**R functions for obtaining the effective degrees of freedom**

Three R functions were developed, with a given experimental design and data, for improving the estimates of the variance components using restricted maximum likelihood (REML) technique. It follows by assessing the effectiveness of the estimates by generating the effective degrees of freedom (EDF).

The formula for computing the EDF from Richard and Kathy’s paper is calculated as twice the square of the mean divided by the variance. These variances can be obtained by calculating the sum of the elements of interest from the variance covariance matrix. The variance covariance matrix is generated from the inverse of the Fisher’s information matrix, which is the expectation of the second derivative of the likelihood function.

The focus of experimental design is the two-phase MudPIT-iTRAQ experiments. The rest of this write-up will describe what does each R function do and finish with a simple example.

The first R function is called getMSEst(). This function extracts the mean squares (MS) and the degrees of freedom (DF) from the ANOVA table of the aov function. Suppose there are m sets of MS from the ANOVA table which are assumed to have a chi-square distribution. Let these MS to be denoted by , the distribution can be written as,

where the denotes the expected MS and is the DF for MS . The likelihood function can be then be shown as

L = constant - .

The first derivative with respect to can be written as

and the expectation of the negative of the second derivative is

The expected Fisher’s information matrix for the MS is the diagonal matrix containing . Hence, the MS and DF can be extracted from the ANOVA table to generate this Fisher’s information matrix.

The MS and DF are extracted from the sources of variation excluding the treatment information, because the variance estimation should not contain any treatment information. However, there are some cases where the treatment effects are completely confounded with the random effects, i.e. balanced incomplete block design. In these cases, the amount of confounding treatment information can be small enough to be neglected.

Since the focus is on the two-phase experiments, I cannot use the aov function directly as the aov function only applies to a single stage of decomposition. However, two-phase experiments requires two stages of the decomposition, i.e. decomposition of the information from the Phase 1 block structure in the Phase 2 block structure and then decomposition of the information from the treatment structure in the Phase 1 block structure. This is achieved by fitting the Phase 1 block structure as the fixed effects and Phase 2 block structure as random effects. The MS and DF from the decomposition of the Phase 1 block structure in the Phase 2 block structure are extracted. The treatment structure is then fitted as the fixed effects and the sum of the Phase 2 and Phase 1 block structure are fixed as the random effects. The MS and DF are again extracted from the sources of variation that were not obtained the first stage of decomposition.

The second R function getGMat(). G matrix is used to transform the score function and the expected Fisher’s information matrix with respect to to with respected to the parameters of interest, denoted by . Hence, if there are k parameters of interest, the G matrix will have the dimensions m-by-k and is represented by(?) . The infoDecompuTE package was used, because this G matrix is basically the variance components structure to each source of variation. The variance components structure is the coefficients of the variance components of the expected mean squares in the ANOVA table.

Note the getMSEst() function only extract the MS and DF of the source of variation without the treatment information. Hence, the variance components structures are extracted in getGMat() function has to match the sources of variation that were extract from the output of getMSEst() function.

Unlike in Richard and Kathy’s paper, their G matrix is a binary matrix of 0 and 1. To allow our function be more general, the G matrix that is generated here contains the coefficients of the variance components. Having the coefficients in the G matrix, it allows parameter of interest, , to be made up of single term of the variation components and each with coefficient of one. This G matrix is used, because sometime the coefficients of the variance components are not always identical for different sources of variation in the ANOVA table. This avoid of adjusting these coefficients with different linear combination of the variance component for a complicated analysis. Hence, the expected mean squares is .

The third and final R function is getVcEDF(). This function generates the newly optimised variance components and the EDF.

The first step of this function is to transform the score function and the expected Fisher’s information matrix with respected to to the parameter of interest . Record the expected Fisher’s information matrix with respected to can be written as

.

The expected Fisher’s information matrix with respected to , denoted by , can be generated from pre- and post-multiply the by the G matrix, i.e.

The score function with respected to is also obtained from pre-multiplying first derivative of the likelihood function by the transpose of the G matrix, this can be written as

From this, the iterative scheme for estimating the parameters of interest can be derived as

.

This is also known as Fisher’s scoring algorithm of the REML method. The Fisher’s information matrix and score function are continuously updated using the newly optimised parameters . Note that the expected mean squares is also continuously updated since

.

This iteration algorithm will stop when the parameters are not changing anymore.

The variances can be obtained by calculating the sum of the elements of interest from the variance covariance matrix. The variance covariance matrix is generated from the inverse of the Fisher’s information matrix. However, since the parameters that are estimated only have coefficients of one, these coefficients have to be re-adjusted based on the variance components structure of the ANOVA table. This adjustment is based on the idea for calculating the sum of the variances with coefficients is use, which its formula can be written as

.

The effective degrees of freedom are then approximated as twice the square of the mean divided by the variance. Both mean and variance are obtained from the newly optimised parameters.

The end result of the function getVcEDF() is the EDF for every source of variation without the treatment information and the newly optimised parameters.

For further development of these three functions, it relies on the inforDEcompuTE package. An improvement is to write its own function to generate the variance component structure and the G matrix. In addition, matching between the ANOVA table from the aov function and infoDecompuTE package can sometime be problematic, because sometimes the names for sources of variation can become confusing with a more complicated analysis. This again can be improved by manually calculates of these MS instead of using the output from the aov function, hence manually output the names of the sources of variation. This is ensure the names for both outputs are properly matched.

Example

I will start with a simple example consisting of a completely randomised design with 4 animals and 2 treatments for first phase, and 4-by-4 iTRAQ experiment for the second phase experiment.