit are less informative than insertion sites in a single gene.

Yet another complication

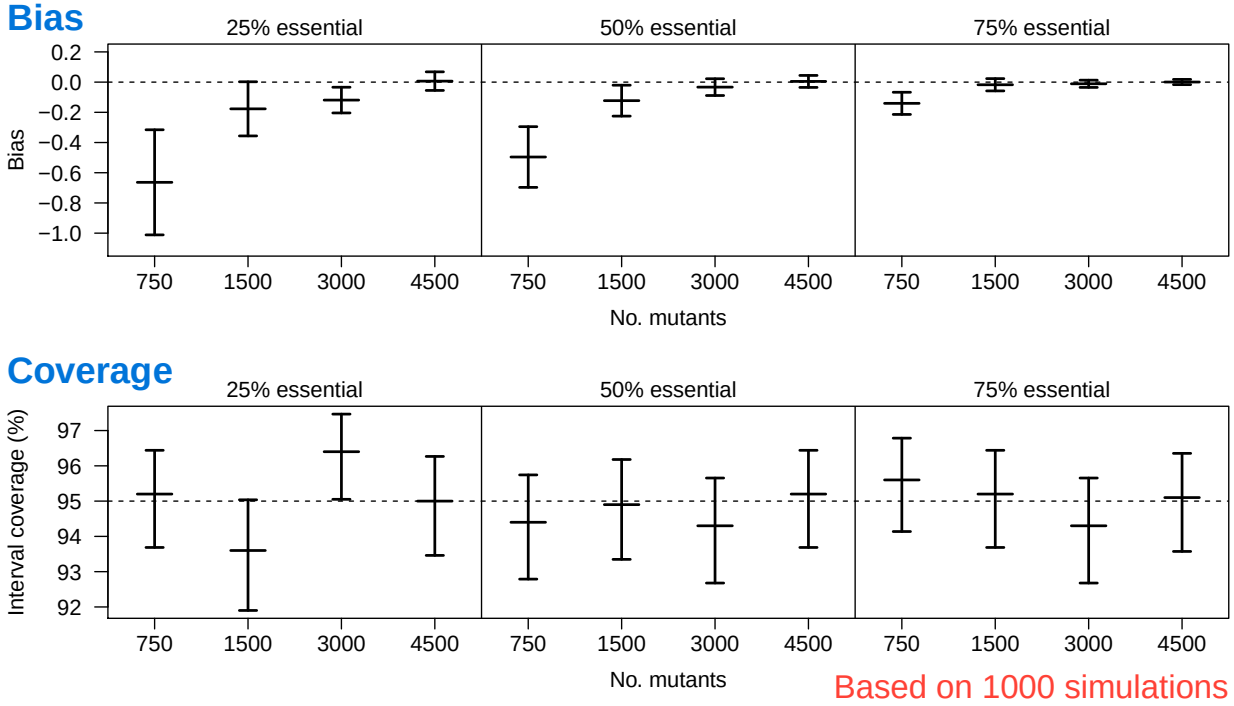
Operon: A group of adjacent genes that are transcribed together as a single unit.

- Insertion at a TA site could disrupt all downstream genes
- If a gene is essential, insertion in any upstream gene would be non-viable
- Re-define the meaning of "essential gene".
- If operons were known, one could get an improved estimate of the proportion of essential genes.
- If one ignores the presence of operons, estimates should still be unbiased.

One other complication that I hadn't mentioned: genes in M. tuberculosis exist in "operons" where a bunch of genes are transcribed together and then translated into protein. Because of this, transposon insertion could disrupt all downstream genes.

If we knew all of the operons, we could incorporate this into our model. The problem is that operons aren't entirely known. There's no hope but to redefine the meaning of "essential" to include being upstream of a truly essential gene.

Frequentist properties



25

How do we know if our approach works? Well, we can simulate.

And being a frequentist at heart, I'm interested in assessing the usual properties of these estimates. Consider different values for the proportion of essential genes, and different sizes of mutant libraries. How do our estimates behave, in terms of bias and interval coverage?

Turns out, they look just fine.

Summary

- Bayesian method, using MCMC, to estimate the proportion of essential genes in a genome with data from random transposon mutagenesis.
- Crucial assumptions:
  - Randomness of transposon insertion.
  - Essentiality is an all-or-none quality.
  - No relationship between essentiality and no. insertion sites.
  - The 80% rule.
- For *M. tuberculosis*, with data on 1400 mutants:
  - 28 – 41% of genes are essential
  - 20 genes which have $\geq$ 64 TA sites and for which no mutant has been observed, have $>$ 75% chance of being essential.

I've described a Bayesian method for making sense of transposon mutagenesis data. I came to it by first forming a natural model for the data and then seeing that the usual sorts of approaches would not work well.

The two key things here were the choice of prior and then forming the Gibbs sampler (and showing that it worked well).

There are some critical assumptions, mostly untestable. And there were some complexities that lead to a bit of ad-hoc-ery, but for the most part the results seem solid.

References

- Lamichhane et al. (2003) Proc Natl Acad Sci USA 100:7213-7218 doi:10.1073/pnas.1231432100
- Blades and Broman (2002) Tech Report MS02-20 bit.ly/ms0220
- R/negenes package cran.r-project.org/package=negenes
- *Reproduction of the results*: Broman (2020) github.com/kbroman/Paper_ReScience2020

Here is the main paper plus a tech report about the work.

The R package is a bit of a mess, but it does still work.

The last paper was a recent effort to reproduce the original results, with some interesting lessons.