Multiple Imputation of Covariate Data

Objective

Most genetic studies and genomewide studies include covariates in the model when testing for association between trait and genotype at a loci. Often certain covariates are missing within a dataset. Analysing the data as is can lead to loss of power. Multiple imputation methods can be used to impute the covariate data. The objective here is to provide a multiple imputation method to impute covariate data.

Data Format

Data should be available in standard linkage format. The data should be provided as a data-frame with the following columns.

Family Id	Individual	Father Id	Mother Id	Genotypes	Cov 1	...	Cov n

The individual id must be at least unique within the family. We recommend following the data format described in the kinship package. This will allow the users to generate a sparse kinship coefficient matrix using the method developed in the kinship package.

Methodology

The multiple imputation (MI) algorithm implemented in this package assumes that the data is missing at random (MAR). In other words, the probability that a particular value of a variable is not observed may depend on the values of other observed variables, but not on the unobserved value of the variable itself.

The user specifies a vector of indices of variables (or columns) in the input data frame. The first step of the MI algorithm is to calculate the maximum likelihood estimate (MLE) of the full covariance matrix of all of these variables. In order to take the missing values of any of these variables into account, the expectation-maximization (EM) algorithm is used for this purpose.

The algorithm used here for imputation is similar to the Markov-Chain Monte Carlo approach (MCMC) as implemented in the MI procedure in SAS. No attempt is made to take advantage of monotonic missing patterns if they exist. For each record with missing data, the parameters determining the distribution of the missing variables conditional on the observed variables is calculated from the current estimate of population parameters. The initial estimate of these parameters is the output of the EM algorithm mentioned above. Values are then drawn from a normal distribution with those calculated parameters to replace the missing values. The overall estimated population parameters are then updated using these imputed values as if they had been observed. These steps are repeated a number of times specified by the user. This number of iterations is referred to as the number of burn-in iterations. No output is produced from the burn-in iterations. When these iterations are complete, they are again repeated a number of times specified by the user. Each of these iterations results in a new copy of the full dataset

with missing data values replaced by random draws from a normal distribution as described in the burn-in specification.

Demo

```
> data(missData)

> names(missData)

[1] "family" "Ind"    "P1"     "P2"     "g"      "Gen"    "Cov1"   "Cov2"

 [9] "Y1"     "Y2"     "Y3"     "Y4"     "Y5"     "Y6"

### Specify the columns of covariates where the probability that a particular value of a variable is not ### observed may depend on the values of other observed variables, but not on the unobserved value ### of the variable itself

> miData<-miCovariate(missData, covInt=c(6,7,8),n.ipmt=10)

>names(miData)

[1] "ID"     "Imp"    "family" "Ind"    "P1"     "P2"     "g"      "Gen"

 [9] "Cov1"   "Cov2"   "Y1"     "Y2"     "Y3"     "Y4"     "Y5"     "Y6"
```

Output

The output object is a data-frame with multiple datasets. In the above example the number of datasets in the data-frame would be 10. Hence this data frame would have 10 times the number of rows as the original dataset. The output data frame also two additional columns. The first column is an ID column added to uniquely identify all the subjects across all imputed datasets. The second column is an imputation number. This lets the user subset based on the imputation number.