Today is December 11, 2012.

```
> library("vcf2R")
> library("trioClasses")
> library("trio")
> data("BMP4-european-all.geno")
> data(ped, package = "trioClasses")
```

After loading th enecessary packages and data we first make sure that the pedigree data frame contains fields F, M and O for father, mother and offspring ids. Note that these ids should match those in the vcf file.

```
> pedigreeInfo <- within(ped.df, {
      F <- as.character(fid)
      M <- as.character(mid)
      O <- as.character(id)
  })
> tg.ped <- Pedigree(pedigreeInfo = pedigreeInfo)
> tg.ped

This pedigree object contains 1812 complete trios.
For access to the data frame use the trios() accessor function.
```

First we do our best to retrieve ids from the vcf/geno data and manipulate them to match the pedigree file.

```
> id.vec <- colnames(geno.mat)
> head(id.vec)

[1] "H_ME-DS11103_02-DS11103_02" "H_ME-DS11103_03-DS11103_03"
[3] "H_ME-DS11107_02-DS11107_02" "H_ME-DS11107_03-DS11107_03"
[5] "H_ME-DS11107_01-DS11107_01" "H_ME-DS11108_02-DS11108_02"

> foo <- strsplit(x = id.vec, split = "-")
> id.vec <- as.character(data.frame((do.call("rbind", foo)))[,
      3])
> length(id.vec)

[1] 218

> head(id.vec)

[1] "DS11103_02" "DS11103_03" "DS11107_02" "DS11107_03" "DS11107_01"
[6] "DS11108_02"
```

There are 218 subjects in the genotpye matrix. Now with ids formatted properly we create the genotype object needed for the gTrio class, as well as the accompanying ped object such that all trio memebrs have data in the genotype matrix. These values may all be NA, as NA is not precluded from the genotype matrix.

```
> genomat <- t(ifelse(geno.mat == "0/0", 0L, ifelse(geno.mat ==
      "0/1", 1L, ifelse(geno.mat == "1/1", 2L, NA))))
> colnames(genomat) <- rownames(geno.mat)
> rownames(genomat) <- id.vec
> geno.trio <- genoMat(tg.ped, genomat)
> (tg.ped.complete <- completeTrios(tg.ped, colnames(geno.trio)))
```

```
This pedigree object contains 65 complete trios.
For access to the data frame use the trios() accessor function.
```

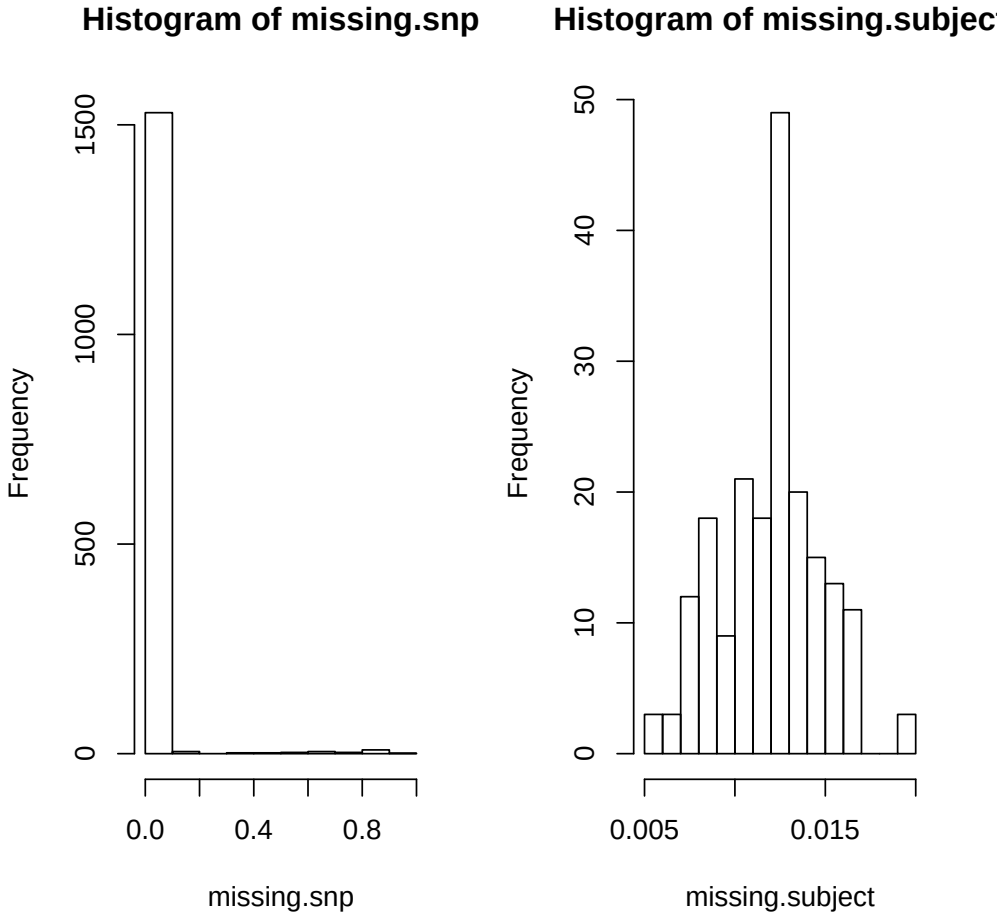
Now we create the gTrio object.

```
> (gTrio.obj <- gTrio(tg.ped.complete, geno = geno.trio))

This object has 65 trios and 1559 markers (likely SNPs).

> missing.snp <- rowSums(is.na(geno(gTrio.obj)))/dim(geno(gTrio.obj))[2]/dim(geno(gTrio.obj))[3]
> missing.subject <- colSums(is.na(geno(gTrio.obj)))/dim(geno(gTrio.obj))[1]
> length(missing.subject)

[1] 195

> length(missing.snp)

[1] 1559


> geno <- geno(gTrio.obj)
> maf <- rowSums(geno[, , 1:2], dims = 1, na.rm = TRUE)/rowSums(!is.na(geno[,
      , 1:2]), dims = 1)/2
> maf <- ifelse(maf >= 0.5, 1 - maf, maf)
> summary(maf)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.00000 0.00000 0.01857 0.00000 0.50000


> geno.trio.2 <- geno.trio[-c(which(missing.snp > 0.1),
      which(maf < 0.05)), ]
> (gTrio.obj <- gTrio(tg.ped.complete, geno = geno.trio.2))

This object has 65 trios and 111 markers (likely SNPs).

> geno <- getGeno(gTrio.obj, type = "holger")

> geno <- geno(gTrio.obj)
> maf2 <- rowSums(geno[, , 1:2], dims = 1, na.rm = TRUE)/rowSums(!is.na(geno[,
      , 1:2]), dims = 1)/2
> maf2 <- ifelse(maf2 >= 0.5, 1 - maf2, maf2)
```

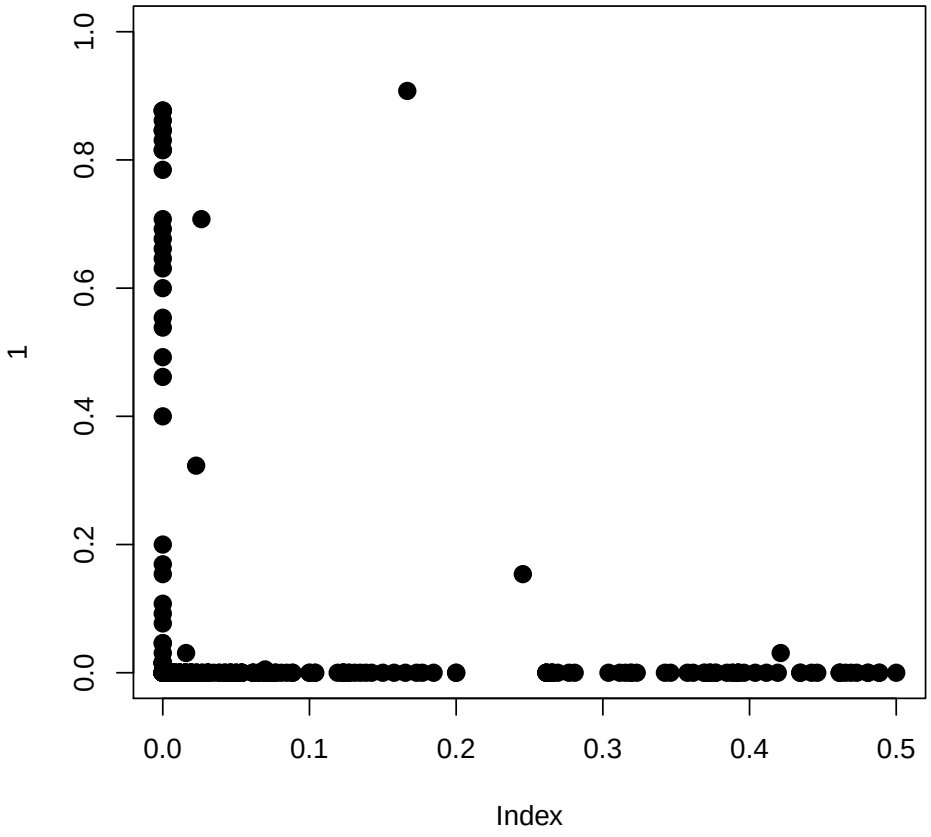
Now, it's easy to perform any method in Holger's trio package, such as alllelicTDT.

```
> geno <- getGeno(gTrio.obj, type = "holger")
> (aTDT <- allelicTDT(mat.snp = geno, size = 10000))

      Allelic TDT

Top 5 SNPs:
            Statistic  p-value
14:54427602     7.143 0.007526
14:54429292     7.143 0.007526
14:54429785     7.143 0.007526
14:54430408     7.143 0.007526
14:54431127     7.143 0.007526
```

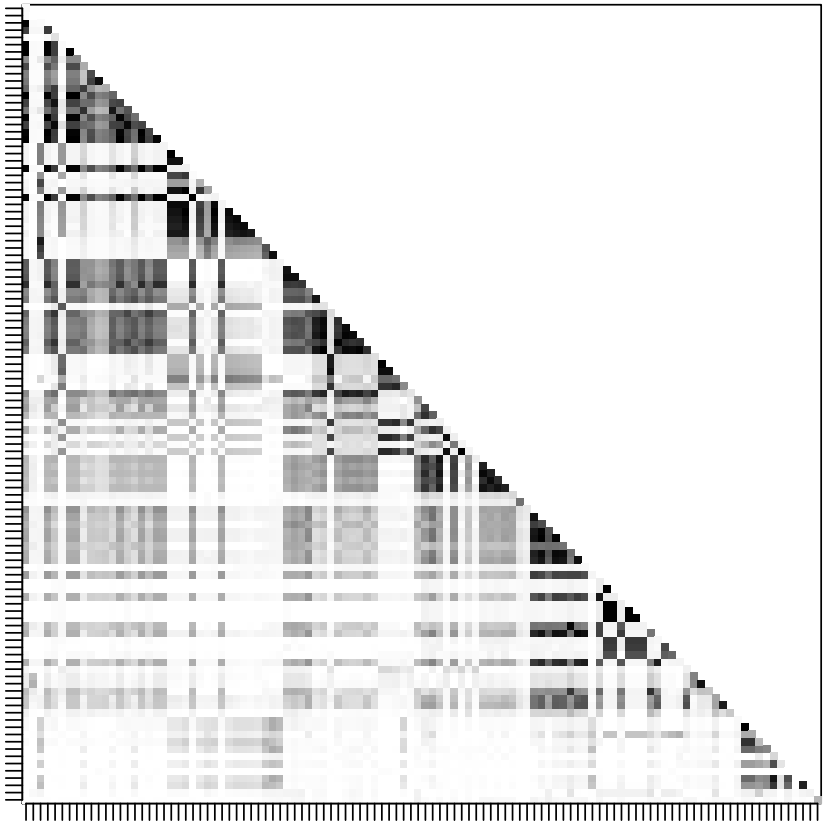
```
> layout(matrix(1:2, ncol = 2, nrow = 1))
> hist(missing.snp, breaks = 10)
> hist(missing.subject, breaks = 10)
```



```
> plot(1, type = "n", ylim = c(0, 1), xlim = c(0, 0.5))
> points(x = maf, y = missing.snp, pch = 20, cex = 2)
```



```
> ld <- getLD(x = geno, which = "both", parentsOnly = TRUE)
> plot(ld, y = "rSquare", xlab = "", ylab = "", cexAxis = 0.01)
```



```
> (gTDT <- tdt(snp = geno))

        Genotypic TDT Based on 3 Pseudo Controls

Model Type: Additive

    Coef     OR  Lower Upper      SE Statistic  p-Value
 -0.0836 0.9198 0.8686 0.974 0.02922     8.183 0.004229
```

```
> hist(maf, breaks = 10)
```