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. The data consist of a dataframe containing the subject and pedigree information (`pedData`) and a `SnpMatrix` containing the genotype data (`genotypes`):

```r
> 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    NA    2    2
   id01836     fam0005     4     1    NA    2    2
   id02533     fam0006     1    NA    NA    2    1
   id01069     fam0006     2    NA    NA    1    1
```

1These 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')
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:

```r
> 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')
```
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:

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

Analysing 1466 potentially complete trios in 733 different pedigrees

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

<table>
<thead>
<tr>
<th>rs</th>
<th>p.values.1df</th>
<th>p.values.2df</th>
</tr>
</thead>
<tbody>
<tr>
<td>rs91126</td>
<td>0.3034837</td>
<td>1.385e-02</td>
</tr>
<tr>
<td>rs82927</td>
<td>0.1113713</td>
<td>2.810e-01</td>
</tr>
<tr>
<td>rs79960</td>
<td>0.6942913</td>
<td>1.506e-05</td>
</tr>
<tr>
<td>rs19348</td>
<td>0.0895551</td>
<td>2.013e-01</td>
</tr>
<tr>
<td>rs99786</td>
<td>0.0072618</td>
<td>2.713e-02</td>
</tr>
<tr>
<td>rs36984</td>
<td>0.1434326</td>
<td>5.476e-03</td>
</tr>
<tr>
<td>rs52628</td>
<td>0.9178502</td>
<td>2.469e-01</td>
</tr>
<tr>
<td>rs6699</td>
<td>0.0001807</td>
<td>4.812e-05</td>
</tr>
<tr>
<td>rs12373</td>
<td>0.4590596</td>
<td>5.772e-01</td>
</tr>
<tr>
<td>rs35215</td>
<td>0.2115224</td>
<td>4.463e-01</td>
</tr>
<tr>
<td>rs41229</td>
<td>0.0159203</td>
<td>3.931e-02</td>
</tr>
<tr>
<td>rs86267</td>
<td>0.1344540</td>
<td>5.129e-04</td>
</tr>
<tr>
<td>rs23261</td>
<td>0.6174657</td>
<td>5.535e-03</td>
</tr>
<tr>
<td>rs69208</td>
<td>0.0854324</td>
<td>1.671e-01</td>
</tr>
<tr>
<td>rs16483</td>
<td>0.6603136</td>
<td>8.925e-01</td>
</tr>
<tr>
<td>rs8558</td>
<td>0.4961518</td>
<td>5.590e-01</td>
</tr>
<tr>
<td>rs55762</td>
<td>0.0689901</td>
<td>1.913e-01</td>
</tr>
<tr>
<td>rs8124</td>
<td>0.2336604</td>
<td>1.364e-01</td>
</tr>
<tr>
<td>rs72056</td>
<td>0.0298914</td>
<td>7.391e-02</td>
</tr>
<tr>
<td>rs82369</td>
<td>0.0813984</td>
<td>2.131e-01</td>
</tr>
<tr>
<td>rs97686</td>
<td>0.5612452</td>
<td>8.410e-01</td>
</tr>
<tr>
<td>rs77065</td>
<td>0.7236736</td>
<td>NA</td>
</tr>
<tr>
<td>rs53106</td>
<td>0.9586501</td>
<td>5.387e-02</td>
</tr>
<tr>
<td>rs37378</td>
<td>0.2194916</td>
<td>8.937e-03</td>
</tr>
<tr>
<td>rs83832</td>
<td>0.8755190</td>
<td>8.936e-01</td>
</tr>
<tr>
<td>rs35431</td>
<td>0.4226781</td>
<td>4.597e-01</td>
</tr>
<tr>
<td>rs61158</td>
<td>0.5343400</td>
<td>5.050e-01</td>
</tr>
<tr>
<td>rs32410</td>
<td>0.0387410</td>
<td>5.317e-02</td>
</tr>
<tr>
<td>rs85906</td>
<td>0.2319977</td>
<td>4.759e-01</td>
</tr>
<tr>
<td>rs83977</td>
<td>0.2807488</td>
<td>2.069e-01</td>
</tr>
<tr>
<td>rs24527</td>
<td>0.2963307</td>
<td>3.894e-01</td>
</tr>
<tr>
<td>rs73721</td>
<td>0.0729240</td>
<td>9.018e-03</td>
</tr>
<tr>
<td>rs36088</td>
<td>0.0324330</td>
<td>3.061e-02</td>
</tr>
<tr>
<td>rs32998</td>
<td>0.5571397</td>
<td>7.510e-01</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>rs5566</td>
<td>0.0672924</td>
<td>3.919e-02</td>
</tr>
<tr>
<td>rs98256</td>
<td>0.5858278</td>
<td>7.995e-01</td>
</tr>
<tr>
<td>rs29479</td>
<td>0.8193228</td>
<td>2.904e-01</td>
</tr>
<tr>
<td>rs42938</td>
<td>0.0611009</td>
<td>1.147e-05</td>
</tr>
<tr>
<td>rs32018</td>
<td>0.7280652</td>
<td>4.997e-02</td>
</tr>
<tr>
<td>rs39483</td>
<td>0.2304232</td>
<td>4.391e-03</td>
</tr>
<tr>
<td>rs42367</td>
<td>0.2674484</td>
<td>2.496e-01</td>
</tr>
<tr>
<td>rs87640</td>
<td>0.0223276</td>
<td>3.507e-02</td>
</tr>
<tr>
<td>rs98918</td>
<td>0.0581770</td>
<td>2.983e-04</td>
</tr>
</tbody>
</table>

```r
> 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.