1 Mathematical details of the methods

A Method by Pepe

The method constructs a dataset with genotypes and disease status for a hypothetical population. The method requires risk allele frequencies and odds ratios (ORs) of the genetic variants, population disease risk, and number of individuals as input parameters. The dataset is constructed using a simulation procedure that involves three steps:

1. **Modelling genotype data:** For each variant, the distribution of the three genotypes in the hypothetical population is based on genotype frequencies, which are calculated from the risk allele frequencies assuming Hardy-Weinberg Equilibrium. Genotypes are randomly distributed over individuals.

2. **Modelling individual disease risks:** Step 3 requires disease risks to assign disease status to all individuals. Disease risks are calculated from the logistic regression equation:

   $$\text{Logit}(\text{risk}_i) = \alpha_0 + \sum_{g=1}^{G} \alpha_g K_{gi},$$

   where

   risk$_i$ is disease risk for individual i,

   α_0 is log(odds of disease for those who carry zero risk alleles),

   $\alpha_g = \log(OR_g)$, with OR_g being the OR of the risk allele of variant g,

   K_{gi} is the number or risk alleles $(0, 1$ or $2)$ for each variant g in individual i.

   When α_g and K_{gi} are known, α_0 is obtained by solving the logistic regression equation such that the average risk in the hypothetical population is equal to the specified population disease risk. Note that the above equation implies that the method explicitly uses weighted risk scores ($\sum_{g=1}^{G} \alpha_g K_{gi}$), defined as the sum of number of risk alleles, each weighted by their corresponding $\log(OR)$.

3. **Modelling disease status:** Disease status (0 or 1) is assigned to each individual with the probability of developing disease (1) being equal to risk$_i$.

The area under the receiver-operating characteristic curve (AUC) is obtained using the method of Hanley and McNeill (1982).

B Method by Janssens

Like the method of Pepe, this method constructs a dataset with genotypes and disease status for a hypothetical population. The method requires frequencies and ORs of the genetic variants, population disease risk, and number of individuals as input parameters. There are two differences between this method and that of Pepe. The method of Pepe requires per allele ORs and frequencies, where this method can handle per allele, per genotype and dominant/recessive effects of the risk alleles. And the method of Pepe estimates disease risks using the logistic regression equation, where this method follows Bayes' theorem. The steps are similar to the method of Pepe and include:.

1. **Modelling genotype data:** For each variant, the distribution of the three genotypes is based on genotype frequencies, which can be directly specified as input parameters or be calculated from risk allele frequencies assuming Hardy-Weinberg Equilibrium. Genotypes are randomly distributed over individuals.

2. **Modelling individual disease risks:** Also this method requires disease risks in step 3 to assign disease status to all individuals. Disease risks are calculated from the posterior (disease) odds using Bayes' theorem:

$$\text{(posterior odds)}_i = \text{(prior odds)} \times \prod_{g=1}^{G} LR_{gji}$$

   where

   prior odds are given by $\frac{d}{1-d}$, with d the population disease risk,

   LR_{gji} is the likelihood ratio for genotype j of variant g in individual i.

   LRs are calculated from the frequencies and ORs of the genetic variants and the population disease risk. Note that this method also considers different effect sizes for genetic variants, which is similar to calculating weighted risk scores.

3. **Modelling disease status:** The third step is the same as in the method of Pepe. Disease status (0 or 1) is assigned to each individual with the probability of developing disease (1) being equal to risk$_i$.

The area under the receiver-operating characteristic curve (AUC) is obtained using the method of Hanley and McNeill (1982).

C Method by Lu

The method estimates the AUC using genotype frequencies and relative risks (RRs) or ORs of the genetic variants, and population disease risk as input parameters. The method involves following six steps.

1. **Obtaining genotype frequency in cases:** When effect sizes are specified as RRs, the method calculates the frequency of genotypes in those who will develop the disease (D) using the equation

   $$p(K_{gj}|D) = \frac{rr_{gj}\, f_{gj}}{\sum_{j=i}^{3} rr_{gj}\, f_{gj}},$$

   where

   rr_{gj} is the RR for genotype j of variant g,

   f_{gj} is the frequency for variant j in genotype g in the population,

   K_{gj} is the number of risk alleles $(0,1,2)$ in genotype j of variant g with $j = 1,2,3$.

   When effect sizes are specified as ORs, the equation is

   $$p(K_{gj}|D) = \frac{p(D|K_{gj}) f_{gj}}{d},$$

   where

   $p(D|K_{gj})$ is the probability of having the disease given genotype,

   d is the population disease risk.

   If the odds ratio for genotype j of variant g is defined as OR_{gj}, then $p(D|K_{gj})$ for any variant g are obtained by solving the equations

   $$p(D|K_{gj}) = \frac{OR_{gj}\, p(D|K_{g1})}{1 + OR_{gj}\, p(D|K_{g1}) - p(D|K_{g1})},$$

   and

   $$\sum_{j=1}^{3} p(D|K_{gj}) f_{gj} = d$$

   where the odds ratio for the reference genotype, $OR_{g1} = 1$. Note that the method considers different effect sizes for genetic variants and therefore implicitly uses weighted risk scores.

2. **Frequencies of genotype combinations in the population:** The frequencies of all genotype combinations ($X_n = (K_{1j}, \ldots, K_{Gj})$) in the population are calculated as

$$p(X_n) = \prod_{g=1}^{G} f_{gj},$$

with $n = 1, \ldots, 3^G$.

3. **Frequencies of genotype combinations conditional on disease status:** The frequencies of the genotype combinations for those who do (D) and those who will not develop the disease ($\bar{D}$) are calculated using the equations:

$$p(X_n|D) = \prod_{g=1}^{G} p(K_{gj}|D)$$

and

$$p(X_n|\bar{D}) = \frac{p(X_n) - p(X_n|D)d}{1-d}.$$

4. **Likelihood ratios (LRs) of genotype combinations:** The likelihood ratios (LRs) are calculated as

$$LR_n = \frac{p(X_n|D)}{p(X_n|\bar{D})}.$$

5. **Obtaining true and false positive rates:** The method then arranges $p(X_n|D)$ and $p(X_n|\bar{D})$ in descending order of their LR_n and calculates the true positive rate (TPR) and false positive rate (FPR) for all cutoff values defined by LR_n using the equations:

$$TPR_n = \sum_{(n)=1}^{n} p(X_{(n)}|D)$$

and

$$FPR_n = \sum_{(n)=1}^{n} p(X_{(n)}|\bar{D}),$$

where (n) is the n^{th} genotype combination in the sequence of descending order of likelihood ratios.

6. **Obtaining AUC:** The AUC is obtained using the trapezoidal rule:

$$AUC = \frac{1}{2}\sum_{n=1}^{3^G} (TPR_n + TPR_{n-1}) \times (FPR_n - FPR_{n-1})\,.$$

D Method by Moonesinghe

The method estimates the AUC using frequencies and RRs for dominant or recessive effects of the genetic variants as input parameters. The method involves two steps.

1. **Constructing distributions of genotype combinations:** The distributions of genotype combinations in those who will develop the disease and in those who will not are approximated by normal distributions when the number of variants is large. When frequencies and RRs are identical for all variants, $f_g = f$ and $rr_g = rr$. Under this assumption the mean and variance of the normal distribution are Gf^* and $Gf^*(1-f^*)$ in those who will develop the disease, and Gf and $Gf(1-f)$ for those who will not develop the disease, with $f^* = \frac{rr\,f}{rr\,f+(1-f)}$.

   Note that the method considers different effect sizes for genetic variants and therefore implicitly uses weighted risk score.

2. **Calculating AUC:** AUC is obtained from the equation:

   $$\Phi\left[\frac{G(f^*-f)}{\sqrt{Gf^*(1-f^*)+Gf(1-f)}}\right]$$

   when RRs and frequencies are identical for all variants, or

   $$\Phi\left[\sqrt{\sum_{g=1}^{G}\frac{(f_g^*-f_g)^2}{f_g^*(1-f_g^*)+f_g(1-f_g)}}\,\right]$$

   when RRs and frequencies differ between variants,

   where

   > Φ is the cumulative distribution function of a standard normal distribution,
   >
   > f_g^* is calculated in the same way as f^*.

E Method by Gail

The method estimates the AUC using risk allele frequencies and ORs of genetic variants as input parameters. The method involves following four steps:

1. **Obtaining relative risks for genotype combinations:** Assuming that the disease is rare, the method computes RRs for all genotype combinations ($X_n = (K_{1j}, \ldots, K_{Gj})$) using the equation:

   $$rr(X_n) = \prod_{g=1}^{G} (OR_g)^{K_{gj}},$$

   where $n = 1, \ldots, 3^G$ and

   K_{gj} is the number of risk alleles $(0, 1, 2)$ in genotype j of variant g,

   OR_g is the OR of the risk allele for variant g.

   Note that the above equation implies that the method explicitly uses weighted risk scores, as $\prod_{g=1}^{G}(OR_g)^{K_{gj}}$ can be rewritten in the logarithmic scale as $\sum_{g=1}^{G} \log(OR_g) K_{gi}$. This formula defines the sum of number of risk alleles weighted by their corresponding $\log(OR)$. This formula is similar to that used by Pepe.

2. **Obtaining distributions of relative risks:** The cumulative distributions of RRs in the total population ($F_{rr}(t)$) and in those who will develop the disease ($FD_{rr}(t)$) are obtained from the distributions of the genotype combinations, and calculated using the equations:

   $$F_{rr}(t) = \sum_{X_n : rr(X_n) \leq t} p(X_n)$$

   and

   $$FD_{rr}(t) = \frac{\sum_{X_n : rr(X_n) \leq t} rr(X_n)\, p(X_n)}{\sum_{all\, X_n} rr(X_n)\, p(X_n)}$$

   where

   t is any threshold value (here of RR),

$p(X_n) = \prod_{g=1}^{G} f_{gj}$, with f_{gj} the frequency of genotype j in variant g. Genotype frequencies are calculated from risk allele frequencies assuming Hardy-Weinberg Equilibrium.

3. **Calculation of absolute risks:** Absolute risks are calculated by multiplying the RRs of the genotype combinations with the proportional constant k, which is the risk of disease for those who carry zero risk alleles. The method calculates the cumulative distributions of absolute risks in the population ($F_r(t)$) and in those who will develop the disease ($FD_r(t)$) across all threshold values (here of absolute risk) by:

   $$F_r(t) = F_{rr}(t/k)$$

   and

   $$FD_r(t) = FD_{rr}(t/k).$$

4. **Calculating AUC:** A curve is constructed that plots $[1 - FD_r(t)]$ against $[1 - F_r(t)]$ across all values of the risk threshold t. The area under this curve is similar to the AUC under the assumption that the disease is rare.

2 Source codes of the methods

We compared the five most common methods that estimate the AUC of prediction models on the basis of epidemiological parameters. All analyses were programmed in R. The R source codes for three methods (Pepe, Janssens and Lu) were available online and for the two others, we programmed our own implementations based on the equations from the papers.

A Method of Pepe

Source code:

The source code and documentation are available from `http://labs.fhcrc.org/pepe/dabs/mgrp-Main.html` [accessed 16 December 2011].

Reference:

Pepe MS, Gu W, Morris DE (2010). *The Potential of Genes and Other Markers to Inform about Risk.* **Cancer Epidemiology, Biomarkers and Prevention** 19(3):655-665.

B Method of Janssens

Source code:

The source code (simulatedDataset) is included in the R package PredictABEL, which together with the documentation is available from `http://www.genabel.org/packages/PredictABEL` [accessed 16 December 2011].

Reference:

Janssens AC, Aulchenko YS, Elefante S, Borsboom GJ, Steyerberg EW, van Duijn CM. *Predictive testing for complex diseases using multiple genes: fact or fiction?* **Genet Med.** 2006;8:395-400.

Kundu S, Aulchenko YS, van Duijn CM, Janssens AC. *PredictABEL: an R package for the assessment of risk prediction models.* **Eur J Epidemiol.** 2011;26:261-4.

C Method of Lu

Source code:

The source code is available from http://darwin.cwru.edu/~qlu/ [accessed 16 December 2011].

Reference:

Lu Q, Elston RC. *Using the optimal receiver operating characteristic curve to design a predictive genetic test, exemplified with type 2 diabetes.* **Am J Hum Genet.** 2008;82:641-51.

D Method of Moonesinghe

Source code when odds ratio (OR) and frequency are the same for all variants considered.

Arguments

`Pg`	Frequency of risk alleles
`d`	Population disease risk (prevalence)
`ORg`	Odds ratio (OR) of risk alleles
`g`	Number of genetic variants included

Value

`AUCDom`	Estimate AUC assuming dominant effects of the risk alleles
`AUCRec`	Estimate AUC assuming recessive effects of the risk alleles

Example

```
AUCMoonesinghe (Pg=.25, d=.25, ORg=2, g=50)
```

Function

```
AUCMoonesinghe <- function (Pg,d,ORg,g)
{
  #============================================================
  # Reconstruct 2*3 table from allelic OR and p to obtain data
  # for calculating dominant/recessive RR and frequencies
  #============================================================
  reconstruct.2x3tableHWE <- function (OR,p,d){
    OR1 <- OR
    OR2 <- OR^2
    p1  <- 2*p*(1-p)
    p2  <- p*p
    a   <- .00001
    eOR <- 0
    while (eOR<=OR2){
      b     <- p2*(1-d)
      snew  <- 1-a-b
      p1new <- p1/(1-p2)
      dnew  <- (d-(a)) / ((d-(a))+ ((1-d)-b))
      c     <- (OR1*p1new*snew*(1-dnew)*dnew*snew) / ((1-p1new)*snew*(1-
          dnew)+OR1*p1new*snew*(1-dnew))
      dd    <- p1new * ((1-d)-b)
      e     <- (d-a)-c
      f     <- ((1-d)-b) - dd
      eOR   <- (a*f) / (b*e)
      tabel <- cbind(a, b, c, dd, e, f, OR1, OR2)
      a     <- a + .00001
      tabel }
    tabel }
  tab3x2 <- reconstruct.2x3tableHWE(OR=ORg, p=Pg, d=d)
  FreqDom <- (tab3x2[1] + tab3x2[2] + tab3x2[3] + tab3x2[4])
  FreqRec <- (tab3x2[1] + tab3x2[2])
  RRDom  <- ((tab3x2[1] + tab3x2[3]) / (tab3x2[1] + tab3x2[2] + tab3x2[3]
      + tab3x2[4])) / (tab3x2[5]/(tab3x2[5] + tab3x2[6]))
  RRRec  <- (tab3x2[1] / (tab3x2[1] + tab3x2[2])) / ((tab3x2[3] + tab3x2
      [5]) / (tab3x2[3] + tab3x2[4] + tab3x2[5] + tab3x2[6]))
  #============================================================
  # Obtain AUC by Moonesinghe's formula
  #============================================================
  GstarDom <- (FreqDom*RRDom) / ( (FreqDom*RRDom)+(1-FreqDom) )
  GstarRec <- (FreqRec*RRRec) / ( (FreqRec*RRRec)+(1-FreqRec) )
  xDom <- (g * (GstarDom-FreqDom)) / sqrt((g*GstarDom*(1-GstarDom)) + (g*
      FreqDom*(1-FreqDom)))
  xRec <- (g *(GstarRec-FreqRec)) / sqrt((g*GstarRec*(1-GstarRec)) + (g*
      FreqRec*(1-FreqRec)))
  AUCDom <- pnorm(xDom)
  AUCRec <- pnorm(xRec)
  list(AUCDom=AUCDom, AUCRec=AUCRec)
}
```

Source code when odds ratios and frequencies differ between variants.

Arguments

`ORg` Matrix with ORs of the genetic variants. The matrix contains two columns and the number of rows is same as the number of genetic variants considered. Genetic variants can be specified as per genotype, per allele, or as dominant/recessive effect of the risk allele. When per genotype data are used, OR of the heterozygous and homozygous risk genotypes are mentioned in the first and second columns. When per allele data are used, the ORs of the risk allele are specified in the first column and the second column is coded as 1. When dominant/recessive effects of the risk alleles are used, the OR of the dominant/recessive variant are specified in the first column, and the second column is coded as 0.

`Pg` Matrix with frequencies of the genetic variants. The matrix contains two columns and the number of rows is same as the number of genetic variants considered. Like `ORg`, the frequencies can be specified as per genotype, per allele, or as dominant/ recessive effect of the risk allele, and the corresponding coding is same as indicated in `ORg`.

`d` Population disease risk (prevalence)

Value

`AUCDom` Estimate AUC assuming dominant effects of the risk alleles

`AUCRec` Estimate AUC assuming recessive effects of the risk alleles

Example

```
# specify the matrix containing the ORs of genetic variants.
# In this example per allele effects of the risk variants are used
OR<-cbind(c(1.35,1.20,1.24,1.16), rep(1,4))

# specify the matrix containing the frequencies of genetic variants
p<-cbind(c(.41,.29,.28,.51),rep(1,4))
```

```
# Obtain the AUC
AUCMoonesinghe(Pg=p,d=.10,ORg=OR)
```

Function

```
AUCMoonesinghe <- function (ORg,Pg,d)
{
  #==========================================================
  # Reconstruct 2*2 table from dominant/recessive ORs and
  # frequencies to obtain data for calculating
  # dominant/recessive RR
  #==========================================================
  reconstruct.2x2table <- function(p,d,OR)
    {
      a  <- 0
      b  <- 0
      c  <- (OR*p*(1-d)*d)/((1-p)*(1-d)+OR*p*(1-d))
      dd <- p-c
      e  <- d-c
      f  <- (1-p)-e
      tabel <- cbind(a,b,c,dd,e,f,OR)
      tabel
    }

  #=============================================================
  # Reconstruct 2*3 table from genotypic OR and p to obtain
  # data for calculating dominant/recessive RR and frequencies
  #=============================================================
  reconstruct.2x3table <- function(OR1,OR2,p1,p2,d){
    a   <- .00001
    eOR <- 0
    while (eOR<=OR2){
      b    <- p2*(1-d)
      snew <- 1-a-b
      p1new <-p1/(1-p2)
      dnew  <- (d-(a)) / ((d-(a))+ ((1-d)-b))
      c     <- (OR1*p1new*snew*(1-dnew)*dnew*snew) / ((1-p1new)*snew*(1-
          dnew)+OR1*p1new*snew*(1-dnew))
      dd    <- p1new * ((1-d)-b)
      e     <- (d-a)-c
      f     <- ((1-d)-b) - dd
      eOR   <- (a*f) / (b*e)
      tabel <- cbind(a, b, c, dd, e, f, OR1, OR2)
      a     <- a + .00001
      tabel
    }
    tabel
  }

  #============================================================
  # Reconstruct 2*3 table from allelic OR and p to obtain data
  # for calculating dominant/recessive RR and frequencies
  #============================================================
  reconstruct.2x3tableHWE <- function (OR,p,d){
    OR1 <- OR
    OR2 <- OR^2
    p1  <- 2*p*(1-p)
    p2  <- p*p
    a   <- .00001
    eOR <- 0
    while (eOR<=OR2){
      b    <- p2*(1-d)
      snew <- 1-a-b
      p1new <-p1/(1-p2)
      dnew  <- (d-(a)) / ((d-(a))+ ((1-d)-b))
      c     <- (OR1*p1new*snew*(1-dnew)*dnew*snew)/((1-p1new)*snew*(1-dnew)
          +OR1*p1new*snew*(1-dnew))
      dd    <- p1new * ((1-d)-b)
      e     <- (d-a)-c
      f     <- ((1-d)-b)-dd
      eOR   <- (a*f)/(b*e)
      tabel <- cbind(a, b, c, dd, e, f, OR1, OR2)
      a     <- a + .00001
      tabel
    }
    tabel
  }

#=======================================
# Obtain AUC by Moonesinghe's formula
#=======================================
  FreqDom <- NULL
  FreqRec <- NULL
  RRDom   <- NULL
  RRRec   <- NULL
  GstarDom <- NULL
  GstarRec <- NULL
  xDom    <- NULL
  xRec    <- NULL
  for(i in 1:dim(ORg)[1])
    {
      tab3x2 <- if(Pg[i,2]==0)
        {
          reconstruct.2x2table(p=Pg[i,1], d, OR=ORg[i,1])
        }
      else
```

```
        {
          if(Pg[i,2]==1)
            {
              reconstruct.2x3tableHWE(OR=ORg[i,1], p=Pg[i,1], d)
            }
          else
            {
              reconstruct.2x3table(OR1=ORg[i,1], OR2=ORg[i,2], p1=Pg[i,1],
                  p2=Pg[i,2], d)
            }
        }
      if((tab3x2[1]==0 && tab3x2[2]==0) )
        {
          FreqDom[i] <- (tab3x2[3]+tab3x2[4])
          FreqRec[i] <- FreqDom[i]
          RRDom[i] <- ((tab3x2[3])/(tab3x2[3]+tab3x2[4]))/(tab3x2[5]/(
              tab3x2[5]+tab3x2[6]))
          RRRec[i] <- RRDom[i]
        }
      else
        {
          FreqDom[i] <- (tab3x2[1]+tab3x2[2]+tab3x2[3]+tab3x2[4])
          FreqRec[i] <- (tab3x2[1]+tab3x2[2])
          RRDom[i] <- ((tab3x2[1]+tab3x2[3])/(tab3x2[1]+tab3x2[2]+tab3x2
              [3]+tab3x2[4]))/(tab3x2[5]/(tab3x2[5]+tab3x2[6]))
          RRRec[i] <- (tab3x2[1]/(tab3x2[1]+tab3x2[2]))/((tab3x2[3]+tab3x2
              [5])/(tab3x2[3]+tab3x2[4]+tab3x2[5]+tab3x2[6]))
        }

      GstarDom[i] <- (FreqDom[i]*RRDom[i]) / ( (FreqDom[i]*RRDom[i])+(1-
          FreqDom[i]))
      GstarRec[i] <- (FreqRec[i]*RRRec[i]) / ( (FreqRec[i]*RRRec[i])+(1-
          FreqRec[i]))
      xDom[i] <- (GstarDom[i] -FreqDom[i])^2/ ((GstarDom[i]*(1-GstarDom[i
          ]))+(FreqDom[i]*(1-FreqDom[i])))
      xRec[i] <- (GstarRec[i] -FreqRec[i])^2/ ((GstarRec[i]*(1-GstarRec[i
          ]))+(FreqRec[i]*(1-FreqRec[i])))
    }
  AUCDom <- pnorm(sqrt(sum(xDom)))
  AUCRec <- pnorm(sqrt(sum(xRec)))
  list(AUCDom=AUCDom, AUCRec=AUCRec)
}
```

Reference:

Moonesinghe R, Liu T, Khoury MJ. *Evaluation of the discriminative accuracy of genomic profiling in the prediction of common complex diseases.* **Eur J Hum Genet.** 2010;18:485-9.

E Method of Gail

Arguments

Pg	Frequency of risk alleles
ORg	Odds ratio (OR) of risk alleles
g	Number of genetic variants included

Value

AUCintegrate
Estimate the area under the ROC curve using integration

AUCtrapizoidal
Estimate the area under the ROC curve using trapezoidal rule

Example

```
# when OR and frequency are same for all variants
AUCGail (Pg=.05, ORg=2, g=6)

# when OR and frequency are different for all variants
AUCGail (Pg=c(.05,.1,.12,.14), ORg=c(1.2,1.4,1.1,1.5), g=4)
```

Function

```
AUCGail <- function (Pg,ORg,g)
{
# if same OR and frequencies, run:
  p  <- as.matrix(rep(Pg,g))
  OR <- as.matrix(rep(ORg,g))

# if different ORs and frequencies, remove # and run:
# p  <- as.matrix(Pg)
# OR <- as.matrix(ORg)

# warning: setting the following 'Max' value to a large
# value can cause the program to run out of memory
  Max <- dim(p)[1]
  if(Max>14) {Max=14}
  Fp <- c()
  for(i in 1:dim(p)[1])
    {
      Fp <- rbind(Fp, c(p[i,1]^2,(2*p[i,1]*(1-p[i,1])),(1-p[i,1])^2))
  }
Fc <- c()
for(i in 1:dim(OR)[1])
  {
    Fc <- rbind(Fc, c(OR[i,1]^2,OR[i,1],1))
  }
Dc <- Dp <- 1
for(i in 1:Max)
  {
    Dc <- as.vector(outer(Dc, Fc[i,]))
    Dp <- as.vector(outer(Dp, Fp[i,]))
  }
Tab <- data.frame (cbind("Freq"=Dp, "RR"=Dc))
dim(Tab)
attach(Tab)
Frr <- function(number, Table)
  {
    TableSub <- Table[Table$RR <= number,]
    sum(TableSub$Freq)
  }
FDr <- function(k, number, Table)
  {
    TableSub <- Table[(k*Table$RR) <= number,]
    s1 <- sum(apply(TableSub, 1, prod))
    s2 <- sum(apply(Table, 1, prod))
    s1/s2
  }
FDrr <- function(number, Table)
  {
    TableSub <- Table[Table$RR <= number,]
    p1 <- sum(apply(TableSub, 1, prod))
    p2 <- sum(apply(Table, 1, prod))
    p1/p2
  }
k <- 1
memory.size(max = TRUE)
RealNumbers <- k * c(min(Tab$RR)-1, sort(unique((Tab$RR))), max(Tab$RR)
    +1)
Frt  <- NULL
FDrt <- NULL
Frrt <- NULL
FDrrt <- NULL
for(i in 1:length(RealNumbers))
  {
    memory.size(max = TRUE)
    Frrt[i] <- Frr(number=(RealNumbers[i]), Table=Tab)
    Frt[i]  <- Frr(number=(RealNumbers[i]/k), Table=Tab)
    FDrt[i] <- FDr(k=k, number=RealNumbers[i], Table=Tab)
    FDrrt[i]<- FDrr(number=(RealNumbers[i]), Table=Tab)
    }

  plot(0, 0, xlim = c(0, 1), ylim = c(0, 1), type = 'l',
      main = "ROC like curve using Gail's model", xlab="1 - Fr(t)",
      ylab = "1 - FDr(t)", pty = 's')
  segments(0, 0, 1, 1, col = 1)
  lines(1-Frt, 1-FDrt, type = 'l', col = 2, lwd = 2)

# There are two ways to obtain AUC, i.e. by integration or by
# using the trapezoidal rule. Both methods yield similar results.
# We report results from trapezoidal rule in the paper.
  AUC1 <- integrate( approxfun(Frt, (1-FDrt)), 0, 1)

  trapz <- function (x, y)
    {
      idx = 2:length(x)
      return(as.double((x[idx] - x[idx - 1]) %*% (y[idx] + y[idx -1]))/2)
    }
  AUC2 <- trapz(x=Frt, y= (1-FDrt)) # AUC using tripezoidal rule
  list(AUCintegrate=AUC1, AUCtrapizoidal=AUC2)
}
```

Note We used $k = 1$ in the calculation of the AUC values, which means that we calculated AUC from the distributions of relative risks instead of the distributions of predicted disease risks. This choice does not impact the estimates.

Reference:

Gail MH. *Discriminatory accuracy from single-nucleotide polymorphisms in models to predict breast cancer risk.* **J Natl Cancer Inst.** 2008; 100: 1037-41.