Approximating the Sum of Independent Non-Identical Binomial Random Variables

by Boxiang Liu and Thomas Quertermous

Abstract The distribution of the sum of independent non-identical binomial random variables is frequently encountered in areas such as genomics, healthcare, and operations research. Analytical solutions for the density and distribution are usually cumbersome to find and difficult to compute. Several methods have been developed to approximate the distribution, among which is the saddlepoint approximation. However, implementation of the saddlepoint approximation is non-trivial. In this paper, we implement the saddlepoint approximation in the **sinib** package and provide two examples to illustrate its usage. One example uses simulated data while the other uses real-world healthcare data. The **sinib** package addresses the gap between the theory and the implementation of approximating the sum of independent non-identical binomials.

Introduction

Convolution of independent non-identical binomial random variables appears in a variety of applications, such as analysis of variant-region overlap in genomics (Schmidt et al., 2015), calculation of bundle compliance statistics in healthcare organizations (Benneyan and Taşeli, 2010), and reliability analysis in operations research (Kotz and Johnson, 1984).

Computing the exact distribution of the sum of non-identical independent binomial random variables requires enumeration of all possible combinations of binomial outcomes that satisfy the totality constraint. However, analytical solutions are often difficult to find for sums of greater than two binomial random variables. Several studies have proposed approximate solutions (Johnson et al., 2005; Jolayemi, 1992). In particular, Eisinga et al. (2013) examined the saddlepoint approximation, and compared them to exact solutions. They note that in practice, these approximations are often as good as the exact solution and can be implemented in most statistical software.

Despite the theoretical development of aforementioned approximate solutions, a software implementation in R is still lacking. The **stats** package includes functions for frequently used distribution such as `dbinom` and `dnorm`, and less frequently used distributions such as `pbirthday`, but it does not contain functions for the distribution of the sum of independent non-identical binomials. The **EQL** package provides a `saddlepoint` function to approximate the mean of i.i.d. random variables, but does not apply to the case where the random variables are not identical. In this paper, we implement a saddlepoint approximation in **sinib** (sum of independent non-identical binomial random variables). The package provides the standard suite of distributional functions for the distribution (`psinib`), density (`dsinib`), quantile (`qsinib`), and random deviates (`rsinib`). The package is accompanied by a detailed documentation, and can be easily integrated into existing applications.

The remainder of this paper is organized as follows. We begin by providing an overview of the distribution of the sum of independent non-identical binomial random variables. Next, we give an overview of the saddlepoint approximation. The following section describes the design and implementation of the saddlepoint approximation in the **sinib** package. We provide two examples and assess the accuracy of saddlepoint approximation in these situations. The final section concludes and discusses future direction.

Overview of the distribution

Suppose $X_1, \ldots, X_m$ are independent non-identical binomial random variables such that $S_m = \sum_{i=1}^{m} X_i$. We are interested in finding the distribution of S_m.

$$P(S_m = s) = P(X_1 + X_2 + \cdots + X_m = s) \tag{1}$$

In the special case of $m = 2$, the probability simplifies to

$$P(S_2 = s) = P(X_1 + X_2 = s) = \sum_{i=0}^{s} P(X_1 = i)P(X_2 = s - i) \tag{2}$$

Computation of the exact distribution often involves enumerating all possible combinations of each variable that sum to a given value, which becomes infeasible when n is large. A fast recursion method to compute the exact distribution has been proposed (Butler and Stephens, 2016; Arthur Woodward and Palmer, 1997). The algorithm is as follows:

1. Compute the exact distribution of each X_i.

2. Calculate the distribution of $S_2 = X_1 + X_2$ using Equation 2 and cache the result.

3. Calculate $S_r = S_{r-1} + X_i$ for $r = 3, 4, \ldots, m$.

Although the recursion speeds up the calculation, studies have shown that the result may be numerically unstable due to round-off error in computing $P(S_r = 0)$ if r is large (Eisinga et al., 2013; Yili). Therefore, approximation methods are still widely used in literature.

Saddlepoint approximation

The saddlepoint approximation, first proposed by Daniels (1954) and later extended by Lugannani and Rice (1980), provides highly accurate approximations for the probability and density of many distributions. In brief, let $M(u)$ be the moment generating function, and $K(u) = \log(M(u))$ be the cumulant generating function. The saddlepoint approximation to the PDF of the distribution is given as:

$$\hat{P}_1(S = s) = \frac{\exp(K(\hat{u}) - \hat{u}s)}{\sqrt{2\pi K''(\hat{u})}} \tag{3}$$

where $\hat{u}$ is the unique value that satisfies $K'(\hat{u}) = s$.

Eisinga et al. (2013) applied the saddlepoint approximation to the sum of independent non-identical binomial random variables. Suppose that $X_i \sim Binomial(n_i, p_i)$ for $i = 1, 2, \ldots, m$. The cumulant generating function of $S_m = \sum_{i=1}^{m} X_i$ is:

$$K(u) = \sum_{i=1}^{m} n_i \ln(1 - p_i + p_i \exp(u)) \tag{4}$$

The first- and second-order derivatives of $K(u)$ are:

$$K'(u) = \sum_{i=1}^{m} n_i q_i \tag{5}$$

$$K''(u) = \sum_{i=1}^{m} n_i q_i (1 - q_i) \tag{6}$$

where $q_i = \frac{p_i exp(u)}{(1 - p_i + p_i \exp(u))}$.

The saddlepoint of $\hat{u}$ can be obtained by solving $K'(\hat{u}) = s$. A unique root can always be found because $K(u)$ is strictly convex and therefore $K'(u)$ is monotonically increasing on the real line.

The above shows the first-order approximation of the distribution. The approximation can be improved by adding a second-order correction term (Daniels, 1987; Akahira and Takahashi, 2001):

$$\hat{P}_2(S = s) = \hat{P}_1(S = s)\left\{1 + \frac{K''''(\hat{u})}{8[K''(\hat{u})]^2} - \frac{5[K'''(\hat{u})]^2}{24[K''(\hat{u})]^3}\right\} \tag{7}$$

where

$$K'''(\hat{u}) = \sum_{i=1}^{m} n_i q_i (1 - q_i)(1 - 2q_i)$$

and

$$K''''(\hat{u}) = \sum_{i=1}^{m} n_i q_i (1 - q_i)[1 - 6q_i(1 - q_i)]$$

Although the saddlepoint equation cannot be solved at boundaries $s = 0$ and $s = \sum_{i=1}^{m} n_i$, their exact probabilities can be computed easily:

$$P(S = 0) = \prod_{i=1}^{m} (1 - p_i)^{n_i} \tag{8}$$

$$P(S = \sum_{i=1}^{m} n_i) = \prod_{i=1}^{m} p_i^{n_i} \tag{9}$$

Incorporation of boundary solutions into the approximation gives:

$$\bar{P}(S = s) = \begin{cases} P(S = 0), & s = 0 \\ [1 - P(S = 0) - P(S = \sum_{i=1}^{m} n_i)] \frac{\hat{P}_2(S=s)}{\sum_{i=1}^{\sum_{i=1}^{m} n_i - 1} \hat{P}_2(S=i)}, & 0 < s < \sum_{i=1}^{m} n_i \\ P(S = \sum_{i=1}^{m} n_i), & s = \sum_{i=1}^{m} n_i \end{cases} \quad (10)$$

We have implemented Equation 10 as the final approxmation of the probability density function. For the cumulative density, Daniels (1987) gave the following approximator:

$$\hat{P}_3(S \geq s) = \begin{cases} 1 - \Phi(\hat{w}) - \phi(\hat{w})(\frac{1}{\hat{w}} - \frac{1}{\hat{u}_1}), & \text{if } s \neq E(S) \text{ and } \hat{u} \neq 0 \\ \frac{1}{2} - \frac{1}{\sqrt{2\pi}} \Big[\frac{K'''(0)}{6K''(0)^{3/2}} - \frac{1}{2\sqrt{K''(0)}} \Big], & \text{otherwise} \end{cases} \quad (11)$$

where $\hat{w} = \text{sign}(\hat{u})[2\hat{u}K'(\hat{u}) - 2K(\hat{u})]^{1/2}$ and $\hat{u}_1 = [1 - \exp(-\hat{u})][K''(\hat{u})]^{1/2}$. The letters Φ and ϕ denotes the probability and density of the standard normal distribution.

The accuracy can be improved by adding a second-order continuity correction:

$$\hat{P}_4(S \geq s) = \hat{P}_3(S \geq s) - \phi(\hat{w}) \Big[\frac{1}{\hat{u}_2} \Big(\frac{\hat{\kappa}_4}{8} - \frac{5\hat{\kappa}_3^2}{24} \Big) - \frac{1}{\hat{u}_2^3} - \frac{\hat{\kappa}_3}{2\hat{u}_2^2} + \frac{1}{\hat{w}^3} \Big] \quad (12)$$

where $\hat{u}_2 = \hat{u}[K''(\hat{u})]^{1/2}$, $\hat{\kappa}_3 = K'''(\hat{u})[K''(\hat{u})]^{-3/2}$, and $\hat{\kappa}_4 = K''''(\hat{u})[K''(\hat{u})]^{-2}$.

We have implemented Equation 12 to approximate the cumulative distribution.

The sinib package

The package uses only base **R** and the **stats** package to minimize compatibility issues. The arguments for the functions in the **sinib** package are designed to have similar meaning to those in the **stats** package, thereby minimizing the learning required. To illustrate, we compare the arguments of the `*binom` and the `*sinib` functions.

From the help page of the binomial distribution, the arguments are as follows:

- x, q: vector of quantiles.
- p: vector of probabilities.
- n: number of observations.
- size: number of trials.
- prob: probability of success on each trial.
- log, log.p: logical; if TRUE, probabilities p are given as log(p).
- lower.tail: logical; if TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.

Since the distribution of sum of independent non-identical binomials is defined by a vector of trial and probability pairs (each pair for one constituent binomial), it was neccessary to redefine these arguments in the `*sinib` functions. Therefore, the following two arguments were redefined:

- size: integer vector of number of trials.
- prob: numeric vector of success probabilities.

All other arguments remain the same. It is worth noting that when size and prob arguments are given as vectors of length 1, the `*sinib` function reduces to `*binom` functions:

```
# Binomial:
dbinom(x = 1, size = 2, prob = 0.5)
[1] 0.5

# Sum of binomials:
library(sinib)
dsinib(x = 1, size = 2, prob = 0.5)
[1] 0.5
```

The next section shows a few examples to illustrate the usage of **sinib**.

Example usage of sinib

This section shows a few examples to illustrate the usage of **sinib**.

Sum of two binomials

We use two examples to illustrate the use of this package, starting from the simplest case of two binomial random variables with the same mean but different sizes: $X \sim Bin(n, p)$ and $Y \sim Bin(m, p)$. The distribution of $S = X + Y$ has an analytical solution, $S \sim Bin(m + n, p)$. We can therefore use different combinations of (m, n, p) to assess the accuracy of the saddlepoint approximation to the cumulative density function. We use $m, n = \{10, 100, 1000\}$ and $p = \{0.1, 0.5, 0.9\}$ to assess the approximation. The ranges of m and n are chosen to be large and the value of p are chosen to represent both boundaries.

```
library(foreach)
library(data.table)
library(cowplot)
library(sinib)

# Gaussian approximator:
p_norm_app = function(q,size,prob){
        mu = sum(size*prob)
        sigma = sqrt(sum(size*prob*(1-prob)))
        pnorm(q, mean = mu, sd = sigma)
}

# Comparison of CDF between truth and approximation:
data=foreach(m=c(10,100,1000),.combine='rbind')%do%{
        foreach(n=c(10,100,1000),.combine='rbind')%do%{
                foreach(p=c(0.1, 0.5, 0.9),.combine='rbind')%do%{
                        a=pbinom(q=0:(m+n),size=(m+n),prob = p)
                        b=psinib(q=0:(m+n),size=c(m,n),prob=c(p,p))
                        c=p_norm_app(q=0:(m+n),size=c(m,n),prob = c(p,p))
                        data.table(s=seq_along(a),truth=a,saddle=b,norm=c,m=m,n=n,p=p)
                }
        }
}
data[,m:=paste0('m = ',m)]
data[,p:=paste0('p = ',p)]
data = melt(data,measure.vars = c('saddle','norm'))

ggplot(data[n==10],aes(x=truth,y=value,color=p,linetype=variable))+
        geom_line()+
        facet_grid(p~m)+
        theme_bw()+
        scale_color_discrete(name='prob',guide = 'none')+
        xlab('Truth')+
        ylab('Approximation')+
        scale_linetype_discrete(name = '', breaks = c('saddle','norm'),
            labels = c('Saddlepoint','Gassian')) +
        theme(legend.position = 'top')
```

Figure 1 shows that the saddlepoint approximations are close to the ground truths across a range of parameters. For parsimony, this figure only shows the case of $n = 10$. In comparison with the saddlepoint method, the Gaussian method (dashed lines) provides a relatively poor approximation. We can further examine the accuracy by looking at the differences between the approximations and the ground truth.

```
data=foreach(m=c(100),.combine='rbind')%do%{
        foreach(n=c(100),.combine='rbind')%do%{
                foreach(p=c(0.1, 0.5, 0.9),.combine='rbind')%do%{
                        a=pbinom(q=0:(m+n),size=(m+n),prob = p)
                        b=psinib(q=0:(m+n),size=as.integer(c(m,n)),prob=c(p,p))
                        c=p_norm_app(q=0:(m+n),size=c(m,n),prob = c(p,p))
```
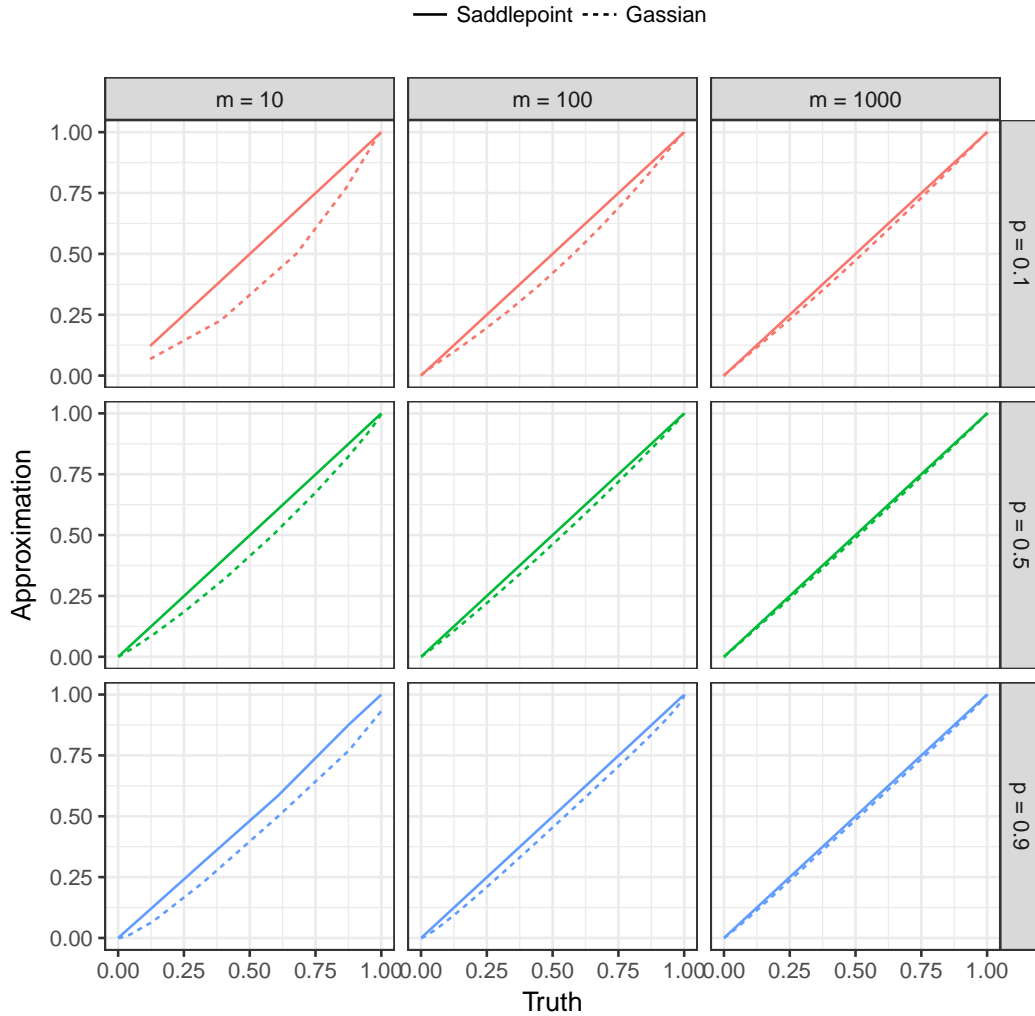
Figure 1: Comparison of CDF between truth and approximation for $n = 10$.

```
                    data.table(s=seq_along(a),truth=a,saddle=b,norm=c,m=m,n=n,p=p)
            }
        }
}
data = melt(data,measure.vars = c('saddle','norm'),value.name = 'approx',
        variable.name = 'Method')
data[,`Relative error` := (truth-approx)/truth]
data[,Error := (truth-approx)]
data[,p:=paste0('p = ',p)]
data = melt(data,measure.vars = c('Relative error','Error'),value.name = 'error',
        variable.name = 'type')

ggplot(data[Method == 'saddle'],aes(x=s,y=error,color=p))+
        geom_point(alpha=0.5)+theme_bw()+
        facet_wrap(type~p,scales='free')+
        xlab('Quantile')+
        ylab(expression(
            'Truth - Approximation'~~~~~~~~~~~~~~~frac(Truth - Approximation,Truth)))+
        scale_color_discrete(guide='none') +
        geom_vline(xintercept=200*0.5,color='green',linetype='longdash')+
        geom_vline(xintercept=200*0.1,color='red',linetype='longdash')+
        geom_vline(xintercept=200*0.9,color='blue',linetype='longdash')
```

Figure 2 shows the difference between the truth and the approximation for $m = n = 100$. The dashed lines indicate the mean of each random variable. The approximations perform well overall.
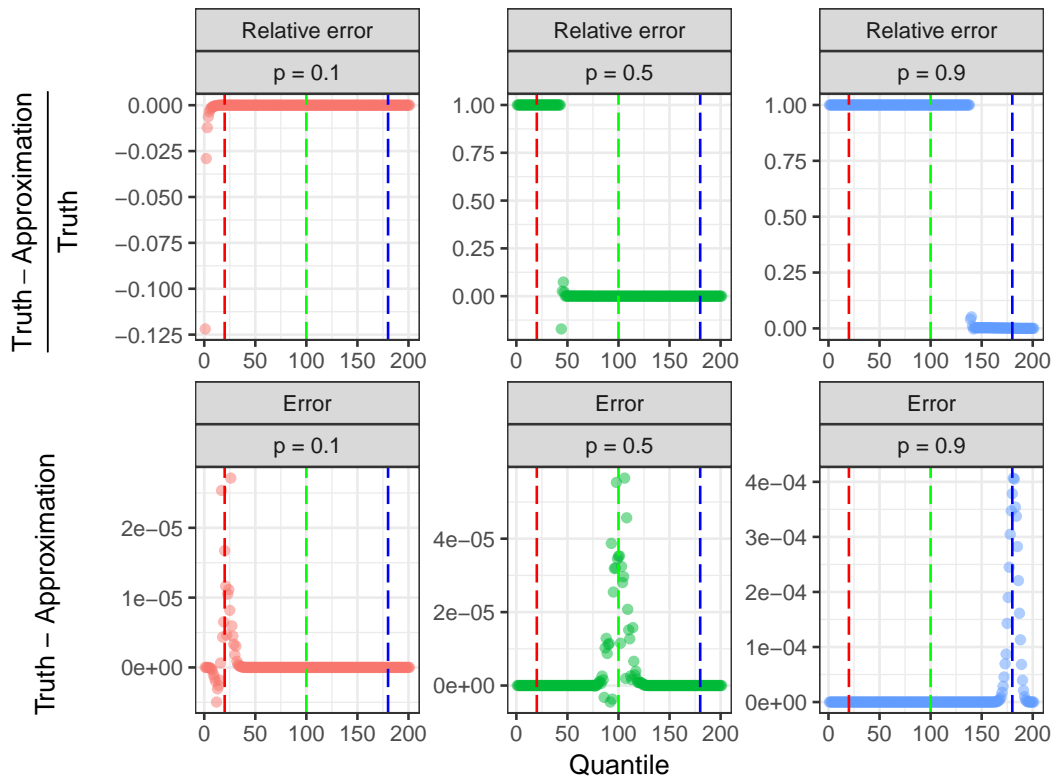
Figure 2: Difference in CDF between the ground truth and the approximation

The largest difference occurs around the mean, which is approximately 4e-4. It is worthwhile to mention that the errors are small for quantiles away from the mean because the the true probabilities are close to zero and one in the tails. To explore the tail behavior, we examine the relative error defined as $\frac{truth-approximation}{truth}$. The relative errors are large for quantiles between zero and the mean because the true probabilities in this interval are close to zero and the saddlepoint approximation returns zero. The relative errors are small for quantiles near and greater than the mean, indicating that the saddlepoint method provides a good approximation in this interval. As a baseline, we calculated the error and relative error derived from the Gaussian approximation (Figure 3). The largest absolute deviation approaches 0.06, two orders of magnitude greater than the deviation obtained from the saddlepoint approximation.

```
# Comparison of PDF between truth and approximation:
d_norm_app = function(x,size,prob){
        mu = sum(size*prob)
        sigma = sqrt(sum(size*prob*(1-prob)))
        dnorm(x, mean = mu, sd = sigma)
}

data=foreach(m=c(10,100,1000),.combine='rbind')%do%{
        foreach(n=c(10,100,1000),.combine='rbind')%do%{
                foreach(p=c(0.1, 0.5, 0.9),.combine='rbind')%do%{
                        a=dbinom(x=0:(m+n),size=(m+n),prob = p)
                        b=dsinib(x=0:(m+n),size=as.integer(c(m,n)),prob=c(p,p))
                        c=d_norm_app(x=0:(m+n),size=c(m,n),prob=c(p,p))
                        data.table(s=seq_along(a),truth=a,saddle=b,norm=c,m=m,n=n,p=p)
                }
        }
}

data[,m:=paste0('m = ',m)]
data[,p:=paste0('p = ',p)]
data = melt(data,measure.vars = c('saddle','norm'))
```
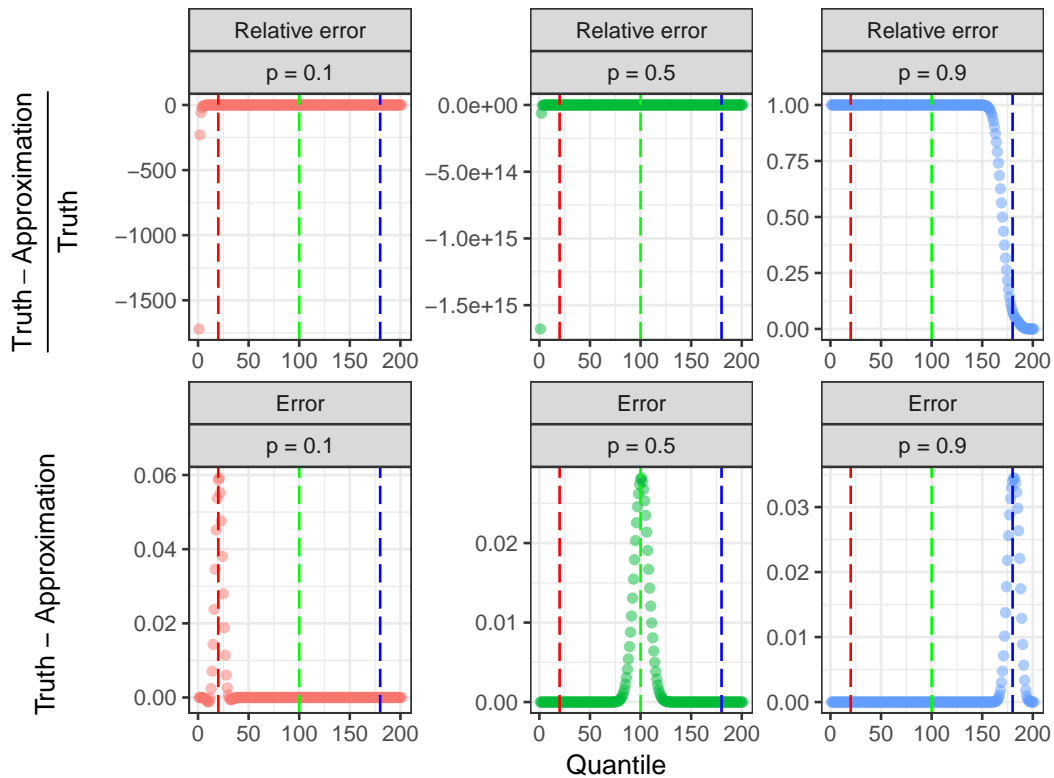
Figure 3: Difference in CDF between the ground truth and Gaussian approximation.

```
ggplot(data[n==10],aes(x=truth,y=value,color=p,linetype=variable))+
    geom_line()+
    facet_wrap(m~p,scales='free')+
    theme_bw()+
    scale_color_discrete(guide = 'none')+
    xlab('Truth')+
    ylab('Approximation') +
    scale_linetype_discrete(name = '', breaks = c('saddle','norm'),
            labels = c('Saddlepoint','Gassian')) +
    theme(legend.position = 'top')
```

We next examine the approximation for the probability density function. Figure 4 shows that the saddlepoint approximation is very close to the ground truth, whereas the Gaussian approximation is farther away. It is worthwhile to mention that the Gaussian method provides a good approximation for $p = 0.5$ because the distribution is symmetrical. Furthermore, we examine the difference between the truth and the approximations. One example for $m = n = 100$ is shown in Figure 5. As before, the saddlepoint approximation degrades around the mean, but the largest deviation is less than 4e-7. As a baseline, we calculated the difference between the true PDF and the Gaussian approximation in Figure 6. The largest deviation from the Gaussian approximation is 0.004, or four orders of magnitude greater than that from the saddlepoint approximation.

```
data=foreach(m=c(100),.combine='rbind')%do%{
        foreach(n=c(100),.combine='rbind')%do%{
                foreach(p=c(0.1, 0.5, 0.9),.combine='rbind')%do%{
                        a=dbinom(x=0:(m+n),size=(m+n),prob = p)
                        b=dsinib(x=0:(m+n),size=as.integer(c(m,n)),prob=c(p,p))
                        c=d_norm_app(x=0:(m+n),size=c(m,n),prob = c(p,p))
                        data.table(s=seq_along(a),truth=a,saddle=b,norm=c,m=m,n=n,p=p)
                }
        }
}
data = melt(data,measure.vars = c('saddle','norm'),value.name = 'approx',
        variable.name = 'Method')
data[,`Relative error` := (truth-approx)/truth]
```
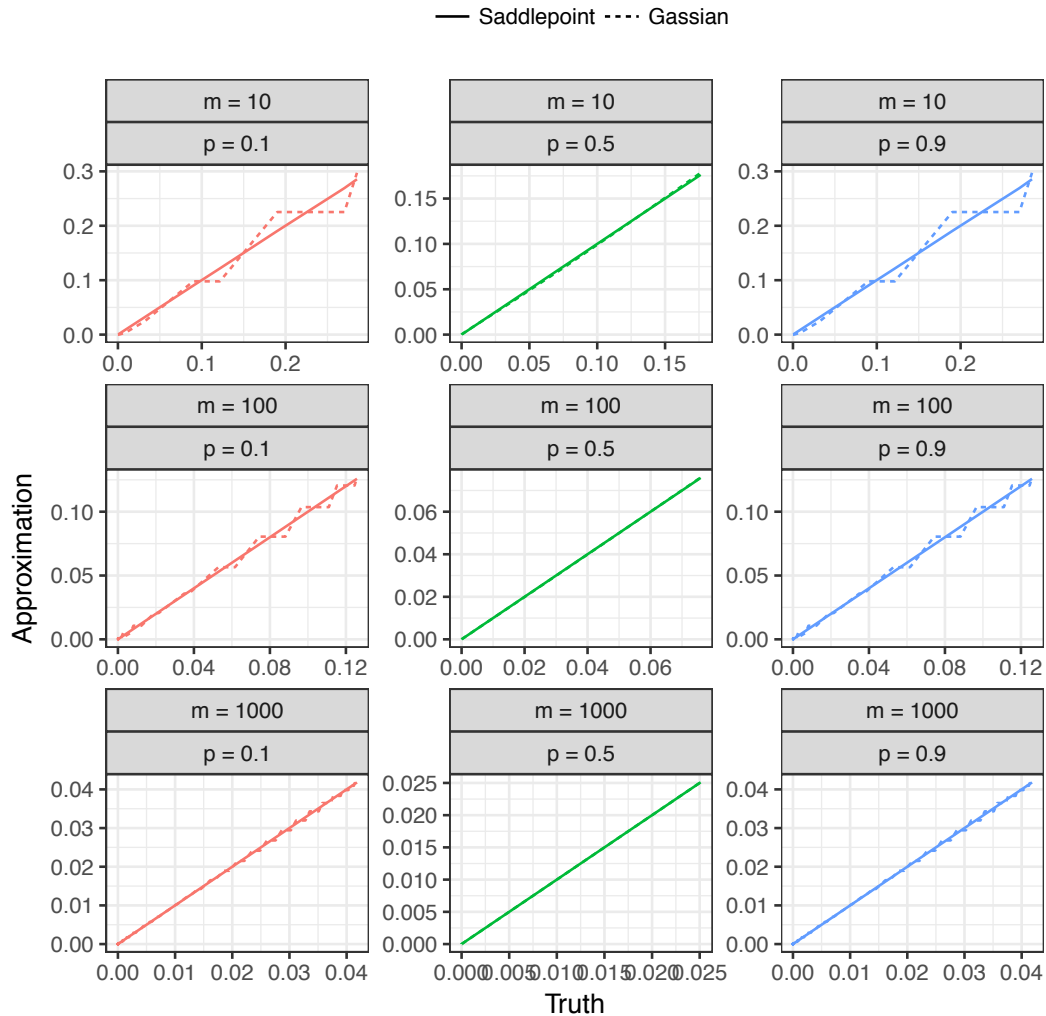
Figure 4: Comparison of PDF between truth and approximation for $n = 10$.

```
data[,Error := (truth-approx)]
data[,p:=paste0('p = ',p)]
data = melt(data,measure.vars = c('Relative error','Error'),
        value.name = 'error', variable.name = 'type')

ggplot(data[Method == 'saddle'],aes(x=s,y=error,color=p))+
        geom_point(alpha=0.5)+theme_bw()+
        facet_wrap(type~p,scales='free')+
        xlab('Quantile')+
        ylab(expression(
            'Truth - Approximation'~~~~~~~~~~~~~~~~frac(Truth - Approximation,Truth)))+
        scale_color_discrete(guide='none') +
        geom_vline(xintercept=200*0.5,color='green',linetype='longdash')+
        geom_vline(xintercept=200*0.1,color='red',linetype='longdash')+
        geom_vline(xintercept=200*0.9,color='blue',linetype='longdash')
```

Healthcare monitoring

In the second example, we used a health system monitoring dataset by Benneyan and Taşeli (2010). To improve compliance with medical devices, healthcare organizations often monitor bundle reliability statistics, each representing a percentage of patient compliance. Suppose n_i and p_i represent the number of patients and percentage of compliant patients for element i in the bundle, and n and p take the following values:
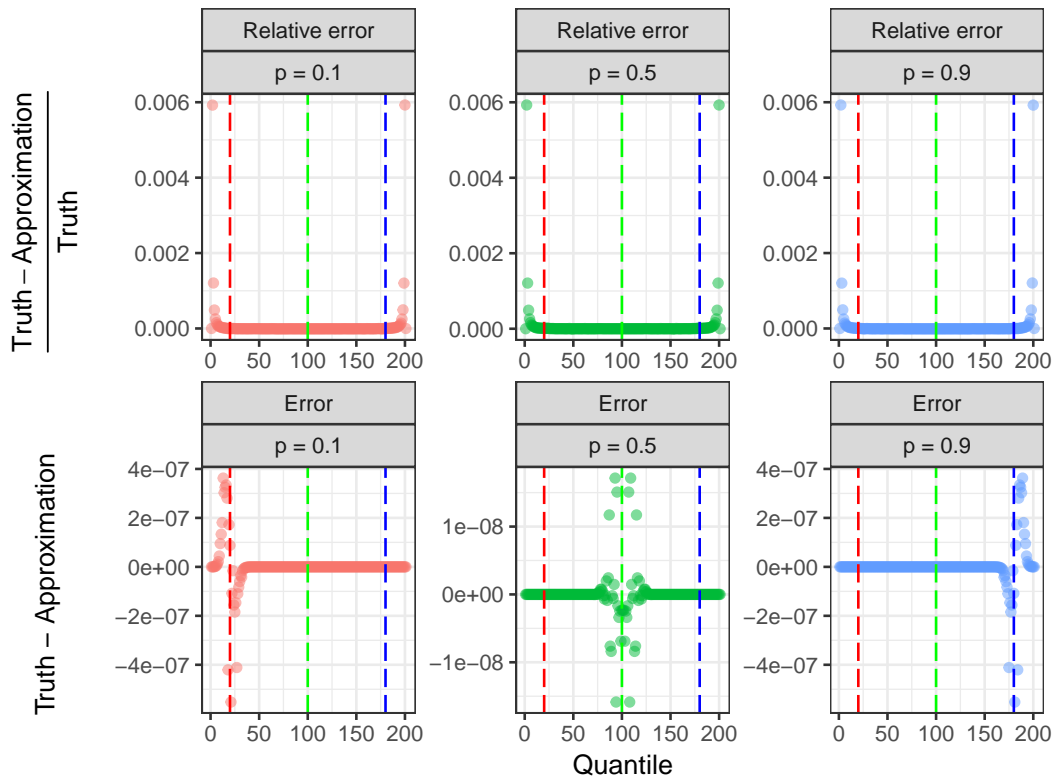
Figure 5: Difference in PDF between truth and the saddlepoint approximation.

```
size=as.integer(c(12, 14, 4, 2, 20, 17, 11, 1, 8, 11))
prob=c(0.074, 0.039, 0.095, 0.039, 0.053, 0.043, 0.067, 0.018, 0.099, 0.045)
```

Since it is difficult to find an analytical solution for the density, we estimated the density with simulation (1e8 trials) and treated it as the ground truth. We then compared simulations with 1e3, 1e4, 1e5, and 1e6 trials, and the saddlepoint approximation to the ground truth. (Note that running simulation will take several minutes.)

```
# Sinib:
approx=dsinib(0:sum(size),size,prob)
approx=data.frame(s=0:sum(size),pdf=approx,type='saddlepoint')

# Gauss:
gauss_approx = d_norm_app(0:sum(size),size,prob)
gauss_approx = data.frame(s=0:sum(size),pdf=gauss_approx,type='gauss')

# Simulation:
data=foreach(n_sim=10^c(3:6,8),.combine='rbind')%do%{
        ptm=proc.time()
        n_binom=length(prob)
        set.seed(42)
        mat=matrix(rbinom(n_sim*n_binom,size,prob),nrow=n_binom,ncol=n_sim)

        S=colSums(mat)
        sim=sapply(X = 0:sum(size), FUN = function(x) {sum(S==x)/length(S)})
        data.table(s=0:sum(size),pdf=sim,type=n_sim)

}

data=rbind(data,gauss_approx,approx)
truth=data[type=='1e+08',]

merged=merge(truth[,list(s,pdf)],data,by='s',suffixes=c('_truth','_approx'))
merged=merged[type!='1e+08',]
```
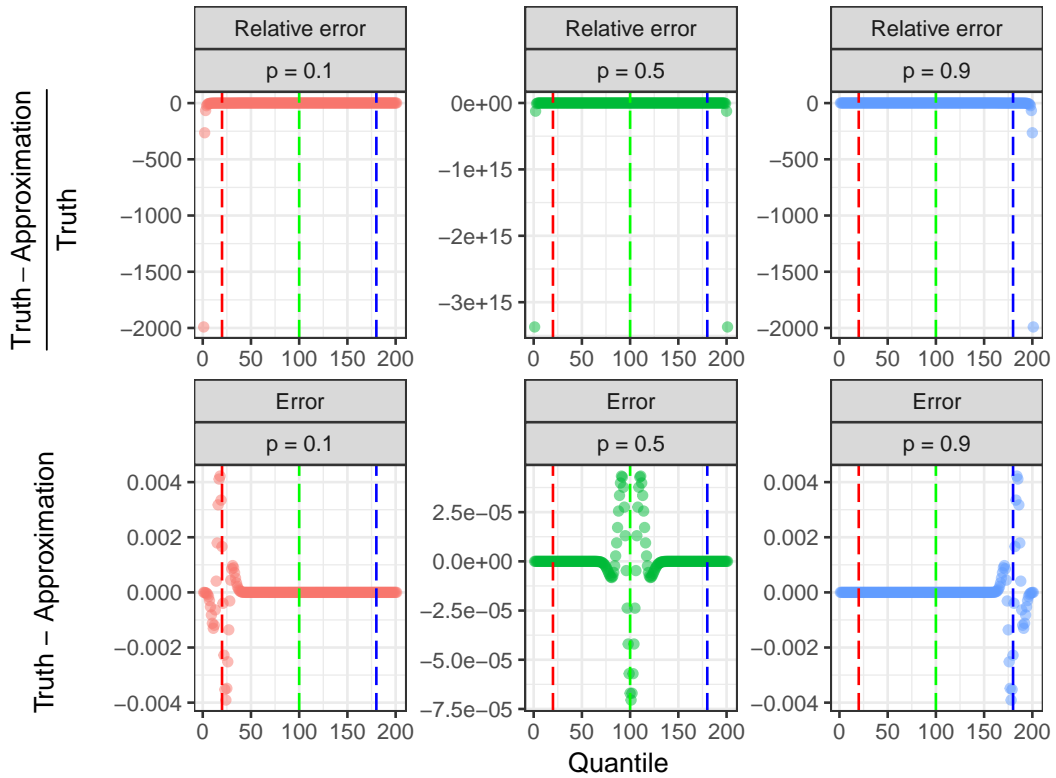
Figure 6: Difference in PDF between truth and the Gaussian approximation.

```
ggplot(merged,aes(pdf_truth,pdf_approx))+
        geom_point()+
        facet_grid(~type)+
        geom_abline(intercept=0,slope=1)+
        theme_bw()+
        xlab('Truth')+
        ylab('Approximation')
```
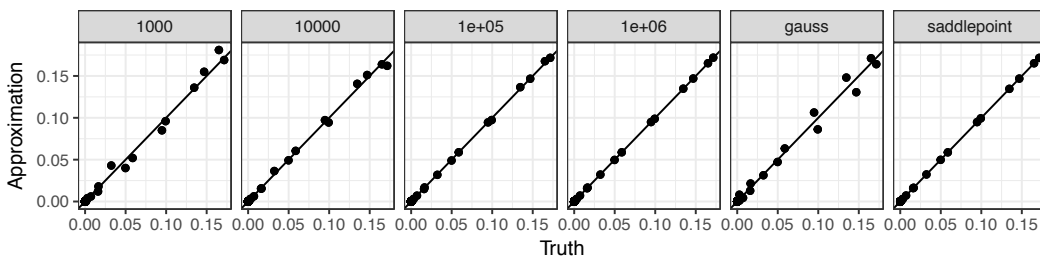

Figure 7: Comparison of PDF between truth and approximation.

Figure 7 shows that the simulation with 1e6 trials and the saddlepoint approximation are indistinguishable from the ground truth, while the Gaussian method and estimates with fewer simulations show clear deviations from the truth. To further examine the magnitude of deviation, we plot the difference in PDF between the truth and the approximation:

```
merged[,Error:=pdf_truth-pdf_approx]
merged[,`Relative Error`:=(pdf_truth-pdf_approx)/pdf_truth]
merged = melt(merged,measure.vars = c('Error','Relative Error'),variable.name = 'error_type',
        value.name = 'error')

p2=ggplot(merged,aes(s,error))+
        geom_point()+
```

```
facet_grid(error_type~type,scales = 'free_y')+
theme_bw()+
xlab('Outcome')+
ylab('Truth-Approx')+
xlim(0,20) +
ylab(expression(frac(Truth - Approximation,Truth)~~~~~~~'Truth - Approximation'))
```
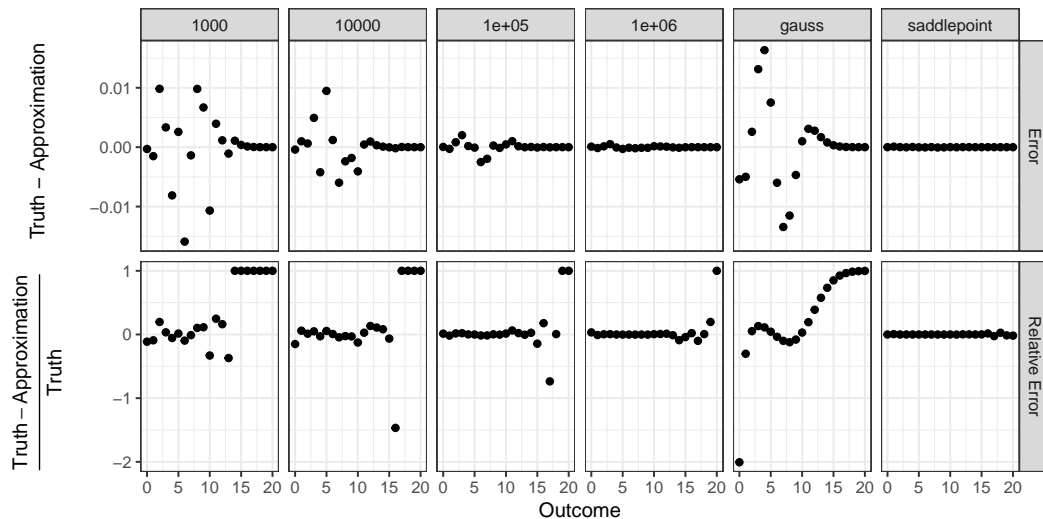

Figure 8: Error and relative error between truth and approximation.

Figure 8 shows that that the saddlepoint method and the simulation with 1e6 draws both provide good approximations, while the Gaussian approximation and simulations of smaller sizes show clear deviations. We also note that the saddlepoint approximation is 5 times faster than the simulation of 1e6 trials.

```
ptm=proc.time()
n_binom=length(prob)
mat=matrix(rbinom(n_sim*n_binom,size,prob),nrow=n_binom,ncol=n_sim)
S=colSums(mat)
sim=sapply(X = 0:sum(size), FUN = function(x) {sum(S==x)/length(S)})
proc.time()-ptm
#    user  system elapsed
#   1.008   0.153   1.173

ptm=proc.time()
approx=dsinib(0:sum(size),size,prob)
proc.time()-ptm
#   user  system elapsed
#  0.025   0.215   0.239
```

Conclusion and future direction

In this paper, we presented an implementation of the saddlepoint method to approximate the distribution of the sum of independent and non-identical binomials. We assessed the accuracy of the method by comparing it with first, the analytical solution in the simple case of two binomials, and second, the simulated ground truth on a real-world dataset in healthcare monitoring. These assessments suggest that the saddlepoint method generally provides an approximation superior to simulation in terms of both speed and accuracy, and outperforms the Gaussian approximation in terms of accuracy. Overall, the **sinib** package addresses the gap between the theory and implementation on the approximation of sum of independent and non-identical binomial random variables.

In the future, we aim to explore other approximation methods such as the Kolmogorov approximation and the Pearson curve approximation described by Butler and Stephens (2016).

Bibliography

M. Akahira and K. Takahashi. A Higher-Order Large-Deviation Approximation for the Discrete Distributions. *Journal of the Japan Statistical Society*, 31(2):257–267, 2001. URL https://doi.org/10.1080/03610929908832322. [p473]

J. Arthur Woodward and C. G. S. Palmer. On the Exact Convolution of Discrete Random Variables. *Applied Mathematics and Computation*, 83(1):69–77, 1997. URL https://doi.org/10.1016/S0096-3003(96)00047-1. [p473]

J. C. Benneyan and A. Taşeli. Exact and Approximate Probability Distributions of Evidence-Based Bundle Composite Compliance Measures. *Health Care Management Science*, 13(3):193–209, 2010. URL https://doi.org/10.1007/s10729-009-9123-x. [p472, 479]

K. Butler and M. A. Stephens. The Distribution of a Sum of Independent Binomial Random Variables. *Methodology and Computing in Applied Probability*, 19(2):557–571, 2016. URL https://doi.org/10.1007/s11009-016-9533-4. [p473, 482]

H. E. Daniels. Saddlepoint Approximations in Statistics. *The Annals of Mathematical Statistics*, 25(4): 631–650, 1954. URL http://www.jstor.org/stable/2236650. [p473]

H. E. Daniels. Tail Probability Approximations. *International Statistical Review / Revue Internationale de Statistique*, 55(1):37–48, 1987. URL http://www.jstor.org/stable/1403269. [p473, 474]

R. Eisinga, M. Te Grotenhuis, and B. Pelzer. Saddlepoint Approximations for the Sum of Independent Non-Identically Distributed Binomial Random Variables. *Statistica Neerlandica*, 67(2):190–201, 2013. URL https://doi.org/10.1111/stan.12002. [p472, 473]

N. L. Johnson, A. W. Kemp, and S. Kotz. *Univariate Discrete Distributions*. Johnson/Univariate Discrete Distributions. John Wiley & Sons, Hoboken, NJ, USA, 2005. URL https://doi.org/10.1002/0471715816. [p472]

J. K. Jolayemi. A Unified Approximation Scheme for the Convolution of Independent Binomial Variables. *Applied Mathematics and Computation*, 49(2-3):269–297, 1992. URL https://doi.org/10.1016/0096-3003(92)90030-5. [p472]

S. Kotz and N. L. Johnson. Effects of False and Incomplete Identification of Defective Items on the Reliability of Acceptance Sampling. *Operations Research*, 32(3):575–583, 1984. URL https://pubsonline.informs.org/doi/abs/10.1287/opre.32.3.575. [p472]

R. Lugannani and S. Rice. Saddle Point Approximation for the Distribution of the Sum of Independent Random Variables. *Advances in Applied Probability*, 12(2):475, 1980. URL http://www.jstor.org/stable/1426607. [p473]

E. M. Schmidt, J. Zhang, W. Zhou, J. Chen, K. L. Mohlke, Y. E. Chen, and C. J. Willer. GREGOR: Evaluating Global Enrichment of Trait-Associated Variants in Epigenomic Features Using a Systematic, Data-Driven Approach. *Bioinformatics*, 31(16):2601–2606, 2015. URL https://doi.org/10.1093/bioinformatics/btv201. [p472]

H. Yili. On Computing the Distribution Function for the Sum of Independent and Non-Identical Random Indicators . URL https://pdfs.semanticscholar.org/fe97/c1358ec01c86cb8bbc4574fa064748f37e94.pdf. [p473]

Boxiang Liu
Stanford University
300 Pasteur Drive, Stanford, CA
United States
ORCID: 0000-0002-2595-4463
bliu2@stanford.edu

Thomas Quertermous
Stanford University
300 Pasteur Drive, Stanford, CA
United States
tomq1@stanford.edu