

```

> library("vcf2R")
> library("snpStats")
> data("BMP4-european-all.sm")
> data(ped, package = "trioClasses")

```

After loading the necessary 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 <- with(ped.df, {
  F <- as.character(fid)
  M <- as.character(mid)
  O <- as.character(id)
  ped <- substr(id, 1, 7)
  father <- F
  mother <- M
  affected <- rep(FALSE, length(O))
  affected[grep("_01", O)] <- TRUE
  data.frame(ped = ped, id = id, father = father, mother = mother,
    affected = affected)
})
> rownames(pedigreeInfo) <- pedigreeInfo$id

```

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

```

> id.vec <- rownames(sm)
> 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"

> rownames(sm) <- id.vec

> maf <- colSums(as(sm, "numeric"), na.rm = TRUE)/colSums(!is.na(sm))/2
> maf <- ifelse(maf > 0.5, 1 - maf, maf)
> sub <- which(maf >= 0.05)

> tdt.obj <- tdt.snp(data = pedigreeInfo, snp.data = sm,
  snp.subset = sub)

```

Analysing 65 potentially complete trios in 65 different pedigrees

```
> qq.chisq(chi.squared(tdt.obj, 1), df = 1)
```

N	omitted	lambda
113.000000	0.000000	2.518584

