**Using an example design from an iTRAQ experiment:**

This experiment consists of a completely randomised design with 8 animals and 2 treatments for the first phase, and a 4-by-4 iTRAQ experiment for the second phase.

The following table shows the allocation of disease status (**Con**trol and **Dis**eased) to runs and tags in the iTRAQ experiments. Since each disease status occurs exactly twice in every run and tag, the disease status is orthogonal to both runs and tags.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Run | Tag | | | |
| 114 | 115 | 116 | 117 |
| 1 | Con | Con | Dis | Dis |
| 2 | Dis | Dis | Con | Con |
| 3 | Dis | Dis | Con | Con |
| 4 | Con | Con | Dis | Dis |

The following table shows the allocation of animals (1 to 8) to runs and tags. For this design, Runs 1 and 2 contain Animals 1 to 4; Runs 3 and 4 contain Animals 5 to 8, hence the animals are not orthogonal to runs. Similarly, Tags 114 and 116 contain Animals 1, 3, 5 and 7 and Tags 115 and 117 contain Animals 2, 4, 6, and 8, hence the animals are also not orthogonal to tags.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Run | Tag | | | |
| 114 | 115 | 116 | 117 |
| 1 | 1 | 2 | 3 | 4 |
| 2 | 3 | 4 | 1 | 2 |
| 3 | 5 | 6 | 7 | 8 |
| 4 | 7 | 8 | 5 | 6 |

Let denote the abundance of a nominal protein in the proteomic sample from animal with disease status labelled with iTRAQ tag assayed in run . The linear model for the above design can then be written as

where µ denotes the overall mean abundance of the nominal protein, τi and γj denote the fixed effects of disease status *i* and tag *j*, respectively; Rk, Al and εijkl denote the random effects of run *k*, animal *l* and measurement error, respectively. These random effects are assumed to be mutually uncorrelated and normally distributed with mean zero and variances of , and .

The following table shows the theoretical ANOVA, with the expected mean square (EMS)corresponding to the above design. The DF is the ... . The mean square (MS), denoted as (i = 1,…, 4), . is the estimate of the th pure error EMS, denoted by (i = 1,…, 4). Pure errors are the EMS which contains only those variance components associated with the random effects.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Source of variation** | **DF** | **MS** | **Pure error of EMS** | **EMS** |
| Between Run |  |  |  |  |
| Between Animal | 1 |  |  |  |
| Residual | 2 |  |  |  |
| Within Run |  |  |  |  |
| Between Animal |  |  |  |  |
| Disease status | 1 |  |  |  |
| Tag | 1 |  |  |  |
| Residual | 4 |  |  |  |
| Within Animal |  |  |  |  |
| Tag | 2 |  |  |  |
| Residual | 4 |  |  |  |

Here, we attempt to illustrate the estimation of the variance components in, using the Fisher’s scoring algorithm. The Fisher’s scoring algorithm is an iterative procedure which can be used to solve maximum likelihood equations. The algorithm stops when the difference between the variance component estimates of two consecutive iterations is less than 1e-7. The formula for the Fisher’s scoring algorithm can be written as

where contains the variance component estimates at the *t*th iteration, contains the variance component estimates at (*t+*1)th iteration, is the inverse of the Fisher’s information matrix and is the score function. Therefore, in order to use the Fisher’s scoring algorithm, we need to define the score function and Fisher’s information matrix. The , are assumed to have a chi-square distribution, i.e.

where is the degrees of freedom corresponding to . The log-likelihood function, L, of the can be then be shown to be

.

The score function, defined as the first derivative of the log-likelihood function with respect to ,

can be written as a vector of four elements

.

The second derivative of the log-likelihood function can then be shown to be the 4-by-4 matrix,

where the diagonal elements can be derived as

,

and the off-diagonal elements can be derived as

Furthermore, the expected values of the diagonal elements can be shown to be

.

The Fisher’s information matrix, defined as the expectation of the negative of the second derivative of the likelihood function, is given by

,

where denotes *i*th diagonal element of the diagonal matrix.

The score function and Fisher’s information matrix defined above can help us to estimate the when applying the Fisher’s scoring algorithm. However, we want to estimate each of the individual of the variance components in. Hence, we want to transform the score function and the expected Fisher’s information matrix with respect to to with respect to a vector of variance components,. This transformation can be achieved by multiplying , this technique is also known as *change of variables*. For this experiment, the transformation can be achieved by using a 4-by-3 matrix, denoted by G matrix, where each element of the G matrix is. The G matrix can be written as

.

As result, the G matrix contains the coefficients of the variance components of EMS.

Hence, the , can be estimated using by

.

Thus, the score function and the Fisher’s information matrix with respect to θ can be shown to be

and

.

**Pseudo code of simulation and Fisher’s scoring algorithm for estimating the variance components**

For this case, 10000 simulated datasets are generated. The variance component estimates are obtained for each simulated data. The means of the 10000 sets of variance components estimates are then computed.

reml.VC = matrix(0, nrow=1, ncol = 3) # matrix used to store the variance component estimates from each simulated data set

Repeat 10000 times{

#Simulate a single dataset based on the linear model.

VC.base = variance component of the measure error.

VC.animal = variance component of the animal effects.

VC.run = variance component of the run effects.

Simulated dataset = N(0, VC.base) + N(0, VC. animal) + N(0, VC.run)

#Construct the theoretical ANOVA table based on the experimental design.

G = a matrix consists of coefficients of the variance components obtained from the theoretical ANOVA table.

DF = vector of degrees of freedom of the corresponding mean square based on the experimental design

#Perform ANOVA on the simulated data.

MS = vector of mean squares from ANOVA based the simulated dataset.

EMS = vector of expected mean squares compute by pre-multiplying the current variance component estimates by the G matrix.

newV = c(VC.base , VC.animal, VC.run) # Vector of current variance component estimates. Initialise VCs to their true values, i.e. values used to simulate the dataset

oldV = c(0, 0, 0) # Vector of previous variance component estimates. Initialise all VCs to zero

counter <- 1 # Initialise counter

#the convergence tolerance is the differences between the current variance component estimates and the previous variance component estimates. This differences should be less than 1e-7

while((newV – oldV) >1e-7){

oldV = NewV

EMS = G’ × oldV

score function =

information matrix =

newV = oldV + (information matrix)-1 × (score function)

if ( counter > 1000 or information matrix is invertible)

stop the iteration of the while loop and start a brand new simulation dataset

counter = counter +1

} #end of while((newV – oldV) >1e-7)

reml.VC = rbind(reml.VC, newV) #store the estimates into a matrix

} #end of repeat 10000 times

apply(reml.VC, 2, mean) #each variance components estimates, i.e. , are then obtained from the means of the variance components estimates from the 10000 simulated datasets