

# Annotated Analytical Code for Rozen et al., *AZFc* Deletions and Spermatogenic Failure...

July 6, 2012

## Contents

<b>1</b>	<b>Input data and sources</b>	<b>1</b>
1.1	From the current population study . . . . .	1
1.2	Prevalence of deletions in men with SSF . . . . .	2
1.2.1	Prevalence of b2/b4 deletion in men with SSF . . . . .	2
1.2.2	Prevalence of gr/gr deletion in men with SSF . . . . .	2
1.2.3	Prevalence of b1/b3 deletion in men with SSF . . . . .	2
1.3	Construct list of data vectors for re-sampling . . . . .	2
<b>2</b>	<b>Main text Table 2 – Prevalence (of deletions) in population</b>	<b>3</b>
<b>3</b>	<b>Main text Table 2 – Prevalence (of deletions) among men with SSF</b>	<b>3</b>
3.1	Deletion prevalence in all SSF . . . . .	3
3.2	Organize top of Table 2 . . . . .	4
<b>4</b>	<b>Function for one set of values for one type of deletion</b>	<b>4</b>
4.1	Inputs for calculations in Table 2 . . . . .	4
4.2	Outputs of calculations in Table 2 . . . . .	4
4.2.1	Estimating the prevalence of SSF in the population . . . . .	4
4.2.2	Specifics of outputs of calculations in Table 2 . . . . .	4
<b>5</b>	<b>Function for one set of values for all types of deletion</b>	<b>6</b>
<b>6</b>	<b>Use function <code>one.replicate</code> to get point estimates</b>	<b>7</b>
<b>7</b>	<b>Bootstrap calculations of confidence intervals</b>	<b>7</b>
<b>8</b>	<b>Format and print Table 2</b>	<b>9</b>
<b>9</b>	<b>Technical information</b>	<b>11</b>
9.1	How to create this PDF file . . . . .	11
9.2	Package versions . . . . .	11

## 1 Input data and sources

### 1.1 From the current population study

```
> pop.study.total      <- 20884
> pop.study.b2b4       <- 9
> pop.study.b1b3       <- 21
```

For analysis of gr/gr deletions we restrict our focus to Poland and the US.

```

> pop.study.total.US      <- 14668
> pop.study.total.Pol    <- 4445
> pop.study.grgr.US      <- 312
> pop.study.grgr.Pol     <- 115
> pop.study.total.eur    <- pop.study.total.US + pop.study.total.Pol
> pop.study.grgr.eur     <- pop.study.grgr.US + pop.study.grgr.Pol

```

## 1.2 Prevalence of deletions in men with SSF

### 1.2.1 Prevalence of b2/b4 deletion in men with SSF

Data for b2/b4 deletions is from Oates et al., 2002. The names of the variables should be self-explanatory.

```

> b2b4.azoo.n           <- 430
> b2b4.oligo.n          <- 283
> b2b4.ssf.n           <- b2b4.azoo.n + b2b4.oligo.n
> b2b4.azoo.del        <- 14 + 7 + 5
> b2b4.oligo.del       <- 16
> b2b4.ssf.del         <- b2b4.azoo.del + b2b4.oligo.del

```

### 1.2.2 Prevalence of gr/gr deletion in men with SSF

For gr/gr deletions among men with SSF (severe spermatogenic failure) broken out by azoospermia and oligospermia we use reports in the literature and new data that are summarized in Table S7.

```

> grgr.azoo.n          <- 2378
> grgr.oligo.n         <- 2307
> grgr.ssf.n          <- grgr.azoo.n + grgr.oligo.n
> grgr.azoo.del       <- 94
> grgr.oligo.del      <- 100
> grgr.ssf.del        <- grgr.azoo.del + grgr.oligo.del
>

```

### 1.2.3 Prevalence of b1/b3 deletion in men with SSF

For gr/gr deletions among men with SSF (severe spermatogenic failure) broken out by azoospermia and oligospermia we use reports in the literature and new data that are summarized in Table S7.

```

> b1b3.total.ssf <- 2049 + 1907 # From Table S7, last line
> b1b3.ssf.and.del <- 3 + 7    # From Table S7, last line

```

## 1.3 Construct list of data vectors for re-sampling

Create a list of vectors of zeros and ones from which we will re-sample. The 'H' variables are for the proportion of men with SSF with the deletion, for example, H.b2b4 is the proportion of men with SSF who have a b2/b4 deletion. The 'F' variables are for the proportion of the population with the deletion, for example, q.b2b4 is the proportion of men in the population with the b2/b4 deletion.

```

> data.vec <-
+ list(
+   H.b2b4=numeric(b2b4.ssf.n), # A vector of zeros. Ones will be
+                               # added later.
+   H.azoo.b2b4 =numeric(b2b4.azoo.n),
+   H.oligo.b2b4=numeric(b2b4.oligo.n),
+

```

```

+       q.b2b4=numeric(pop.study.total),
+
+       H.grgr=numeric(grgr.ssf.n),
+       H.azoo.grgr=numeric(grgr.azoo.n),
+       H.oligo.grgr=numeric(grgr.oligo.n),
+
+       # We use only US and Poland data from the population study in order
+       # to better match the mostly European data in Table S7.
+       q.grgr=numeric(pop.study.total.eur),
+
+       H.b1b3=numeric(b1b3.total.ssf),
+       q.b1b3=numeric(pop.study.total)
+     )
> data.vec[["H.b2b4"]][1:b2b4.ssf.del]      <- 1
> data.vec[["H.azoo.b2b4"]][1:b2b4.azoo.del] <- 1
> data.vec[["H.oligo.b2b4"]][1:b2b4.oligo.del] <- 1
> data.vec[["q.b2b4"]][1:pop.study.b2b4]    <- 1
> data.vec[["H.grgr"]][1:grgr.ssf.del]      <- 1
> data.vec[["H.azoo.grgr"]][1:grgr.azoo.del] <- 1
> data.vec[["H.oligo.grgr"]][1:grgr.oligo.del] <- 1
> data.vec[["q.grgr"]][1:pop.study.grgr.eur] <- 1
> data.vec[["H.b1b3"]][1:b1b3.ssf.and.del]  <- 1
> data.vec[["q.b1b3"]][1:pop.study.b1b3]    <- 1

```

## 2 Main text Table 2 – Prevalence (of deletions) in population

```

> prev.in.pop.line1.and.2 <-
+   list(b2b4=prop.test(pop.study.b2b4, pop.study.total),
+       grgr=prop.test(pop.study.grgr.eur, pop.study.total.eur),
+       b1b3=prop.test(pop.study.b1b3, pop.study.total)
+     )
> prev.in.pop.line1 <- c(pop.study.total, pop.study.total.eur, pop.study.total)
> prev.in.pop.line2 <- c(pop.study.b2b4, pop.study.grgr.eur, pop.study.b1b3)
> line1 <-
+   formatC(unlist(lapply(prev.in.pop.line1.and.2, function(x) x[["estimate"]])) * 100,
+           digits=2, drop0trailing=F)
> line2 <- lapply(prev.in.pop.line1.and.2, function(x) x[["conf.int"]])
> line2 <-
+   unlist(lapply(line2,
+                 function(x) paste(formatC(x * 100, digits=2), collapse=' to ')))

```

## 3 Main text Table 2 – Prevalence (of deletions) among men with SSF

### 3.1 Deletion prevalence in all SSF

```

> prev.in.SSF.n <- c(b2b4.ssf.n, grgr.ssf.n, b1b3.total.ssf)
> prev.in.SSF.del <- c(b2b4.ssf.del, grgr.ssf.del, b1b3.ssf.and.del)
> line3.and.4 <-
+   list(b2b4=prop.test(b2b4.ssf.del, b2b4.ssf.n),
+       grgr=prop.test(grgr.ssf.del, grgr.ssf.n),
+       b1b3=prop.test(b1b3.ssf.and.del, b1b3.total.ssf)
+     )

```

```

> line3 <-
+   prettyNum(unlist(lapply(line3.and.4, function(x) x[['estimate']])) * 100,
+               digits=2)
> line4 <- lapply(line3.and.4, function(x) x[['conf.int']])
> line4 <-
+   unlist(lapply(line4,
+                 function(x) paste(prettyNum(x * 100, digits=2), collapse=' to ')))

```

### 3.2 Organize top of Table 2

```

> t2.top <-
+   rbind(prev.in.pop.line1, prev.in.pop.line2,
+         line1, line2,
+         prev.in.SSF.n,
+         prev.in.SSF.del,
+         line3, line4
+       )
> colnames(t2.top) <- c('b2b4', 'grgr', 'b1b3')
> rownames(t2.top) <- c('Prev in pop number',
+                      'Prev in pop number w del',
+                      'Prev in pop \\%',
+                      '\\ \\ Prev in pop CI',
+                      'Prev in SSF number',
+                      'Prev in SSF number w del',
+                      'Prev in SSF (\\%)',
+                      '\\ \\ Prev in SSF CI'
+                    )

```

## 4 Function for one set of values for one type of deletion

### 4.1 Inputs for calculations in Table 2

For each deletion type,  $\Delta$ , we have the following as inputs.

$H$  The proportion of men with SSF who have  $\Delta$ .

$q$  The proportion of men in the population who have  $\Delta$ .

### 4.2 Outputs of calculations in Table 2

#### 4.2.1 Estimating the prevalence of SSF in the population

For many of the calculations in Table 2 we need  $F_{SSF}$ , the proportion of all men who have spermatogenic failure. We estimate  $F_{SSF}$  as  $q_{b2b4} / H_{b2b4}$ , where  $q_{b2b4}$  and  $H_{b2b4}$  are  $q$  and  $H$  for b2/b4 deletions.  $H_{b2b4} = 42/713$  from Oates et al., 2000 (see Section 1.2) and  $q_{b2b4} = 9/20884$  from the current study (see Section 1.1). The point estimate of  $F_{SSF}$  is then  $(\text{pop.study.b2b4} / \text{pop.study.total}) / (\text{b2b4.ssf.del} / \text{b2b4.ssf.n}) = 0.00731592196349906$ . We calculate confidence intervals below.

#### 4.2.2 Specifics of outputs of calculations in Table 2

We will estimate the following for each type of deletion,  $\Delta$ , using calculations described in the Supplementary Note.

$s$  Deficit in fitness of Y chromosomes with  $\Delta$  compared to Y chromosomes without  $\Delta$ .

$G$  The proportion of men with deletion  $\Delta$  who have SSF.

$\mu$  ( $\mu$ ) The rate per generation at which deletion  $\Delta$  arises by new mutation.

**ARpct** The “attributable risk percent”, i.e. the percent of the prevalence of SSF among men with  $\Delta$  that is due to  $\Delta$ .

**PARpct** The “population attributable risk percent”, i.e. the percent of the prevalence of SSF in the population (with and without  $\Delta$ ) that is due to  $\Delta$ .

The function `fn1` embodies the logic of the computations. `fn1` operates on a single deletion type (e.g. `b2/b4`, `gr/gr`, or `b1/b3`). Note that `fn1.names` is used in several places, not just in `fn1`.

```
> fn1.names <- c('G', 'mu', 'ARpct', 'PARpct', 's', 'P.u', 'RR')
> fn1 <-
+   function(F.SSF, # Prevalence of SSF in population.
+               # See comments below in function
+               # one.replicate for details on how
+               # F.SSF is calculated.
+
+           H.vec, # Vector of 1s and 0s from re-sampling
+               # the number of deletions (1s) among
+               # men with SSF.
+
+           q.vec # Vector of 1s and 0s from re-sampling
+               # the number of deletions in the population
+               # (from the current study).
+           ) {
+
+   H <- sum(H.vec) / length(H.vec)
+   q <- sum(q.vec) / length(q.vec)
+
+   # F.del.SSF is the proportion of _all_ men
+   # who have the deletion and have SSF.
+   # This is H (the proportion of men
+   # with SSF who have the deletion) times
+   # F.SSF, the proportion of men in the population
+   # who have SSF.
+   F.del.SSF <- H * F.SSF
+
+   # G is the proportion of men with the deletion who
+   # have SSF.
+   #
+   #  $G = (H * F.SSF) / q$ , in which
+   # the numerator is the proportion of the entire population
+   # who have the deletion (F.del.SSF) AND hav SSF
+   G <- F.del.SSF / q
+
+   F.notdel.SSF <- F.SSF - F.del.SSF
+   F.notdel     <- 1 - q
+
+   # P.u is the prevalence of SSF among the
+   # 'unexposed' (those without the deletion)
+   P.u         <- F.notdel.SSF / F.notdel
+
+   RR <- G/P.u
+
+   # Assuming that all loss of fitness is
+   # due to SSF, s is the loss of fitness due to
+   # the deletion in question, relative to
+   # the fitness of undeleted Y chromosomes.
```

```

+   s <- (G - P.u) / (1 - P.u)
+
+   # From Repping et al., 2003 we have
+   # mu = s * q.
+   mu <- s * q
+
+   ARpct      <- 100 * (G - P.u) / G
+   PARpct     <- 100 * (F.SSF - P.u) / F.SSF
+
+   ret <- c(G, mu, ARpct, PARpct, s, P.u, RR)
+   names(ret) <- fn1.names
+   ret
+ }

```

## 5 Function for one set of values for all types of deletion

```

> one.replicate <-
+   function(
+     vec, show.f.ssf=F
+   ) {
+
+     H.b2b4 <- # Prevalence of b2b4 deletions in men with SSF
+               # (from Oates et al., 2002).
+               sum(vec[["H.b2b4"]]) / length(vec[["H.b2b4"]])
+
+     q.b2b4 <- # Prevalence of b2b4 deletions in unselected
+               # population (from current study).
+               sum(vec[["q.b2b4"]]) / length(vec[["q.b2b4"]])
+
+     # F.SSF is the proportion of the population with SSF.
+     # It is calculated as the proportion of men in
+     # the population who have a b2/b4 deletion (q.b2b4)
+     # over the proportion of men with SSF who have a b2/b4
+     # deletion (H.b2b4). (This will be calculated after
+     # bootstrap re-sampling.)
+     F.SSF <- q.b2b4 / H.b2b4
+
+     b1b3 <- fn1(F.SSF, vec[["H.b1b3"]], vec[["q.b1b3"]])
+     names(b1b3) <- paste(names(b1b3), 'b1b3', sep='.')
+
+     grgr <- fn1(F.SSF, vec[["H.grgr"]], vec[["q.grgr"]])
+     names(grgr) <- paste(names(grgr), 'grgr', sep='.')
+
+     b2b4 <- fn1(F.SSF, vec[["H.b2b4"]], vec[["q.b2b4"]])
+     names(b2b4) <- paste(names(b2b4), 'b2b4', sep='.')
+
+     out1 <- c(b2b4, grgr, b1b3 )
+
+     if (show.f.ssf) {
+       c(out1, f.ssf=F.SSF) }
+     else
+       { out1 }
+   }

```

## 6 Use function `one.replicate` to get point estimates

This is really a test of `one.replicate`. Compare with the values in main text Table 2.

```
> m1 <- matrix(one.replicate(data.vec), ncol=3)
> colnames(m1) <- c('b2b4', 'grgr', 'b1b3')
>                               # 'b2b3'
> rownames(m1) <- fn1.names
> # Checking:
> m1
```

	b2b4	grgr	b1b3
G	1.000000e+00	1.356008e-02	1.839108e-02
mu	4.309519e-04	1.437182e-04	1.122991e-05
ARpct	9.931121e+01	4.710034e+01	6.028088e+01
PARpct	5.850029e+00	1.950366e+00	1.523784e-01
s	1.000000e+00	6.432989e-03	1.116788e-02
P.u	6.887938e-03	7.173235e-03	7.304774e-03
RR	1.451813e+02	1.890371e+00	2.517679e+00

## 7 Bootstrap calculations of confidence intervals

```
> def.n.iter <- 20000
> fn.boot <-
+ function(n.iter = def.n.iter) {
+   out <- matrix(nrow=ncol(m1) * length(fn1.names),
+                 ncol=n.iter)
+
+   out.f.ssf <- numeric(n.iter)
+
+   for (i in 1:n.iter) {
+
+     tmp.vec <- list()
+     ldv <- length(data.vec)
+     for (jj in 1:ldv) {
+       nm <- names(data.vec)[jj]
+       tmp.vec[[nm]] <- sample(data.vec[[jj]], replace=T)
+     }
+
+     tmp <- one.replicate(tmp.vec, show.f.ssf=T)
+     len <- length(tmp)
+     out[,i] <- tmp[1:(len-1)]
+     out.f.ssf[i] <- tmp[len]
+   }
+
+   rownames(out) <- names(tmp[-len])
+
+   smry <-
+   function(x) {
+     tmp <- quantile(x, probs=c(0.05, 0.95), na.rm=T)
+     list(mean=mean(x, na.rm=T), med=median(x, na.rm=T),
+          five=tmp[1], ninety.five=tmp[2])
+   }
+
+   list(out=apply(out, 1, smry),
```

```
+      f.ssf=smry(out.f.ssf)
+    }
```

```
> set.seed(489)
> boot.out <- fn.boot(100000)
```

Show the bootstrap confidence intervals for  $F_{SSF}$  (F.SSF)

```
> boot.out[['f.ssf']]
```

```
$mean
[1] 0.007472643
```

```
$med
[1] 0.007187572
```

```
$five
      5%
0.003414097
```

```
$ninety.five
      95%
0.0124149
```

Next we look at the other outputs (for the deletions) as a sanity check:

```
> b1 <- boot.out[['out']]
> b2 <-
+ lapply(b1, function (x) {
+   m <- x[['med']]
+   q1 <- x[['five']]
+   q2 <- x[['ninety.five']]
+   strings <-
+     formatC(c(m, q1, q2), format='e', digits=2)
+   paste(strings[1], '(', strings[2], ',', strings[3], ')', sep='')
+ }
+ )
> b2 # as sanity check
```

```
$G.b2b4
[1] "1.00e+00(1.00e+00,1.00e+00)"
```

```
$mu.b2b4
[1] "4.31e-04(1.92e-04,6.70e-04)"
```

```
$ARpct.b2b4
[1] "9.93e+01(9.88e+01,9.97e+01)"
```

```
$PARpct.b2b4
[1] "5.85e+00(4.45e+00,7.38e+00)"
```

```
$s.b2b4
[1] "1.00e+00(1.00e+00,1.00e+00)"
```

```
$P.u.b2b4
[1] "6.79e-03(3.22e-03,1.18e-02)"
```

```
$RR.b2b4
```



```

[1] "1.47e+02(8.47e+01,3.10e+02)"

$G.grgr
[1] "1.33e-02(6.28e-03,2.35e-02)"

$mu.grgr
[1] "1.39e-04(6.31e-05,2.58e-04)"

$ARpct.grgr
[1] "4.71e+01(3.86e+01,5.41e+01)"

$PARpct.grgr
[1] "1.95e+00(1.43e+00,2.48e+00)"

$s.grgr
[1] "6.23e-03(2.79e-03,1.18e-02)"

$P.u.grgr
[1] "7.05e-03(3.36e-03,1.22e-02)"

$RR.grgr
[1] "1.89e+00(1.63e+00,2.18e+00)"

$G.b1b3
[1] "1.75e-02(6.29e-03,4.10e-02)"

$mu.b1b3
[1] "1.01e-05(1.37e-06,2.64e-05)"

$ARpct.b1b3
[1] "6.01e+01(1.67e+01,7.84e+01)"

$PARpct.b1b3
[1] "1.49e-01(2.24e-02,2.95e-01)"

$s.b1b3
[1] "1.03e-02(1.24e-03,3.09e-02)"

$P.u.b1b3
[1] "7.18e-03(3.41e-03,1.24e-02)"

$RR.b1b3
[1] "2.50e+00(1.20e+00,4.63e+00)"

> rm(b2)

```

## 8 Format and print Table 2

Format the confidence intervals.

```

> format1 <-
+ function (x, factor=1) {
+   q1 <- x[['five']]
+   q2 <- x[['ninety.five']]
+   strings <- prettyNum(c(q1, q2) * factor, digits=2)
+   paste(strings, collapse=' to ')

```

```

+   }
> CI.s <-
+   lapply(b1[c('s.b2b4', 's.grgr', 's.b1b3')],
+         function (x) {
+           q1 <- x[['five']]
+           q2 <- x[['ninety.five']]
+           # Leave in scientific notation:
+           strings <- formatC(c(q1, q2), format='e', digits=1)
+           paste(strings, collapse=' to ')
+         }
+       )
> CI.mu <-
+   lapply(b1[c('mu.b2b4', 'mu.grgr', 'mu.b1b3')],
+         function (x) {
+           q1 <- x[['five']]
+           q2 <- x[['ninety.five']]
+           # Leave in scientific notation:
+           strings <- formatC(c(q1, q2), format='e', digits=1)
+           paste(strings, collapse=' to ')
+         }
+       )
> CI.G <-
+   lapply(b1[c('G.b2b4', 'G.grgr', 'G.b1b3')],
+         function (x) format1(x, 100)
+       )
> CI.PARpct <-
+   lapply(b1[c('PARpct.b2b4', 'PARpct.grgr', 'PARpct.b1b3')],
+         format1)
> CI.ARpct <-
+   lapply(b1[c('ARpct.b2b4', 'ARpct.grgr', 'ARpct.b1b3')],
+         format1)
> CI.RR <-
+   lapply(b1[c('RR.b2b4', 'RR.grgr', 'RR.b1b3')],
+         format1)
> b3.m <- rbind('s CI'      =unlist(CI.s),
+              'mu CI'     =unlist(CI.mu),
+              'G CI'      =unlist(CI.G),
+              'PARpct CI' =unlist(CI.PARpct),
+              'ARpct CI'  =unlist(CI.ARpct),
+              'RR CI'     =unlist(CI.RR))
> colnames(b3.m) <- colnames(m1)
> b3.m['G CI', 'b2b4'] <- '(assumed by model)'
> # Model posits that everyone with the b2b4 deletions has SSF.
> # This is used in calculating the prevalence of SSF.

```

Format the main table

```

> t2.cols <- c('b2b4', 'grgr', 'b1b3')
> t2.bottom <-
+   rbind(
+     'RR'          =prettyNum(m1['RR', t2.cols], digits=2), # new
+     '\\ \\ RR CI' =b3.m['RR CI', t2.cols], # new
+
+     '\\% men w/ del who have SSF ($100 G$)'=prettyNum(m1['G', t2.cols]*100,digits=2),
+     '\\ \\ CI'    =b3.m['G CI', t2.cols],
+
+     'AR (\\%)'    =prettyNum(m1['ARpct', t2.cols], digits=3),

```

```

+   '\\ \\ AR CI' =b3.m[ARpct CI', t2.cols] ,
+
+   'PAR (\\%)'   =prettyNum(m1[PARpct', t2.cols], digits=3),
+   '\\ \\ PAR CI' =b3.m[PARpct CI', t2.cols],
+
+   'mu'          =formatC(m1[mu', t2.cols], digits=2, format='e'),
+   '\\ \\ mu CI' =b3.m[mu CI', t2.cols],
+
+   's'           =formatC(m1[s', t2.cols], digits=2, format='e'),
+   '\\ \\ s CI'  =b3.m[s CI', t2.cols]
+ )
> table2 <-
+   rbind(t2.top, t2.bottom)
> colnames(table2) = c('b2/b4', 'gr/gr', 'b1/b3')
> # For mu for b2/b4 the main text Table 2 uses the prevalence
> # information directly:
> table2[mu',          'b2/b4'] <- ''
> table2['\\ \\ mu CI', 'b2/b4'] <- ''

> require(xtable)
> print(
+   xtable(table2,
+           align=c('l', rep('c', ncol(table2))),
+           caption='Draft of main text Table 2'
+           ),
+   table.placement='t',
+   caption.placement='bottom',
+   sanitize.rownames.function = function(x) { x },
+   hline.after=c(-1,0,2,4,6,8,10,12,14,16,18,20)
+ )

```

## 9 Technical information

### 9.1 How to create this PDF file

Within R execute:

```
> Sweave('Rozen-et-al-code.Rnw', stylepath = TRUE, keep.source=T)
```

This should create the file Rozen-et-al-code.tex. To create the PDF from this file, in the shell, run

```
pdflatex Rozen-et-al-code.tex
```

This should create the file Rozen-et-al-code.pdf.

### 9.2 Package versions

```
> toLatex(sessionInfo())
```

- R version 2.15.0 (2012-03-30), x86\_64-unknown-linux-gnu
- Locale: LC\_CTYPE=en\_US.UTF-8, LC\_NUMERIC=C, LC\_TIME=en\_US.UTF-8, LC\_COLLATE=en\_US.UTF-8, LC\_MONETARY=en\_US.UTF-8, LC\_MESSAGES=en\_US.UTF-8, LC\_PAPER=C, LC\_NAME=C, LC\_ADDRESS=C, LC\_TELEPHONE=C, LC\_MEASUREMENT=en\_US.UTF-8, LC\_IDENTIFICATION=C
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils

	b2/b4	gr/gr	b1/b3
Prev in pop number	20884	19113	20884
Prev in pop number w del	9	427	21
Prev in pop %	0.043	2.2	0.1
Prev in pop CI	0.021 to 0.085	2 to 2.5	0.064 to 0.16
Prev in SSF number	713	4685	3956
Prev in SSF number w del	42	194	10
Prev in SSF (%)	5.9	4.1	0.25
Prev in SSF CI	4.3 to 7.9	3.6 to 4.8	0.13 to 0.48
RR	145	1.9	2.5
RR CI	85 to 310	1.6 to 2.2	1.2 to 4.6
% men w/ del who have SSF (100 <i>G</i> )	100	1.4	1.8
CI	(assumed by model)	0.63 to 2.3	0.63 to 4.1
AR (%)	99.3	47.1	60.3
AR CI	99 to 100	39 to 54	17 to 78
PAR (%)	5.85	1.95	0.152
PAR CI	4.4 to 7.4	1.4 to 2.5	0.022 to 0.29
mu	Same as	1.44e-04	1.12e-05
mu CI	prevalence	6.3e-05 to 2.6e-04	1.4e-06 to 2.6e-05
s	1.00e+00	6.43e-03	1.12e-02
s CI	1.0e+00 to 1.0e+00	2.8e-03 to 1.2e-02	1.2e-03 to 3.1e-02

Table 1: Draft of main text Table 2

- Other packages: xtable 1.7-0
- Loaded via a namespace (and not attached): compiler 2.15.0, tools 2.15.0