Methods for Solving the Dyadic Model Equations

Neville Jackson

30 Jan 2025
For dmm_3.1-1

Contents

1 Introduction

The function *dmm()* sets up equations which relate the observed covariance of pairs of individuals or dyads, to their expectation in terms of postulated genetic and environmental variance and covariance components.

These equations, termed *dyadic model equations* (DME's), can be solved directly to obtain estimates of variance and covariance components. The DME's are linear equations, and are exactly analogous to a set of multi-trait multiple regression equations.

The function *dmm()* therefore effectively turns variance component estimation into a regression problem. All of the statistical techniques for fitting a linear multiple regression are therefore available for solving the DME's. The function *dmm()* uses the qr() function by default, but can optionally use *lm()*, robust regression (*lmrob()*), principal component regression (*pls package*), or feasable generalised least squares (*fgls*).

Assumptions about dyadic residuals have to be considered in choosing a method to solve the DME's. The simplest assumption, that dyadic residuals are uncorrelated and equal in variance, leads to ordinary least squares (OLS). In *dmm()* the QR, lm, and pls options all use ordinary least squares. The robust regression (lmrob) option uses OLS after deleting dyads considered to be outliers. The fgls option is an implementation of generalised least squares (GLS), so it takes account of the known covariance structure of the dyadic residuals (see Section ....).

The type of variance component estimate that is obtained from *dmm()* depends on

1. The method (OLS or GLS) used to fit the fixed effects model, from which the residual variances to be partitioned into components are obtained
2. The method used to fit the dyadic or random effects model which does variance component estimation.

The relationship between fitting methods and the type of variance component estimate obtained is summarized in Table 1

The types of variance component estimate are as follows

MINQUE minimum variance quadratic unbiased estimate defined by Rao(1971) [5]

BCML bias corrected maximum likelihood estimate as derived by Anderson(1978) [1]

REML restricted maximum likelihood estimate Patterson and Thompson(1974) [4]

RMINQUE robust MINQUE estimate

RML robust ML estimate

Table 1: Relationship between fitting methods for fixed and dyadic models and tpyes of variance component estimate obtained

Method for fixed model	Method for dyadic model		
	Ordinary Least Squares qr or lm or pls	Generlised Least Squares fgls	Robust Regression lmrob
ols	MINQUE	REML	robust MINQUE
gls	BCML	REML	robust ML

The *dmm()* program can obtain any of the above estimates. By default *dmm()* uses ordinary least squares (OLS) for both the fixed model and the dyadic model. So the default variance component estimate type is MINQUE. If *gls* is specified for the fixed model *dmm()* does *ols* for the fixed model first, then does the *gls* iteration, and reports both *ols* and *gls* results. If *fgls* is specified for the dyadic model, *dmm* obtains REML variance component estimates, regardless of the method used for fixed effects. It is a property of REML estimates that they are independent of the method used to remove fixed effects. If *lmrob* is specified for the dyadic model, the corresponding robust estimates are obtained.

There are some limitations with *dmm()*. Robust regression can only be used for univariate models. The *fgls* option which leads to REML estimates is severely restricted by memory requirements. Obtaining REML estimates by doing generalised least squares on the dyadic model equations is an order N^4 problem where N is the number of individuals with data. In practice the *fgls* option can only handle about 200 individuals, so it is of academic interest only. If you want REML estimates use the *gremlin()* CRAN package (Wolak(2024) [8] All the other *dmm()* options are of order N^2 and they are useful up to about 10000 individuals.

Multivariate models require extra caution in interpretation. Multivariate analysis amounts to treating all the traits as if they were one trait and analysing the total variance. Multivariate MINQUE estimates will be idential to the univariate MINQUE estimate for each trait. Multivariate BCML estimates will usually be similar to univariate BCML estimates for each trait, with minor deviations for traits that are highly correlated. Multivariate REML estimates will be quite different from univariate REML estimates, because using GLS on the dyadic model equations takes acount of the correlated error structure of the dyadic model.

MINQUE, BCML, and REML estimates will agree for simple balanced data sets. Robust regression estimates will differ. For complex pedigrees, MINQUE and BCML estimates of variance components will be estimates of the variances existing in the population of related individuals from which the data were obtained. REML estimates will be adjusted to estimate what the variances would have been if the individuals were unrelated. This is because using GLS on the dyadic model equations allows for the covariance structure of the dyadic errors. The dyadic error covariances come from two sources, the covariance structure of products, and the relationship matrix between individuals.

Because of the above, REML estimates need to be interpreted differently from MINQUE and BCML. If you want the conventional genetic parameters which are adjusted to a hypothetical population of unrelated individuals, use REML. If you want to know what the selection responses might be in the population at hand, use MINQUE or BCML.

In many of the datasets available to quantitative geneticists, the (co)variance components which we would like to estimate are partially confounded, sometimes to the point where they are not separably estimable. This is particularly so in dealing with nonadditive genetic components. The function *dmm()* offers an experimental approach (*pls()* option) which allows partially confounded components to be estimated by constraining some components, using principal component regression. This is still ordinary least squares, but it is constrained to a certain subspace of the full variance component space.

A related issue is omitted variable bias. If you leave a variance component which is large and significant out of the fitted dyadic model, its variance does not necessarily go into dyadic error, but will be spread unpredictably among the components which are fitted, resulting in biased estimates. It is therefore always important to include all significant variances in the dyadic model. There will be problems if all significant variances are not separable with the data at hand. The *pls* option may help with this, by allowing a fit of a combination of variance components, rather than fitting them individually.

2 The dyadic model equations

The dyadic model is presented in Section 6.2.2 of the document *dmmOvervire.pdf* [2]. It results in a set of equations (the DME's) which are given in matrix form as equation 12, which is reproduced below

$$\Psi = W\Gamma + \Delta \tag{1}$$

It is important to understand each of the matrix components of these equations, so we elaborate as follows First, the following variables set the size of the problem and the sizes of the above matrices

n number of individuals with data

m number of individuals in pedigree

l number of traits

k number of fixed effects

c number of variance components to be estimated

Now we explain each matrix

$\boldsymbol{\Psi}$ $n^2 \times l^2$ matrix of dyadic covariances for each pair of individuals (row) and each traitpair (col). Each covariance needs to be appropriately adjusted for fixed effects. The columns of $\boldsymbol{\Psi}$ become the dependent variables in a multi-trait multiple regression.

$\boldsymbol{W}$ $n^2 \times c$ matrix containing the coefficients of the dyadic model equations, which become the independent variables of a multiple regression. Each column of $\boldsymbol{W}$ has the form $Vec(\boldsymbol{M}\boldsymbol{Z}_c\boldsymbol{R}_c\boldsymbol{Z}_c'\boldsymbol{M}')$ where Vec is an operator that vectorizes a matrix, $\boldsymbol{M}$ is a matrix from the fixed effect model such that $Y - X\hat{\alpha} = MY$, $\boldsymbol{Z}_c$ is an incidence matrix relating individuals with data to individuals in the pedigree, and $\boldsymbol{R}_c$ is a relationship matrix relevant to component c. Note that relationship matrices are used, not their inverse.

$\boldsymbol{\Gamma}$ $c \times l^2$ matrix of (co)variance component parameters to be estimated, which become the partial regression coefficients of a multiple regression.

$\boldsymbol{\Delta}$ $n^2 \times l^2$ matrix of dyadic model residuals. Note the variance of these is not the individual environmental variance component - that has to be explicitely fitted as one of the columns of matrix $\boldsymbol{W}$ and appears as one row of matrix $\boldsymbol{\Gamma}$. The $\boldsymbol{\Delta}$ matrix elements are the extent to which each dyadic covariance in matrix $\boldsymbol{\Psi}$ deviates from its expectation. The (co)variances of the elements of $\boldsymbol{\Delta}$ enter into the standard errors of variance component estimates, in the same way that the (co)variances of residuals enter into the standard errors of any regression coefficient estimates.

Matrices $\boldsymbol{\Psi}$ and $\boldsymbol{W}$ have n^2 rows, so we are looking at solving n^2 equations in c unknowns. This is an overdetermined system. We can use least squares to obtain an approximate solution.

3 Checking the dyadic model assumptions

Before attempting a regression fit of model (1), it is worth looking at how well the data conform to the assumptions made in using using least squares to fit a multiple regression model. The critical assumptions are

- the independent variables (columns of $\boldsymbol{W}$) are uncorrelated
- the residuals (columns of $\boldsymbol{\Delta}$) are uncorrelated with each other and with the independent variates

A least squares fit does not involve assumptions regarding the distribution of residuals, but this does become involved when using residuals to obtain standard errors of parameter estimates.

3.1 The assumption of independence of columns of $\boldsymbol{W}$

The correlations among independent variables (columns of $\boldsymbol{W}$) are returned by *dmm()* in the attribute *dme.correl* of the returned object. These are pairwise correlations between the columns of the $\boldsymbol{W}$ matrix. They are not quite the same thing as sampling correlations between variance component estimates, but they will be similar . They are also not quite the same thing as correlations between estimable components as obtained from correlations between relationship matrix elements, because the columns of $\boldsymbol{W}$ involve $\boldsymbol{M}$ and $\boldsymbol{Z}$ matrices as well as the relationship matrix.

The $\boldsymbol{Z}$ matrix simply selects a subset of the relationship matrix corresponding to those individuals which have observations. The $\boldsymbol{M}$ matrix comes from the fixed effects model, and consists of a set of weights which adjust for the degrees of freedom involved in each of the fixed effects applicable to each individual. So the columns of the $\boldsymbol{W}$ matrix are a weighted subset of the elements of the appropriate relationship matrix. Their correlations are therefore not the same as relationship matrix element correlations.

In the case of a dataset with mean only, and all individuals in the pedigree with data, such as the *warcolak* dataset, the subset is all of the relationship matrix and the weights in $\boldsymbol{M}$ are all equal, so the correlations of the columns of $\boldsymbol{W}$ are exactly the same as the correlations of the elements of the relationship matrices, in this special case.

In regression analysis it is generally considered that if there are collinearities among the independent variates amounting to correlations greater than around 0.5 then the estimates of regression coefficients are suspect. Translating this to our dyadic model, the variance component estimates are not likely to achieve a realistic separation if the columns of $\boldsymbol{W}$ have correlations exceeding 0.5. The option of using principal component regression (*dmeopt="pcr"* has been developed as an experimental approach to dealing with serious collinearities among the components.

If we want to plot these correlations (eg as scatteplots) we need argument *dmekeep=T* in calling *dmm()*. The dafault for *dmm()* is not to save the DME's in its return object. This can be overridden with argument *dmekeep=T*. This results in 2 attributes *dme.psi* and *dme.wmat* being added to the returned object. Caution, this can result in a very large returned object. The attribute *dme.wmat* is the $\boldsymbol{W}$ matrix, and its columns can be plotted with the standard *plot()* routine.

3.2 The assumptions of independence of dyadic residuals

To check the dyadic residuals, we can use the S3 *plot()* method included in the *dmm* package. This will output histograms, qqnorm plots, and scatterplots of residuals against fitted and observed values. Plots tend to be more informative for datasets with smaller numbers of individuals.

Dyadic residuals are usually not far from normally distributed, but may be leptokurtic and slightly skewed to the right.

If dyadic residuals are correlated with fitted values of the components, then there is something wrong with the model, probably some extra component should be fitted.

One can expect dyadic residuals to be correlated with observed dyadic covariances. It is normal for the fitted components in a dyadic model to explain only a small fraction of the total variation, and this of course leads to the observed covariances being highly correlated with residuals.

The question of patterns of correlation among the dyadic residuals themselves is a seriously difficult area. The covariances (or correlations) among the dyadic residuals for one trait form an $n^2 \times n^2$ matrix - ie n^4 elements. Too many to compute and cant be stored in R. This issue is discussed in the document *dmmOverview.pdf* [2] in relation to why *dmm()* is not able to do REML estimates. REML estimates require that the covariance (or correlation) matrix of the dyadic residuals be constructed and used to compute a GLS rather than an OLS solution to the DME's. That is not computationally feasible, and neither is an examination of the residual covariance/correlation matrix.

The covariance structure of dyadic residuals is actually known. It is derived in Searle et al (1992) [6] on pages 407-413. It involves fourth moments of the observations. It is discussed in the document *dyadicerrorcov.pdf* [3].

4 Examples

4.1 Using principal component regression to investigate independence of the variance component parameters in the *warcolak* dataset

Using the *warcolak* dataset from the *nadiv* package (Wolak(2014) [7], and just analysing Trait2, we first setup the data file making all the appropriate relationship matrices, then run the usual analysis fitting 4 variance components, using the default "qr" option for solving the DME's, and setting options to save the DME's and the fit object.

```
> library(dmm)
> data(warcolak)
> warcolak.df <- warcolak.convert(warcolak)
> warcolak.mdf.univ <- mdf(warcolak.df,pedcols=c(1:3),factorcols=4,ycols=c(5:6),
               sexcode=c("M","F"),relmat=c("E","A","D","S"),keep=T)
  .....
Warning message:
In makeS(ped2, heterogametic = sexcode[1], returnS = T, DosageComp = "ngdc") :
  Assuming male heterogametic (e.g., XX/XY) sex chromosome system

> warcolak.fit.t2 <- dmm(warcolak.mdf.univ, Trait2 ~ 1,
            components=c("VarE(I)","VarG(Ia)", "VarG(Id)","VarGs(Ia)"),
            relmat = "withdf",dmekeep=T,dmekeepfit=T)
  .....
> summary(warcolak.fit.t2)
Call:
summary.dmm(object = warcolak.fit.t2)

Coefficients fitted by OLS for fixed effects:

   Trait Estimate StdErr CI95lo CI95hi
1 Trait2     1.54  0.015   1.51   1.57


Components partitioned by DME from residual var/covariance after OLS-fixed-effects fit:

             Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Trait2:Trait2    0.715 0.0438 0.6295  0.801
VarG(Ia)  Trait2:Trait2    0.138 0.0159 0.1066  0.169
VarG(Id)  Trait2:Trait2    0.153 0.0452 0.0645  0.241
VarGs(Ia) Trait2:Trait2    0.270 0.0154 0.2392  0.300

>
```

These results differ from those for Trait 2 in the analysis reported in dmmOverview.pdf [2]. The difference is that we have not fitted a Sex fixed effect and we have saved some additional information.

The correlations among columns of $\boldsymbol{W}$ are, of course, the same

```
> round(warcolak.fit.t2$dme.correl,digits=3)
          VarE(I) VarG(Ia) VarG(Id) VarGs(Ia)
VarE(I)     1.000    0.486    0.919     0.408
VarG(Ia)    0.486    1.000    0.626     0.827
VarG(Id)    0.919    0.626    1.000     0.525
VarGs(Ia)   0.408    0.827    0.525     1.000
>
```

Two of these correlations are somewhat larger than the nominal 0.5 mentioned above. We need to look and see if the multiple regression has been able to separate these 4 components properly. To that end we will redo the analysis using *dmeopt="pcr"* instead of the default *"qr"*. Using the principal component regression option requires at least two runs - the first retaining all 4 principal components, and then one or more reruns omitting the least important principal components. The first run is as follows

```
> warcolak.fitpcr1 <- dmm(warcolak.mdf.univ, Trait2 ~ 1,
            components=c("VarE(I)","VarG(Ia)","VarG(Id)","VarGs(Ia)"),
            relmat = "withdf",dmekeep=T,dmekeepfit=T,dmeopt="pcr")
  .....
DME substep:
PCR option on dyadic model equations:
Data:  X dimension: 29160000 4
Y dimension: 29160000 1
Fit method: svdpc
Number of components considered: 4

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps
CV           1.023    1.022    1.022    1.022    1.022
adjCV        1.023    1.022    1.022    1.022    1.022

TRAINING: % variance explained
       1 comps   2 comps   3 comps    4 comps
X     79.46086  92.89539  99.28012  100.00000
evec   0.02946   0.03157   0.03158    0.03158
DME substep completed:
OLS-b step completed:
>
> loadings(warcolak.fitpcr1$dme.fit)
Loadings:
            Comp 1 Comp 2 Comp 3 Comp 4
‘VarE(I)‘    0.209  0.663  0.192 -0.692
‘VarG(Ia)‘   0.699         -0.705
‘VarG(Id)‘   0.275  0.630  0.111  0.718
‘VarGs(Ia)‘  0.626 -0.393  0.674

               Comp 1 Comp 2 Comp 3 Comp 4
SS loadings      1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00
>
```

What we need from this at the moment is the % of variance explained by various numbers of principal components. Obviously using all 4 components explains 100% of the variation, but what is interesting is that 3 components explain 99% and 2 components 92%. This is a signal that we should try regressing the observations on 3 principal components of the columns of $\boldsymbol{W}$, and see how it affects the estimates and their standard errors. We do that with a rerun

```
> warcolak.fitpcr2 <- dmm(warcolak.mdf.univ, Trait2 ~ 1,
            components=c("VarE(I)","VarG(Ia)","VarG(Id)","VarGs(Ia)"),
            relmat = "withdf",dmeopt="pcr",ncomp=3)
  .....
```

```
> summary(warcolak.fitpcr2)
Call:
summary.dmm(object = warcolak.fitpcr2)

Coefficients fitted by OLS for fixed effects:

   Trait Estimate StdErr CI95lo CI95hi
1 Trait2   -0.063 0.0138  -0.09 -0.036


Components partitioned by DME from residual var/covariance after OLS-b fit:

              Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Trait2:Trait2    0.286 0.0127  0.262  0.311
VarG(Ia)  Trait2:Trait2    0.315 0.0182  0.279  0.351
VarG(Id)  Trait2:Trait2    0.310 0.0118  0.286  0.333
VarGs(Ia) Trait2:Trait2    0.146 0.0205  0.106  0.186
VarP(I)   Trait2:Trait2    1.057 0.0244  1.010  1.105

>
```

So comparing these estimates with those from the "qr" fit (which are the same as obtained with "pcr" with all 4 principal components), we find that omitting one principal component has not changed the estimated components substantially, and has reduced the standard errors of the two constrained components. By omitting the fourth principal component we have in effect set it to zero, which amounts to constraining the components estimates to be on the plane defined by

$$-0.692 \times VarE(I) + 0.718 \times VarG(Id) = 0 \tag{2}$$

We get this equation from the *loadings* which were given at the end of the run with all four principal components above. If we substitute the estimates of VarE(I) and VarG(Id) from the 3 component run into the above equation we find that they do indeed fall on the constraint plane. So we still get estimates of all 4 components, but two of them are constrained to be in a ratio 0.692/0.718, or approximately equal.

To my mind, that is a more satisfactory analysis than the unconstrained result from a"qr" fit. By regressing on the principal components of columns of $\boldsymbol{W}$ instead of on the columns themselves we have avoided violating the assumption of independence, and by applying one constraint we have improved the standard errors.

One can go on and try omitting two principal components. This leads to two constraint equations, so the variance components would be constrained to lie on the intersection of two planes. We shall not do it here. We have done enough to demonstrate the method.

4.2 Using fgls to allow for correlated dyadic residuals. Example with the *Harvey* dataset

The dataset called *harv101.df* is fully described in the *dmmOverview.pdf* [2] document. Here we will only look at the use of feasable GLS to solve the dyadic model equations allowing for correlated residuals and heterogenic residual variances.

```
> library(dmm)
> data(harv101.df)

> harv101.mdf <- mdf(harv101.df, pedcols=c(1:3), factorcols=c(4:5,9),
                  ycols=8, sexcode=c(1,2),keep=T,relmat=c("E","A"))
```

```
> harv101.fgls.fit2 <- dmm(harv101.mdf,Adg ~ 1 + Line + Agedam + Age + Weight,
   dmeopt="fgls", dmekeepfit=T,dmekeep=T,relmat="withdf")
Dyadic mixed model fit for datafile: harv101.mdf
Data file is a normal dataframe:
Random effect partitioned into components: Residual:
OLS-fixed-effects step:
no of fixed effect df (k) =  7
no of traits (l) =  1
Setup antemodel matrices:
No of factors with specific components: 0
No of non-specific components partitioned: 2
No of factors with specific components: 0
No of specific variance components partitioned (per component): 0
No of specific variance and covariance components partitioned (per component): 0
no of individuals in pedigree (m) =  139
no of individuals with data and X codes (n) =  65
Rank of X: 7   No of Fixed Effects: 7
OLS-fixed-effects step completed:
DME substep:
No of components defined =  2
No of components estimable =  2
Checking dyadic model equations:
fgls iteration starting siga from ols:
            Adg:Adg
VarE(I)  0.01674900
VarG(Ia) 0.05195215
Residual var for DME (vard):
            Adg:Adg
Adg:Adg 0.003253705
Total var for DME (vart):
            Adg:Adg
Adg:Adg 0.003320518
Iteration(fgls) round:  0
siga in fgls.update:
           [,1]
[1,] 0.01783227
[2,] 0.05069428
Round =  1  Stopcrit =  0.0009364542
Iteration(fgls) completed - count =  1
Convergence achieved (fgls)
diagvsiga:
             Adg:Adg
VarE(I)  0.001643798
VarG(Ia) 0.002516052
diagvsiga posdef:
             Adg:Adg
VarE(I)  0.001643798
VarG(Ia) 0.002516052
sesiga:
            Adg:Adg
VarE(I)  0.04054378
VarG(Ia) 0.05016026
DME substep with OLS-fixed-effects completed:
```

```
>
>
>
> summary(harv101.fgls.fit2)
Call:
summary.dmm(object = harv101.fgls.fit1)

Coefficients fitted by OLS for fixed effects:

            Trait Estimate  StdErr    CI95lo   CI95hi
(Intercept)   Adg  2.85434 0.50581  1.862946  3.84574
Line2         Adg  0.09427 0.09566 -0.093231  0.28176
Line3         Adg  0.03574 0.07817 -0.117471  0.18895
Agedam4       Adg -0.03049 0.09770 -0.221988  0.16100
Agedam5       Adg -0.14197 0.08708 -0.312642  0.02871
Age           Adg -0.00816 0.00289 -0.013820 -0.00251
Weight        Adg  0.00252 0.00090  0.000755  0.00428


Components partitioned by DME from residual var/covariance after OLS-fixed-effects fit:

         Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)    Adg:Adg   0.0176 0.0405 -0.0619 0.0971
VarG(Ia)   Adg:Adg   0.0509 0.0502 -0.0474 0.1493

>
```

These are REML estimates of the VarE(I) and VarG(Ia) components.

We can view the dyadic residuals using the S3 *plot()* method included in the *dmm* package. We do this for the "fgls" fit as follows

```
> plot(harv101.fgls.fit2)
```

Five plots will be displayed in five separate graphics screens. The resulting plots are shown in Figure 1 to Figure 5. We can see that the histogram of residuals is more peaked than a normal distribution (Figure 1), that the qqplot is sigmoidal (Figure 2), that the residuals are not associated with fitted values (Figure 3), that the observed values are strongly correlated with the residuals (Figure 4), and that the fitted values and observed values are not strongly correlated (Figure 5).

The covariance matrix of residuals will not be diagonal, that is there are correlations between residuals. That is why "fgls" is the optimal method for fitting the dyadic model and why it leads to REML estimates of the variance components. The detailed covariance structure of the residuals from a dyadic model is dealt with in a separate document *dyadicerrorcov.pdf* [3].

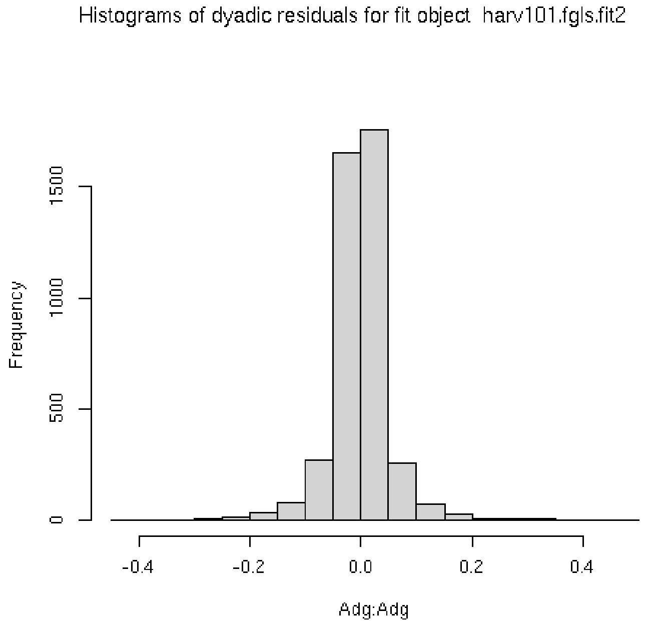

Figure 1: Histogram of dyadic residuals for Trait ADG of Harvey dataset with two components fitted usng "fgls" method

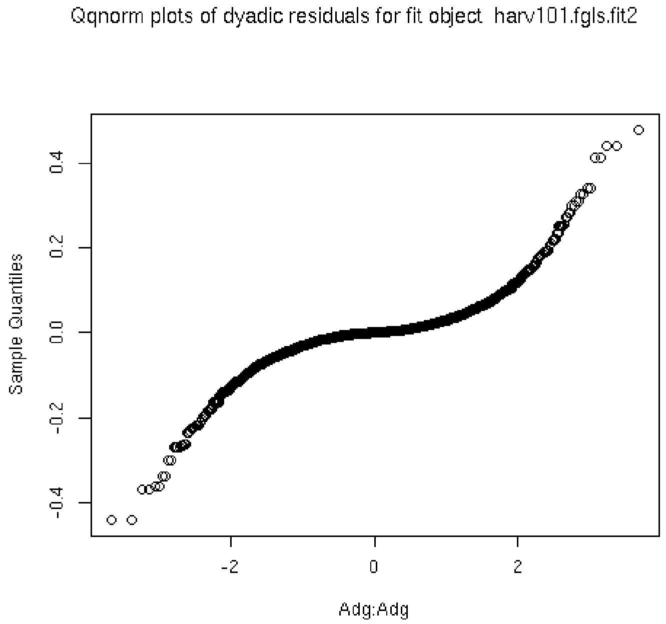

Figure 2: A qqplot of dyadic residuals for the Harvey dataset with two components fitted usng "fgls" method

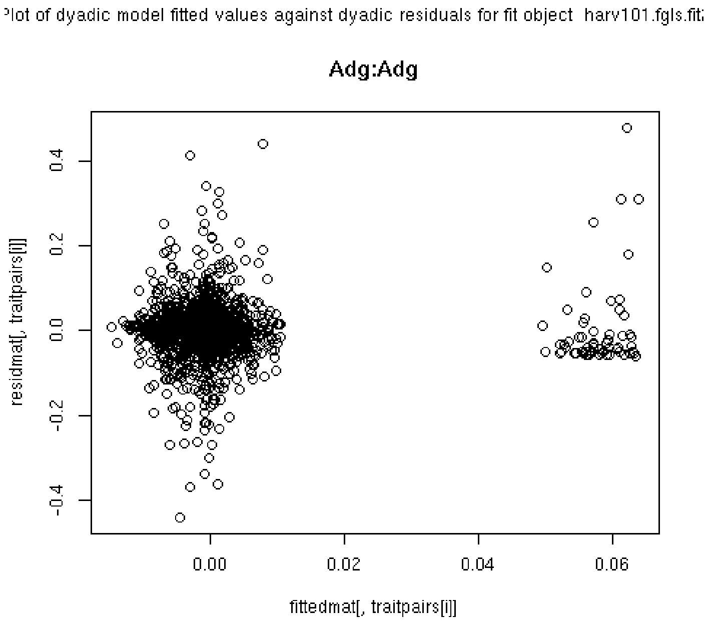

Figure 3: A plot of dyadic model fitted values against dyadic residuals for Trait ADG of Harvey dataset with two components fitted usng "fgls" method

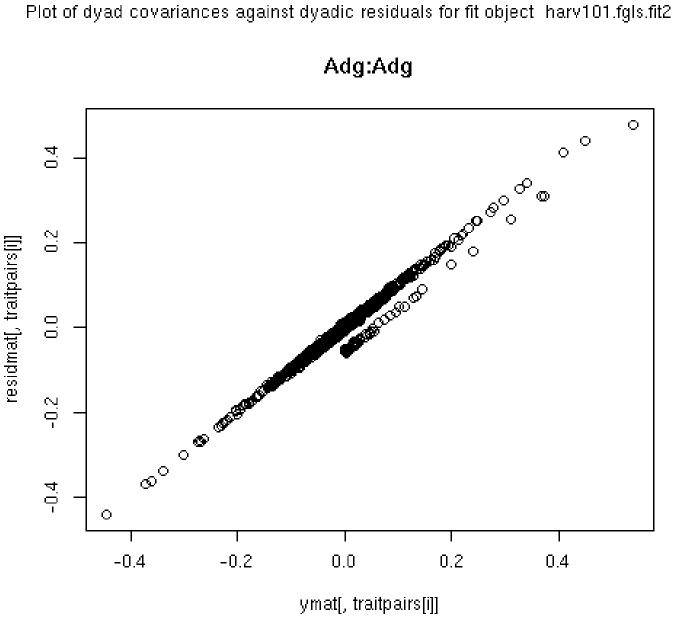

Figure 4: A plot of dyadic model observed values (ie covariances) against dyadic residuals for Trait ADG of Harvey dataset with two components fitted usng "fgls" method

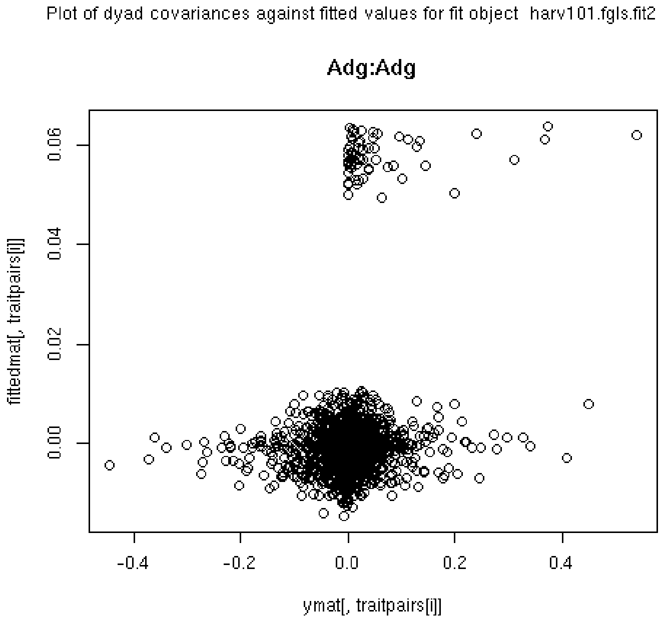

Figure 5: A plot of dyadic model observed values against dyadic model fitted values for Trait ADG of Harvey dataset with two components fitted usng "fgls" method

5 The size of the dyadic residuals

The dyadic residuals $Delta$ from equation 1 are not the same thing as residuals from a monadic model such as $\boldsymbol{P} = \boldsymbol{G} + \boldsymbol{E}$, and their variances are not VarE(I). The dyadic model equates the product of the measured values of pairs of individuals to its expectation in terms of their relationship matrix elements. The dyadic model fit is a regression of those expectarions on products of pairs of values. So the dyadic model error is the errors of that regression fit. Obvioulsy a product of 2 values is a very poor estimate of a covariance. So the dyadic errors are large. The dyadic model is usually a very poor fit. We can see how poor from some examples.

Figure 6 shows relationship matrix elements plotted against products of data values for 65^2 pairs of animals from the harvey dataset.

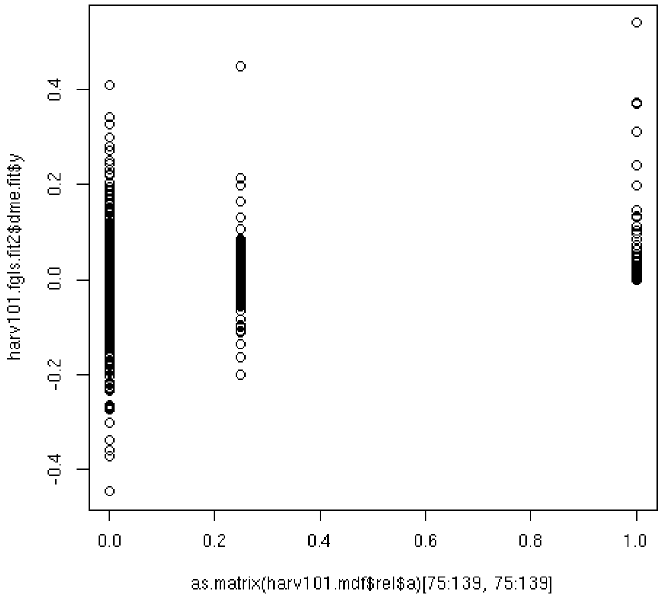

Figure 6: Plot of relationship matrix elements against product of ADG data values for pairs of individuals from the Harvey dataset

We see 3 groups of data points.. those with zero relationship those with a positive relationship coefficient of 0.25, and those with a relationship coefficientof 1.0 (ie pairs of individuals with self). The regresssion line in this plot is the estimator of the VarG(Ia) variance component (not quite it is a multiple regression - see below). We can see there is considerable variation around a regression line.

Figure 7 shows the same data with fitted values plotted instead of actual data values.

The regresssion line in this plot is the estimator of the VarG(Ia) variance component. This shows the variation left after the regression equation removes fitted variance.

Note these graphs only show the VarG(Ia) component. The actual fit is a multiple regression equation involving VarG(Ia) and VarE(I).

It is quite common for the variance accounted for by the fit to be only about 3 percent of the variance of dyadic residuals. This is why variance component estimation in quantitative genetics is a difficult task. It is like trying to detect a message with a signal to noise ratio of about 0.03.

6 Discussion

There is nothing special about the dyadic model used by *dmm()*. Quantitative genetics has always been about covariances between relatives and measures of relationship, that is about pairs of individuals. It is just not usually called a dyadic model, but that is what it is. The term is common in the social sciences where interactions between pairs of individuals are analysed with a dyadic model.

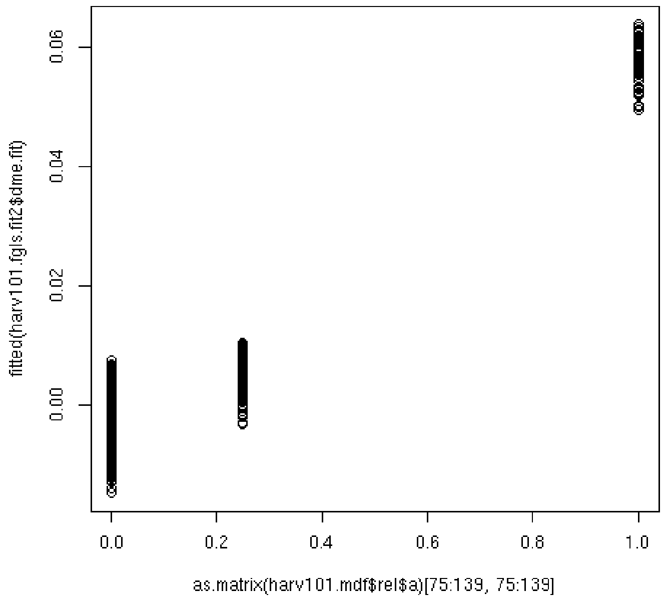

Figure 7: Plot of relationship matrix elements against fitted values for 65 pairs of individuals from Harvey dataset

What is important is what the dyadic model allows us to do, not the terminology. By turning variance component estimation into a regression problem, a dyadic model opens the door to using the wide range of established regression techniques for variance component estimation. That includes techniques for dealing with collinearities among the independent variables, and these could be quite useful in quantitative genetic applications where the variance components which we wish to estimate are often partially confounded, as in the example of Sectuion 4.1. There is a full presentation on the use of principal components regression in *dmmOverview.pdf* [2] Section 7.4. There are some issues, the interface to the "pcr" option via the *pls* package is clumsy and its use is seriously memory intensive. Some further work is indicated. The "fgls" option for solving the DME's does lead to REML estimates of (co)variances, but it is seriously memory intensive and only practical with example datasets of up to about 200 animals.

The dyadic residuals ($\boldsymbol{\Delta}$) are usually large and highly correlated with the observed dyadic covariances ($\boldsymbol{\Psi}$), as in Figure 4. This is because the covariance for each dyad is obtained from only one replicate pair of observations. The R^2 for a dyadic model is tiny - only 3 percent of the variance in the case of the *warcolak* example above. This highlights the central problem of quantitative genetic analysis - it is trying to extract information from a system with a signal to noise ration of 0.03. Modelling variances is much more demanding than modelling observations. It does not matter whether you do it by maximizing a likelihood, fitting a regression, or doing an AOV. The problem is instinsic - *dmm()* just makes it obvious by attacking the problem directly.

References

[1] Anderson R.D. (1978) Studies on the esatimation of variance components. PhD Thesis, Cornell University, Ithaca, New York.

[2] Jackson, N. (2015) An Overview of the R package dmm. From http://cran.r-project.org/package=dmm Or https://github.com/cran/dmm

[3] Jackson, N. (2024) Error covariance structure for the dyadic model used in *dmm()* From http://cran.r-project.org/package=dmm Or https://github.com/cran/dmm

[4] Patterson H.D. and Thompson R. (1974) Maximum likelihood estimation of components of variance. Proc. Eighth Internat. Biom. Conf., 197-209

[5] Rao, C.R. (1971) Estimation of variance and covariance components – MINQUE theory. J. Multivariate Analysis 1:257-275

[6] Searle, S.R., Casella, G., and McCullock, C.E. (1992) Variance Components. John Wiley and Sons, New York.

[7] Wolak, M.E. (2014) nadiv: an R package to create relatedness matrices for estimating non-additive genetic variances in animal models. Methods in Ecology and Evolution 3:792-796.

[8] Wolak, M.E. (2024) gremlin: Mixed effects REMP incorporating generalized inverses From https://CRAN.R-project.org/package=gremlin