Supplementary Material for
"Simultaneous confidence intervals for comparing margins of multivariate binary data"

by Bernhard Klingenberg and Ville Satopää

1 Simulation settings for $J = 6, 10, 15$ and 20

The multinomial probabilities $\pi_i(\boldsymbol{a}), i = 1, 2$ used in the simulations for constructing Table 3 are based on the association structure displayed in Table A. The diagonal shows the marginal probabilities (in bold) $\pi(j)$ and the entries below the diagonal show the joint probabilities $\pi_i(j, j')$ for the success probabilities of the two variables Y_{ij} and $Y_{ij'}$ that are implied by $\boldsymbol{\pi}_i(\boldsymbol{a})$. Similar to Table 1, entries above the diagonal show the odds ratio between Y_{ij} and $Y_{ij'}$. For the simulations with $J = k$ the first k variables $Y_{i1}, \ldots, Y_{ik}$ displayed in Table A were used, $k = 6, 10, 15$ and 20, so that for instance for the simulation with $J = 10$, the 10×10 upper subtable of Table A is relevant.

	$Y_{1,1}$	$Y_{1,2}$	$Y_{1,3}$	$Y_{1,4}$	$Y_{1,5}$	$Y_{1,6}$	$Y_{1,7}$	$Y_{1,8}$	$Y_{1,9}$	$Y_{1,10}$	$Y_{1,11}$	$Y_{1,12}$	$Y_{1,13}$	$Y_{1,14}$	$Y_{1,15}$	$Y_{1,16}$	$Y_{1,17}$	$Y_{1,18}$	$Y_{1,19}$	$Y_{1,20}$
$Y_{1,1}$	**0.26**	12.4	9.5	11.8	15.6	4.1	9.2	9.9	8.3	14.3	5.5	10.9	11.2	5.4	6.3	5.9	4.2	3.0	5.7	6.6
$Y_{1,2}$	0.12	**0.17**	7.1	16.0	23.1	4.6	9.8	4.2	4.1	27.9	5.3	25.1	5.1	4.4	8.9	9.2	7.3	2.6	4.6	6.7
$Y_{1,3}$	0.04	0.03	**0.06**	9.6	10.0	1.8	8.8	7.5	9.3	10.6	4.2	11.2	9.9	3.9	7.9	7.1	2.5	1.9	3.8	6.9
$Y_{1,4}$	0.05	0.05	0.02	**0.07**	17.6	4.4	23.2	2.4	3.5	15.8	2.7	17.8	4.0	1.3	4.3	3.8	2.9	3.5	2.2	2.9
$Y_{1,5}$	0.16	0.13	0.04	0.06	**0.23**	5.1	16.3	8.9	9.0	20.9	5.6	26.7	13.7	4.2	9.2	10.2	7.7	3.0	4.1	6.7
$Y_{1,6}$	0.15	0.11	0.03	0.05	0.14	**0.33**	5.8	2.9	2.9	3.3	2.5	2.8	2.0	1.4	2.3	1.7	2.5	7.3	3.0	3.6
$Y_{1,7}$	0.13	0.10	0.04	0.06	0.14	0.13	**0.20**	6.9	7.0	13.3	4.2	9.8	9.9	3.6	7.6	6.3	5.2	3.5	2.5	4.7
$Y_{1,8}$	0.08	0.04	0.02	0.01	0.07	0.06	0.06	**0.11**	32.5	3.1	7.6	2.7	10.6	11.3	3.9	3.8	6.7	2.5	4.9	9.0
$Y_{1,9}$	0.06	0.04	0.02	0.02	0.06	0.05	0.05	0.06	**0.09**	4.34	12.1	3.3	8.9	15.5	3.6	2.1	4.4	2.3	6.9	32.1
$Y_{1,10}$	0.06	0.06	0.02	0.03	0.06	0.05	0.06	0.02	0.02	**0.08**	3.5	32.0	6.0	2.7	8.6	9.3	2.3	3.0	2.7	5.2
$Y_{1,11}$	0.03	0.02	0.01	0.01	0.03	0.03	0.02	0.02	0.02	0.01	**0.05**	3.4	3.3	7.5	1.8	2.2	3.3	1.8	7.9	14.2
$Y_{1,12}$	0.04	0.04	0.02	0.03	0.05	0.03	0.04	0.01	0.01	0.03	0.01	**0.05**	6.9	2.4	6.2	7.3	3.8	2.8	3.9	5.9
$Y_{1,13}$	0.03	0.02	0.01	0.01	0.03	0.02	0.03	0.02	0.02	0.01	0.01	0.01	**0.04**	5.5	44.9	25.5	2.4	5.1	3.2	4.4
$Y_{1,14}$	0.02	0.02	0.01	0.00	0.02	0.02	0.02	0.02	0.02	0.01	0.01	0.00	0.01	**0.04**	2.1	3.8	4.4	1.3	3.6	7.1
$Y_{1,15}$	0.02	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.00	0.01	0.01	0.00	**0.02**	58.7	1.4	1.7	1.8	0.8
$Y_{1,16}$	0.01	0.01	0.01	0.00	0.02	0.01	0.01	0.01	0.00	0.01	0.00	0.01	0.01	0.00	0.01	**0.02**	4.0	1.7	1.8	1.6
$Y_{1,17}$	0.01	0.01	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.02**	0.0	0.0	7.0
$Y_{1,18}$	0.01	0.01	0.00	0.00	0.01	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.02**	1.8	0.0
$Y_{1,19}$	0.01	0.01	0.00	0.00	0.01	0.01	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.01**	5.6
$Y_{1,20}$	0.01	0.01	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.02**

Table A1: Association structure for simulations in Table 3 for the first group ($i = 1$).

	$Y_{2,1}$	$Y_{2,2}$	$Y_{2,3}$	$Y_{2,4}$	$Y_{2,5}$	$Y_{2,6}$	$Y_{2,7}$	$Y_{2,8}$	$Y_{2,9}$	$Y_{2,10}$	$Y_{2,11}$	$Y_{2,12}$	$Y_{2,13}$	$Y_{2,14}$	$Y_{2,15}$	$Y_{2,16}$	$Y_{2,17}$	$Y_{2,18}$	$Y_{2,19}$	$Y_{2,20}$
$Y_{2,1}$	**0.24**	12.5	11.0	9.6	19.3	4.3	14.6	10.8	10.9	11.9	6.2	9.8	13.4	5.7	4.9	9.3	7.1	2.2	6.6	4.5
$Y_{2,2}$	0.08	**0.11**	3.9	15.1	25.9	5.4	8.6	4.6	3.3	22.4	4.0	24.2	6.2	4.8	4.4	6.2	5.0	4.0	4.7	2.9
$Y_{2,3}$	0.04	0.02	**0.06**	4.1	10.7	3.0	16.8	12.0	16.8	8.8	6.6	7.3	13.2	4.9	8.1	12.0	6.0	0.8	2.6	7.4
$Y_{2,4}$	0.03	0.03	0.01	**0.05**	26.9	7.1	36.8	3.4	3.0	11.7	2.5	14.3	4.9	2.9	3.2	4.2	3.8	6.2	3.4	1.5
$Y_{2,5}$	0.14	0.08	0.04	0.04	**0.19**	5.9	23.5	11.2	12.1	24.7	6.0	23.0	15.1	5.4	6.8	9.7	5.8	4.4	13.4	5.1
$Y_{2,6}$	0.03	0.02	0.01	0.01	0.03	**0.06**	6.2	3.2	3.2	3.6	2.9	5.0	3.6	1.7	1.9	2.1	1.0	1.3	3.5	0.4
$Y_{2,7}$	0.10	0.05	0.04	0.04	0.09	0.03	**0.13**	14.4	12.8	11.9	4.2	12.5	14.9	5.2	8.1	11.2	6.0	3.2	6.3	3.7
$Y_{2,8}$	0.07	0.03	0.03	0.01	0.07	0.02	0.06	**0.11**	44.8	5.7	2.0	3.3	12.5	10.7	3.8	7.2	6.3	0.8	7.4	10.4
$Y_{2,9}$	0.07	0.02	0.03	0.01	0.06	0.01	0.05	0.06	**0.09**	6.5	2.7	3.4	13.7	9.9	9.3	10.0	7.2	1.0	7.1	27.3
$Y_{2,10}$	0.03	0.03	0.01	0.01	0.04	0.01	0.03	0.02	0.02	**0.04**	3.9	26.8	9.8	5.3	8.2	12.9	4.6	4.9	3.0	2.9
$Y_{2,11}$	0.03	0.02	0.01	0.01	0.03	0.01	0.02	0.03	0.03	0.01	**0.05**	1.9	5.8	5.7	2.0	4.6	3.4	1.9	2.5	14.6
$Y_{2,12}$	0.03	0.03	0.01	0.01	0.03	0.01	0.02	0.01	0.01	0.02	0.00	**0.04**	6.5	3.9	5.1	8.8	3.9	1.5	3.9	1.2
$Y_{2,13}$	0.03	0.02	0.02	0.01	0.03	0.01	0.03	0.02	0.02	0.01	0.01	0.01	**0.04**	4.7	14.8	41.4	3.3	2.3	2.0	9.6
$Y_{2,14}$	0.02	0.01	0.01	0.00	0.02	0.00	0.02	0.02	0.02	0.01	0.01	0.00	0.01	**0.04**	7.2	7.9	1.6	1.2	4.2	7.1
$Y_{2,15}$	0.01	0.01	0.01	0.00	0.01	0.00	0.01	0.01	0.01	0.01	0.00	0.00	0.01	0.00	**0.02**	26.8	2.0	2.0	1.7	5.8
$Y_{2,16}$	0.01	0.01	0.01	0.00	0.01	0.00	0.01	0.01	0.01	0.01	0.00	0.00	0.01	0.00	0.01	**0.02**	4.5	2.5	3.3	9.4
$Y_{2,17}$	0.01	0.01	0.00	0.00	0.01	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.02**	0.0	1.1	5.8
$Y_{2,18}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.01**	7.2	0.0
$Y_{2,19}$	0.01	0.00	0.00	0.00	0.01	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.01**	3.5
$Y_{2,20}$	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.01**

Table A2: Association structure for simulations in Table 3 for the second group ($i = 2$).

2 R code for computing simultaneous intervals

Here we show selected R code to compute many of the simultaneous confidence intervals mentioned in Section 2. First, load ("source") the necessary functions into R and also load the two provided datasets X (treatment group) and Y (placebo group). Note that due to confidentiality these are only a small subset of the original data and are not representative of the entire vaccine data set. You need to have the R package "mvtnorm" installed for computing multivariate normal probabilities.

2.1 Non-grid based methods

```
> source("http://sites.williams.edu/bklingen/files/2012/10/simCIRD.r")
> X <- read.table(file=
+ "http://sites.williams.edu/bklingen/files/2011/10/datagroup1.txt", header=TRUE)
> Y <- read.table(file=
+ "http://sites.williams.edu/bklingen/files/2011/10/datagroup2.txt", header=TRUE)
> head(X)
  HEADACHE MALAISE PYREXIA ARTHRALGIA
1        1       0       0          0
2        0       0       0          0
3        0       1       0          0
4        1       0       0          0
5        1       1       0          0
6        0       0       0          0
> head(Y)
  HEADACHE MALAISE PYREXIA ARTHRALGIA
1        0       0       0          0
2        0       0       0          0
3        0       0       0          0
4        1       0       0          1
5        1       1       0          0
6        0       0       0          0
> ##### Simultaneous CI based on the Local2 approach:
> simRD.score(X,Y) #Local2 is the default
Loading required package: mvtnorm
$CI
                    LB           UB
HEADACHE   -0.12725417  0.071634526
MALAISE    -0.02759339  0.132489353
PYREXIA    -0.11343504 -0.005667696
ARTHRALGIA -0.06612705  0.049281807

$conflev
[1] 0.95
```

```
$method
[1] "local2"

$c
[1] 2.473396

$bounds
[1] "two.sided"


> ##### Simple Bonferroni adjusted score intervals:
> simRD.score(X,Y, method="bonf")
$CI
                    LB           UB
HEADACHE   -0.12822607  0.072614487
MALAISE    -0.02834103  0.133318290
PYREXIA    -0.11407753 -0.005125528
ARTHRALGIA -0.06672213  0.049910639

$conflev
[1] 0.95

$method
[1] "bonf"

$c
[1] 2.497705

$bounds
[1] "two.sided"


> ##### Unadjusted score intervals:
> simRD.score(X,Y, method="unadj")
$CI
                    LB          UB
HEADACHE   -0.10671567  0.05096156
MALAISE    -0.01074193  0.11538731
PYREXIA    -0.10014456 -0.01671463
ARTHRALGIA -0.05300203  0.03643611

$conflev
[1] 0.95

$method
```

```
[1] "unadj"

$c
[1] 1.959964

$bounds
[1] "two.sided"


> ##### Adjusted Wald intervals:
> simRD.Wald(X,Y)
$CI
                     LB           UB
HEADACHE   -0.12748740  0.071931849
MALAISE    -0.02782028  0.130994880
PYREXIA    -0.10755008 -0.003561028
ARTHRALGIA -0.06343174  0.047558729

$conflev
[1] 0.95

$method
[1] "max"

$adj
[1] TRUE

$c
[1] 2.474043

$bounds
[1] "two.sided"

> ##### Adjusted Wald interval based on Bonferroni adjustment:
> simRD.Wald(X,Y, method="bonf")
$CI
                     LB           UB
HEADACHE   -0.12844106  0.072885505
MALAISE    -0.02857976  0.131754361
PYREXIA    -0.10804738 -0.003063735
ARTHRALGIA -0.06396252  0.048089504
...
> ##### Adjusted Wald interval based on no multiplicity adjustment:
> simRD.Wald(X,Y, method="unadj")
$CI
```

```
                    LB          UB
HEADACHE   -0.10676884  0.05121328
MALAISE    -0.01132026  0.11449486
PYREXIA    -0.09674619 -0.01436492
ARTHRALGIA -0.05190044  0.03602743
...
```

2.2 One-sided simultaneous confidence intervals

To request upper bounds, include `bounds="upper"`, for lower bounds `bounds="lower"`, e.g., for upper bounds with the Local2 approach:

```
> simRD.score(X,Y, bounds="upper")
$CI
           LB          UB
HEADACHE   -1  0.06070595
MALAISE    -1  0.12345433
PYREXIA    -1 -0.01158045
ARTHRALGIA -1  0.04242177

$conflev
[1] 0.95

$method
[1] "local2"

$c
[1] -2.202333

$bounds
[1] "upper"
```

2.3 Grid-based Methods

The following provides code and functionality for the methods that need a grid search to determine the simultaneous confidence intervals, such as the global score or the GEE approach. The reason a grid search is necessary is that the correlation matrix is estimated under the null, and so the critical value changes for each different value of $\boldsymbol{\delta}_0$. Prof. J. Lang (`http://homepage.stat.uiowa.edu/ jblang`) provides R code (mph.fit) for restricted ML estimation. Part of his code has been implemented in the R functions below. You will also need packages "inline" and "RcppArmadillo" (for calling C++ code from within R) installed. Finally, if you want to use parallel processing for a faster grid search, you can set up a cluster via the pre-installed "parallel" package. If you have only one CPU, replace "makeCluster(16)" with "makeCluster(1)" below. We ran the below code on a cluster running Ubuntu.

```
> ## Grid search methods using C++ code and parallel processing
> require("inline")
> require("RcppArmadillo")
> require("parallel")
> cl <- makeCluster(16)  #this uses 16 CPUs for the grid search
> ## For full score and Local1 approach
> #load the C++ code on each node
> clusterCall(cl, fun = function(z) source(file=
+ "http://sites.williams.edu/bklingen/files/2012/12/mph_fast.r"))
> clusterExport(cl,list("step.search", "pvalue.score", "partition.pvalue",
+ "score.z"))
> clusterEvalQ(cl,require(mvtnorm))
> #####Global Score simultaneous CIs:
> simRD.score(X,Y,method="global", cluster=cl)
Loading required package: mvtnorm
$CI
                    LB           UB
HEADACHE   -0.12941983  0.074362646
MALAISE    -0.02879446  0.134844549
PYREXIA    -0.11288478 -0.004281882
ARTHRALGIA -0.06660003  0.049068035

$conflev
[1] 0.95

$method
[1] "global"

$prec
    HEADACHE      MALAISE      PYREXIA   ARTHRALGIA
0.0004030230 0.0003209628 0.0002101602 0.0002243099

> #####Local1 approach with Malaise and Pyrexia in a group:
> simRD.score(X,Y,method="local1", Part=list(1,c(2,3),4), cluster=cl)
$CI
                    LB           UB
HEADACHE   -0.12879317  0.073148550
MALAISE    -0.02839548  0.133056770
PYREXIA    -0.11171417 -0.005155428
ARTHRALGIA -0.06535061  0.048725334

$conflev
[1] 0.95

$method
```

```
[1] "local1"

$Part
$Part[[1]]
[1] 1

$Part[[2]]
[1] 2 3

$Part[[3]]
[1] 4


$prec
    HEADACHE      MALAISE      PYREXIA   ARTHRALGIA
0.0006717051 0.0005349380 0.0003502669 0.0003738499

> ## For restricted GEE estimation:
> #load the C++ code on each node
> clusterCall(cl, fun = function(z) source(file=
+ "http://sites.williams.edu/bklingen/files/2013/01/GEE_fast.r"))
> clusterExport(cl,list("step.search", "pvalue.GEE", "score.z"))
> clusterEvalQ(cl,require(mvtnorm))
> #####GEE with unstructured correlation structure:
> simRD.score(X,Y,method="GEE.unstr", cluster=cl)
$CI
                    LB           UB
HEADACHE   -0.12744641  0.071456513
MALAISE    -0.02725165  0.131350981
PYREXIA    -0.10702082 -0.005904385
ARTHRALGIA -0.06254673  0.046536256

$conflev
[1] 0.95

$method
[1] "GEE.unstr"

$prec
    HEADACHE      MALAISE      PYREXIA   ARTHRALGIA
0.0004030230 0.0003209628 0.0002101602 0.0002243099

> #####GEE with independence correlation structure:
> simRD.score(X,Y,method="GEE.ind", cluster=cl)
$CI
```

```
                    LB           UB
HEADACHE   -0.12742370  0.071474391
MALAISE    -0.02723357  0.131214120
PYREXIA    -0.10697777 -0.005916228
ARTHRALGIA -0.06250564  0.046498207
...
> stopCluster(cl)
```