

binGroup: A Package for Group Testing

by Christopher R. Bilder, Boan Zhang, Frank Schaarschmidt and Joshua M. Tebbs

Abstract When the prevalence of a disease or of some other binary characteristic is small, group testing (also known as pooled testing) is frequently used to estimate the prevalence and/or to identify individuals as positive or negative. We have developed the **binGroup** package as the first package designed to address the estimation problem in group testing. We present functions to estimate an overall prevalence for a homogeneous population. Also, for this setting, we have functions to aid in the very important choice of the group size. When individuals come from a heterogeneous population, our group testing regression functions can be used to estimate an individual probability of disease positivity by using the group observations only. We illustrate our functions with data from a multiple vector transfer design experiment and a human infectious disease prevalence study.

Introduction

Group testing, where individuals are composited into pools to screen for a binary characteristic, has a long history of successful application in areas such as human infectious disease detection, veterinary screening, drug discovery, and insect vector pathogen transmission (Pilcher et al., 2005; Peck, 2006; Remlinger et al., 2006; Tebbs and Bilder, 2004). Group testing works well in these settings because the prevalence is usually small and individual specimens (e.g., blood, urine, or cattle ear notches) can be composited without loss of diagnostic test accuracy. Group testing is performed often by assigning each individual to a group and testing every group for a positive or negative outcome of the binary characteristic. Using these group responses alone, estimates of overall prevalence or subject-specific probabilities of positivity can be found. When further individual identification of the binary characteristic is of interest, re-testing of specimens within positive groups can be performed to decode the individual positives from the negatives. There are other variants to how group testing is applied, and some will be discussed in this paper. A recent review of group testing for estimation and identification is found in Hughes-Oliver (2006).

Our **binGroup** package (Zhang et al., 2010) is the first dedicated to the group testing estimation problem within homogeneous or heterogeneous populations. We also provide functions to determine the optimal group size based on prior knowledge of what the overall prevalence may be. All of our functions

have been written in familiar formats to those where individual testing is used (e.g., `binom.confint()` in **binom** (Dorai-Raj, 2009) or `glm()` in **stats**).

Homogeneous populations

Group testing has been used traditionally in settings where one overall prevalence of a binary characteristic within a homogeneous population is of interest. Typically, one assumes that each individual is independent and has the same probability p of the characteristic, so that p is the overall prevalence. In the next section, we will consider the situation where individuals have different probabilities of positivity. Here, we let θ denote the probability that a group of size s is positive. One can show then $p = 1 - (1 - \theta)^{1/s}$ when diagnostic testing is perfect. This equation plays a central role in making estimates and inferences about individuals when only the group responses are known and each individual is within only one group.

We have written two functions to calculate a confidence interval for p . First, the `bgtCI()` function calculates intervals for p when a common group size is used throughout the sample. For example, Ornaghi et al. (1999) estimate the probability that the female *Delphacodes kuscheli* (planthopper) transfers the Mal Rio Cuarto (MRC) virus to maize crops. In stage 4 of the experiment, $n = 24$ enclosed maize plants each had $s = 7$ planthopper vectors placed on them for forty-eight hours and there were $y = 3$ plants that tested positive for the MRC virus after two months. The 95% confidence interval for the probability of transmission p is calculated by

```
> bgtCI(n = 24, y = 3, s = 7,
+   conf.level = 0.95,
+   alternative = "two.sided",
+   method = "Score")

95 percent Score confidence interval:
 [ 0.006325, 0.05164 ]
Point estimate: 0.0189
```

where the score (Wilson) interval was used. While the score interval is usually one of the best in terms of coverage (Tebbs and Bilder, 2004), other intervals calculated by `bgtCI()` include the Clopper-Pearson, the asymptotic second-order corrected, and the Wald. The maximum likelihood estimate for p is $1 - (1 - 3/24)^{1/7} = 0.0189$.

Group testing is applied usually with equally sized groups. When a common group size is not used, perhaps due to physical or design constraints, our `bgtvs()` function can calculate the exact interval proposed by Hepworth (1996, equation 5). The arguments to `bgtvs()` include the following vectors: s , the different group sizes occurring in the design; n ,

the corresponding numbers of groups for each group size; and \bar{y} , the corresponding numbers of observed positive groups for each group size. Note that the algorithm becomes computationally expensive when the number of different group sizes is more than three.

One of the most important design considerations is the choice of the group size. Choosing a group size that is too small may result in few groups testing positive, so more tests are used than necessary. Choosing a group size that is too large may result in almost all groups testing positive, which leads to a poor estimate of p . As a rule of thumb, one tries to choose a group size so that about half of the groups test positive. More formally, one can choose an s that minimizes the mean square error (MSE) for a fixed n and a prior estimate of p (Swallow, 1985). If we use a prior prevalence estimate of 0.0189, 24 groups, and a maximum possible group size of 100, our `estDesign()` function finds the optimal choice of s to be 43:

```
> estDesign(n = 24, smax = 100, p.tr = 0.0189)
group size s with minimal mse(p) = 43
```

```
$varp [1] 3.239869e-05
```

```
$mse [1] 3.2808e-05
```

```
$bias [1] 0.0006397784
```

```
$exp [1] 0.01953978
```

The function provides the corresponding variance, MSE, bias, and expected value for the maximum likelihood estimator of p . While $s = 43$ is optimal for this example, large group sizes can not necessarily be used in practice (e.g., dilution effects may prevent using a large group size), but this can still be used as a goal.

Our other functions for homogeneous population settings include `bgtTest()`, which calculates a p -value for a hypothesis test involving p . Also, `bgtPower()` calculates the power of the hypothesis test. Corresponding to `bgtPower()`, the `nDesign()` and `sDesign()` functions calculate the power with increasing n or s , respectively, with `plot.bgtDesign()` providing a plot. These functions allow researchers to design their own experiment in a similar manner to that described in Schaarschmidt (2007).

Heterogeneous populations

When covariates for individuals are available, we can model the probability of positivity as with any binary regression model. However, the complicating aspect here is that only the group responses may be available. Also, if both group responses and responses from re-tests are available, the correlation between these responses makes the analysis more difficult. Vansteelandt et al. (2000) and Xie

(2001) have both proposed ways to fit these models. Vansteelandt et al. (2000) use a likelihood function written in terms of the initial group responses and maximize it to obtain the maximum likelihood estimates of the model parameters. This fitting procedure can not be used when re-tests are available. Xie (2001) writes the likelihood function in terms of the unobserved individual responses and uses the EM algorithm for estimation. This approach has an advantage over Vansteelandt et al. (2000) because it can be used in more complicated settings such as when re-tests are available or when individuals appear in multiple groups (e.g., matrix or array-based pooling). However, while Xie's fitting procedure is more general, it can be very slow to converge for some group testing protocols.

The `gtreg()` function fits group testing regression models in situations where individuals appear in only one group and no re-tests are performed. The function call is very similar to that of `glm()` in the `stats` package. Additional arguments include sensitivity and specificity of the group test; group numbers for the individuals, and specification of either the Vansteelandt or Xie fitting methods. Both model-fitting methods will produce approximately the same estimates and corresponding standard errors.

We illustrate the `gtreg()` function with data from Vansteelandt et al. (2000). The data were obtained through a HIV surveillance study of pregnant women in rural parts of Kenya. For this example, we model the probability that a women is HIV positive using the covariates age and highest attained education level (treated as ordinal). The data structure is

```
> data(hivsurv)
> tail(hivsurv[,c(3,5,6:8)], n = 7)
  AGE  EDUC HIV gnum groupres
422  29    3  1  85      1
423  17    2  0  85      1
424  18    2  0  85      1
425  18    2  0  85      1
426  22    3  0  86      0
427  30    2  0  86      0
428  34    3  0  86      0
```

Each individual within a group (`gnum` is the group number) is given the same group response (`groupres`) within the data set. For example, individual #422 is positive (1) for HIV, and this leads to all individuals within group #85 to have a positive group response. Note that the individual HIV responses are known here because the purpose of the original study was to show group testing works as well as individual testing (Verstraeten et al., 1998). Continuing, the `gtreg()` function fits the model and the fit is summarized with `summary()`:

```
> fit1 <- gtreg(formula = groupres ~ AGE + EDUC,
+ data = hivsurv, groupn = gnum, sens = 0.99,
+ spec = 0.95, linkf = "logit",
+ method = "Vansteelandt")
```

```
> summary(fit1)

Call: gtreg(formula = groupres ~ AGE + EDUC,
  data = hivsurv, groupn = gnum, sens = 0.99,
  spec = 0.95, linkf = "logit",
  method = "Vansteelandt")

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1811 -0.9384 -0.8219  1.3299  1.6696

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.99039    1.59911  -1.870  0.0615 .
AGE          -0.05163    0.06748  -0.765  0.4443
EDUC          0.73621    0.43885   1.678  0.0934 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*'
0.05 '.' 0.1 ' ' 1

Null deviance: 191.4 on 427 degrees of freedom
Residual deviance: 109.4 on 425 degrees of freedom
AIC: 115.4

Number of iterations in optim(): 138
```

The results from `gtreg()` are stored in `fit1` here, which has a "gt" class type. The estimated model can be written as

$$\text{logit}(\hat{p}_{ik}) = -2.99 - 0.0516\text{Age}_{ik} + 0.7362\text{Educ}_{ik}$$

where \hat{p}_{ik} is the estimated probability that the i th individual in the k th group is positive. In addition to the `summary.gt()` function, method functions to find residuals and predicted values are available.

We have also written a function `sim.g()` that simulates group test responses for a given binary regression model. The options within it allow for a user-specified set of covariates or one covariate that is simulated from a gamma distribution. Individuals are randomly put into groups of a specified size by the user. The function can also be used to simulate group testing data from a homogeneous population by specifying zero coefficients for covariates.

One of the most important innovations in group testing is the development of matrix or array-based pooling (Phatarfod and Sudbury, 1994; Kim et al., 2007). In this setting, specimens are placed into a matrix-like grid so that they can be pooled within each row and within each column. Potentially positive individuals occur at the intersection of positive rows and columns. If identification of these positive individuals is of interest, individual re-testing can be done on specimens at these intersections. With the advent of high-throughput screening, matrix pooling has become easier to perform because pooling and testing is done with minimal human intervention.

The `gtreg.mp()` function fits a group testing regression model in a matrix pooling setting. The row and column group responses can be used alone to fit the model. If individual re-testing is performed on

the positive row and column intersections, these re-tests can be included when fitting the model. Note that the speed of model convergence can be improved by including re-tests. Within `gtreg.mp()`, we implement the EM algorithm given by Xie (2001) for matrix pooling settings where individual re-tests may or may not be performed. Due to the complicated response nature of matrix pooling, this algorithm involves using Gibbs sampling for the E-step in order to approximate the conditional expected values of a positive individual response.

Through personal communication with Minge Xie, we discovered that while he suggested the model fitting procedure could be used for matrix pooling, he had not implemented it; therefore, to our knowledge, this is the first time group testing regression models for a matrix pooling setting have been put into practice. Zhang and Bilder (2009) provide a technical report on the model fitting details. We hope that the `gtreg.mp()` function will encourage researchers to include covariates when performing matrix pooling rather than assume one common p , as has been done in the past.

The `sim.mp()` function simulates matrix pooling data. In order to simulate the data for a 5×6 and a 4×5 matrix, we can implement the following:

```
> set.seed(9128)
> sala <- sim.mp(par = c(-7,0.1), n.row = c(5,4),
+ linkf = "logit", n.col = c(6,5), sens = 0.95,
+ spec = 0.95)
> sal <- sala$dframe
> head(sal)
      x col.resp row.resp coln rown arrayn retest
1 29.961      0      0     1     1      1     NA
2 61.282      0      1     1     2      1     NA
3 34.273      0      1     1     3      1     NA
4 46.190      0      0     1     4      1     NA
5 39.438      0      1     1     5      1     NA
6 45.880      1      0     2     1      1     NA
```

where `sal` contains the column, row, and re-test responses along with one covariate x . The `coln`, `rown`, and `arrayn` variables are the column, row, and array numbers, respectively, for the responses. The covariate is simulated using the default gamma distribution with shape parameter 20 and scale parameter 2 (a user-specified matrix of covariates can also be used with the function). The `par` argument gives the coefficients in the model of $\text{logit}(p_{ijk}) = -7 + 0.1x_{ijk}$ where x_{ijk} and p_{ijk} are the covariate and positivity probability, respectively, for the individual in row i , column j , and array k . We fit a model to the data using the following:

```
> fit1mp <- gtreg.mp(formula = cbind(col.resp,
+ row.resp) ~ x, data = sal, coln = coln,
+ rown = rown, arrayn = arrayn, sens = 0.95,
+ spec = 0.95, linkf = "logit", n.gibbs = 2000)
> coef(fit1mp)
(Intercept)          x
-6.23982684  0.08659878
```

The coefficients are similar to the ones used to simulate the data. Methods to summarize the model's fit and to perform predictions are also available.

Conclusion

Group testing is used in a vast number of applications where a binary characteristic is of interest and individual specimens can be composited. Our package combines together the most often used and recommended confidence intervals for p . Also, our package makes the regression methods of Vansteelandt et al. (2000) and Xie (2001) easily accessible for the first time. We hope this will encourage researchers to take into account potentially important covariates in a group testing setting.

We see the current form of the **binGroup** package as a beginning rather than an end to meeting researcher needs. There are many additions that would be further helpful to researchers. For example, there are a number of re-testing protocols, such as halving (Gastwirth and Johnson, 1994) or sub-dividing positive groups of any size (Kim et al., 2007), that could be implemented, but would involve a large amount of new programming due to the complex nature of the re-testing. Also, the **binGroup** package does not have any functions solely for individual identification of a binary characteristic. For example, the optimal group size to use for identification alone is usually different to the optimal group size to use when estimating p . Given these desirable extensions, we encourage others to send us their functions or write new functions of their own. We would be willing to work with anyone to include them within the **binGroup** package. This would enable all researchers to have one group testing package rather than many small packages with much duplication.

Acknowledgements

This research is supported in part by Grant R01 AI067373 from the National Institutes of Health.

Bibliography

- S. Dorai-Raj. **binom**: Binomial Confidence Intervals for Several Parameterizations, 2009. URL <http://cran.r-project.org/web/packages/binom>. R package version 1.0-4.
- J. Gastwirth and W. Johnson. Screening with cost-effective quality control: Potential applications to HIV and drug testing. *Journal of the American Statistical Association*, 89:972–981, 1994.
- G. Hepworth. Exact confidence intervals for proportions estimated by group testing. *Biometrics*, 52: 1134–1146, 1996.
- J. Hughes-Oliver. Pooling experiments for blood screening and drug discovery. In *Screening: Methods for Experimentation in Industry, Drug Discovery, and Genetics*, edited by A. Dean and S. Lewis, New York: Springer.
- H. Kim, M. Hudgens, J. Dreyfuss, D. Westreich, and C. Pilcher. Comparison of group testing algorithms for case identification in the presence of test error. *Biometrics*, 63:1152–1163, 2007.
- J. Ornaghi, G. March, G. Boito, A. Marinelli, A. Beviacqua, J. Giuggia, and S. Lenardon. Infectivity in natural populations of *Delphacodes kuscheli* vector of 'Mal Rio Cuarto' virus. *Maydica*, 44:219–223, 1999.
- C. Peck. Going after BVD. *Beef*, 42:34–44, 2006.
- R. Phatarfod and A. Sudbury. The use of a square array scheme in blood testing. *Statistics in Medicine*, 13:2337–2343, 1994.
- C. Pilcher, S. Fiscus, T. Nguyen, E. Foust, L. Wolf, D. Williams, R. Ashby, J. O'Dowd, J. McPherson, B. Stalzer, L. Hightow, W. Miller, J. Eron, M. Cohen, and P. Leone. Detection of acute infections during HIV testing in North Carolina. *New England Journal of Medicine*, 352:1873–1883, 2005.
- K. Remlinger, J. Hughes-Oliver, S. Young, and R. Lam. Statistical design of pools using optimal coverage and minimal collision. *Technometrics*, 48: 133–143, 2006.
- W. Swallow. Group testing for estimating infection rates and probabilities of disease transmission. *Phytopathology*, 75:882–889, 1985.
- F. Schaarschmidt. Experimental design for one-sided confidence intervals or hypothesis tests in binomial group testing. *Communications in Biometry and Crop Science*, 2:32–40, 2007.
- J. Tebbs and C. Bilder. Confidence interval procedures for the probability of disease transmission in multiple-vector-transfer designs. *Journal of Agricultural, Biological, and Environmental Statistics*, 9: 75–90, 2004.
- S. Vansteelandt, E. Goetghebeur, and T. Verstraeten. Regression models for disease prevalence with diagnostic tests on pools of serum samples. *Biometrics*, 56:1126–1133, 2000.
- T. Verstraeten, B. Farah, L. Duchateau, and R. Matu. Pooling sera to reduce the cost of HIV surveillance: a feasibility study in a rural Kenyan district. *Tropical Medicine and International Health*, 3: 747–750, 1998.
- M. Xie. Regression analysis of group testing samples. *Statistics in Medicine*, 20:1957–1969, 2001.

B. Zhang and C. Bilder. The EM algorithm for group testing regression models under matrix pooling. Technical Report, University of Nebraska-Lincoln, Department of Statistics. URL <http://www.chrisbilder.com/grouptesting>.

B. Zhang, C. Bilder, and F. Schaarschmidt. **binGroup**: Evaluation and experimental design for binomial group testing, 2010. URL <http://cran.r-project.org/web/packages/binGroup>. R package version 1.0-5.

Christopher R. Bilder
Department of Statistics
University of Nebraska-Lincoln, Lincoln, NE
chris@chrisbilder.com

Boan Zhang
Department of Statistics
University of Nebraska-Lincoln, Lincoln, NE
boan.zhang@huskers.unl.edu

Frank Schaarschmidt
Institut für Biostatistik
Leibniz Universität Hannover, Germany
schaarschmidt@biostat.uni-hannover.de

Joshua M. Tebbs
Department of Statistics
University of South Carolina, Columbia, SC
tebbs@stat.sc.edu