

# TDT vignette

## Use of `snpStats` in family-based studies

David Clayton

May 1, 2024

### Pedigree data

The `snpStats` package contains some tools for analysis of family-based studies. These assume that a subject support file provides the information necessary to reconstruct pedigrees in the well-known format used in the *LINKAGE* package. Each line of the support file must contain an identifier of the *pedigree* to which the individual belongs, together with an identifier of subject within pedigree, and the within-pedigree identifiers for the subject's father and mother. Usually this information, together with phenotype data, will be contained in a dataframe with rownames which link to the rownames of the `SnpMatrix` containing the genotype data. The following commands read some illustrative data on 3,017 subjects and 43 (autosomal) SNPs<sup>1</sup>. The data consist of a dataframe containing the subject and pedigree information (`pedData`) and a `SnpMatrix` containing the genotype data (`genotypes`):

```
> require(snpStats)
> data(families)
> genotypes
```

```
A SnpMatrix with 3017 rows and 43 columns
Row names: id02336 ... id02732
Col names: rs91126 ... rs98918
```

```
> head(pedData)
```

	familyid	member	father	mother	sex	affected
id02336	fam0005	1	NA	NA	1	1
id00695	fam0005	2	NA	NA	2	1
id02750	fam0005	3	1	2	2	2
id01836	fam0005	4	1	2	2	2
id02533	fam0006	1	NA	NA	2	1
id01069	fam0006	2	NA	NA	1	1

---

<sup>1</sup>These data are on a much smaller scale than would arise in genome-wide studies, but serve to illustrate the available tools. Note, however, that execution speeds are quite adequate for genome-wide data.

The first family comprises four individuals: two parents and two sibling offspring. The parents are “founders” in the pedigree, *i.e.* there is no data for their parents, so that their `father` and `mother` identifiers are set to `NA`. This differs from the convention in the *LINKAGE* package, which would code these as zero. Otherwise coding is as in *LINKAGE*: `sex` is coded 1 for male and 2 for female, and disease status (`affected`) is coded 1 for unaffected and 2 for affected.

## Checking for mis-inheritances

The function `misinherits` counts non-Mendelian inheritances in the data. It returns a logical matrix with one row for each subject who has any mis-inheritances and one column for each SNP which was ever mis-inherited.

```
> mis <- misinherits(data=pedData, snp.data=genotypes)
> dim(mis)
```

```
[1] 114  37
```

Thus, 114 of the subjects and 37 of the SNPs had at least one mis-inheritance. The following commands count mis-inheritances per subject and plot its frequency distribution, and similarly, for mis-inheritances per SNP:

```
> per.subj <- apply(mis, 1, sum, na.rm=TRUE)
> per.snp <- apply(mis, 2, sum, na.rm=TRUE)
> par(mfrow = c(1, 2))
> hist(per.subj, main='Histogram per Subject', xlab='Subject')
> hist(per.snp, main='Histogram per SNP', xlab='SNP')
```

### Histogram per Subject



### Histogram per SNP



Note that mis-inheritances must be ascribed to offspring, although the error may lie with the parent data. The following commands first extract the pedigree identifiers for mis-inheriting subjects and go on to chart the numbers of mis-inheritances per family:

```
> fam <- pedData[rownames(mis), "familyid"]  
> per.fam <- tapply(per.subj, fam, sum)  
> par(mfrow = c(1, 1))  
> hist(per.fam, main='Histogram per Family', xlab='Family')
```

## Histogram per Family



None of the above analyses suggest serious problems with the data, although there are clearly a few genotyping errors.

## TDT tests

At present, the package only allows testing of discrete disease phenotypes in case–parent trios — basically the Transmission/Disequilibrium Test (TDT). This is carried out by the function `tdt.snp`, which returns the same class of object as that returned by `single.snp.tests`; allelic (1 df) and genotypic (2 df) tests are computed. The following commands compute

the tests, display the  $p$ -values, and plot quantile–quantile plots of the 1 df tests chi-squared statistics:

```
> tests <- tdt.snp(data = pedData, snp.data = genotypes)
```

Analysing 1466 potentially complete trios in 733 different pedigrees

```
> cbind(p.values.1df = p.value(tests, 1),
+       p.values.2df = p.value(tests, 2))
```

	p.values.1df	p.values.2df
rs91126	0.3034837	1.385e-02
rs62927	0.1113713	2.810e-01
rs79960	0.6942913	1.506e-05
rs19348	0.0895551	2.013e-01
rs99786	0.0072618	2.713e-02
rs36984	0.1434326	5.476e-03
rs52628	0.9178502	2.469e-01
rs6699	0.0001807	4.812e-05
rs12373	0.4590596	5.772e-01
rs35215	0.2115224	4.463e-01
rs41229	0.0159203	3.931e-02
rs86267	0.1344540	5.129e-04
rs23261	0.6174657	5.535e-03
rs69208	0.0854324	1.671e-01
rs16483	0.6603136	8.925e-01
rs8558	0.4961518	5.590e-01
rs55762	0.0689901	1.913e-01
rs8124	0.2336604	1.364e-01
rs72056	0.0298914	7.391e-02
rs82369	0.0813984	2.131e-01
rs97686	0.5612452	8.410e-01
rs77065	0.7236736	NA
rs53106	0.9586501	5.387e-02
rs37378	0.2194916	8.937e-03
rs83832	0.8755190	8.936e-01
rs35431	0.4226781	4.597e-01
rs61158	0.5343400	5.050e-01
rs32410	0.0387410	5.317e-02
rs85906	0.2319977	4.759e-01
rs83977	0.2807488	2.069e-01
rs24527	0.2963307	3.894e-01
rs73721	0.0729240	9.018e-03
rs36088	0.0324330	3.061e-02

rs32998	0.5571397	7.510e-01
rs5566	0.0672924	3.919e-02
rs98256	0.5858278	7.995e-01
rs29479	0.8193228	2.904e-01
rs42938	0.0611009	1.147e-05
rs32018	0.7280652	4.997e-02
rs39483	0.2304232	4.391e-03
rs42367	0.2674484	2.496e-01
rs87640	0.0223276	3.507e-02
rs98918	0.0581770	2.983e-04

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

	N omitted	lambda
	43.000	0.000 3.401



Since these SNPs were all in a region of known association, the overdispersion of test statistics is not surprising. Note that, because each family had two affected offspring, there were twice as many parent-offspring trios as families. In the above tests, the contribution of the two trios in each family to the test statistic have been assumed to be independent. When there is *linkage* between the genetic locus and disease trait, this assumption is incorrect and an alternative variance estimate can be used by specifying `robust=TRUE` in the call. However, in practice, linkage is very rarely strong enough to require this correction.