

Estimating the Number of Essential Genes using Occugene

Oliver Will

October 27, 2020

This vignette contains code from a chapter of the forthcoming book, A Osterman and S Gerdes. *Gene Essentiality: Protocols and Bioinformatics*. Humana Press. This package has similar functionality as the R package `negenes` written by Karl Broman.

Biologists have built large random mutagenesis libraries for prokaryotes. For a description of the biology, see MA Jacobs (et al). (2003) Comprehensive transposon mutant library of *Pseudomonas aeruginosa*. *Proc Natl Acad Sci USA*. **100**:14339–14344. The `occugene` package provides statistical tools to help build random libraries.

We model the number of insertions per clone as a Multinomial(n, p_1, \dots, p_k) random vector. The number of knockouts in the library follows the occupancy distribution of the multinomial random variable. We compute the expected number of genes hit if there were no essential genes.

```
> library("occugene")
> n <- 60
> p <- c(seq(10, 1, -1), seq(10, 1, -1), 18)/124
> p <- p/sum(p)
> eMult(n,p)
```

```
[1] 17.74773
```

```
> varMult(n,p)
```

```
[1] 1.744004
```

We approximate the moments of the occupancy distribution using Monte Carlo integration.

```
> eMult(n,p,iter=100,seed=4)
```

```
[1] 17.64
```

```
> varMult(n,p,iter=100,seed=4)
```

```
[1] 2.050909
```

We load an example hit table and experimental results to analyze. The format of the hit table is different than what is used in `negenes` because we wish to track the order of insert locations.

```
> data(sampleAnnotation)
> data(sampleInsertions)
> print(sampleAnnotation)
```

| | idNum | first | last | orientation |
|----|-------|-------|------|-------------|
| 1 | 1 | 2 | 11 | 0 |
| 2 | 2 | 13 | 21 | 0 |
| 3 | 3 | 23 | 30 | 0 |
| 4 | 4 | 32 | 38 | 0 |
| 5 | 5 | 40 | 45 | 0 |
| 6 | 6 | 44 | 48 | 0 |
| 7 | 7 | 50 | 53 | 0 |
| 8 | 8 | 55 | 57 | 0 |
| 9 | 9 | 59 | 60 | 0 |
| 10 | 10 | 62 | 62 | 0 |
| 11 | 11 | 64 | 73 | 0 |
| 12 | 12 | 75 | 83 | 0 |
| 13 | 13 | 85 | 92 | 0 |
| 14 | 14 | 94 | 100 | 0 |
| 15 | 15 | 102 | 107 | 0 |
| 16 | 16 | 106 | 110 | 0 |
| 17 | 17 | 112 | 115 | 0 |
| 18 | 18 | 117 | 119 | 0 |
| 19 | 19 | 121 | 122 | 0 |
| 20 | 20 | 124 | 124 | 0 |

```
> print(sampleInsertions)
```

| | position |
|----|----------|
| 1 | 69 |
| 2 | 2 |
| 3 | 36 |
| 4 | 34 |
| 5 | 99 |
| 6 | 33 |
| 7 | 91 |
| 8 | 113 |
| 9 | 120 |
| 10 | 8 |
| 11 | 93 |
| 12 | 35 |

| | |
|----|-----|
| 13 | 11 |
| 14 | 120 |
| 15 | 52 |
| 16 | 55 |
| 17 | 122 |
| 18 | 69 |
| 19 | 121 |
| 20 | 94 |
| 21 | 90 |
| 22 | 124 |
| 23 | 62 |
| 24 | 61 |
| 25 | 84 |
| 26 | 100 |
| 27 | 58 |
| 28 | 101 |
| 29 | 63 |
| 30 | 64 |
| 31 | 68 |
| 32 | 31 |
| 33 | 111 |
| 34 | 84 |
| 35 | 58 |
| 36 | 122 |
| 37 | 56 |
| 38 | 72 |
| 39 | 49 |
| 40 | 2 |
| 41 | 116 |
| 42 | 31 |
| 43 | 67 |
| 44 | 18 |
| 45 | 113 |
| 46 | 9 |
| 47 | 113 |
| 48 | 112 |
| 49 | 91 |
| 50 | 67 |
| 51 | 49 |
| 52 | 93 |
| 53 | 112 |
| 54 | 98 |
| 55 | 99 |
| 56 | 52 |
| 57 | 17 |
| 58 | 17 |

```
59      112
60      92
```

```
> a.data <- sampleAnnotation
> experiment <- sampleInsertions
```

We estimate the number of genes that will be knocked out in the next 10 clones using the Efron and Thisted estimator.

```
> orf <- cbind(a.data$first,a.data$last)
> clone <- experiment$position
> etDelta(10,orf,clone)
```

```
$expected
[1] 0.1190665
```

```
$variance
[1] 0.02936508
```

We use the Will and Jacobs' bootstrap to estimate the number of knockouts made in the next 10 clones.

```
> orf <- cbind(a.data$first,a.data$last)
> clone <- experiment$position
> fFit(orf,clone,FALSE)
```

```
Nonlinear regression model
model: noOrfs ~ b0 - b1 * exp(-b2 * x)
data: cumul
      b0      b1      b2
12.34393 12.05475 0.06668
residual sum-of-squares: 15.62
```

```
Number of iterations to convergence: 6
Achieved convergence tolerance: 7.669e-07
```

```
> unbiasedDelta0(10,orf,clone,iter=10,seed=4,alpha=0.05,TR=F)
```

```
$delta0
[1] 0.2959099
```

```
$CI
[1] -0.1730529 0.5010068
```

We estimate the number of essential genes using the Will and Jacobs' bootstrap.

```
> unbiasedB0(orf,clone,iter=10,seed=4,alpha=0.05,TR=F)
```

```
$b0  
[1] 14.42681
```

```
$CI  
[1] 10.87780 16.83998
```

Finally, we convert `occugene`'s data format into the format for `negenes`.

```
> newOrf <- occup2Negenes(orf,clone)  
> print(newOrf)
```

| | n.sites | n.sites2 | counts | counts2 |
|----|---------|----------|--------|---------|
| 1 | 10 | 0 | 5 | 0 |
| 2 | 9 | 0 | 3 | 0 |
| 3 | 8 | 0 | 0 | 0 |
| 4 | 7 | 0 | 4 | 0 |
| 5 | 4 | 2 | 0 | 0 |
| 6 | 3 | 0 | 0 | 0 |
| 7 | 4 | 0 | 2 | 0 |
| 8 | 3 | 0 | 2 | 0 |
| 9 | 2 | 0 | 0 | 0 |
| 10 | 1 | 0 | 1 | 0 |
| 11 | 10 | 0 | 7 | 0 |
| 12 | 9 | 0 | 0 | 0 |
| 13 | 8 | 0 | 4 | 0 |
| 14 | 7 | 0 | 5 | 0 |
| 15 | 4 | 2 | 0 | 0 |
| 16 | 3 | 0 | 0 | 0 |
| 17 | 4 | 0 | 6 | 0 |
| 18 | 3 | 0 | 0 | 0 |
| 19 | 2 | 0 | 3 | 0 |
| 20 | 1 | 0 | 1 | 0 |

We outlined the basic features of the `occugene` package in this Sweave document.