An Overview of the R Package dmm

Neville Jackson

3 Jun 2020
For dmm_2.1-6

Contents

1 Acknowledgement

I would like to thank those programmers, many anonymous, who have made their work publicly available in the R environment. This package builds on their work, and its development would not have been feasable without their efforts.

2 Abstract

Package *dmm* estimates a variety of genetic and environmental (co)variance components from pedigree data. It can transform these to genetic parameters, and to estimates of response to selection. The statistical method used is an heirarchical model with the first level a normal fixed effects model and the second level a dyadic model or derived model which equates dyadic covariances to their expectations. The dyadic model is linear and therefore reduces variance component estimation to a linear regression problem. The package uses standard regression techniques (qr or lm functions) to estimate components as regression coefficients, and also offers robust regression (lmrob function) and principal component regression (pls package) as experimental alternative options. Estimating variance components in this way is feasable only for datasets of less than around 10000 individuals. The package may therefore be useful for modest sized research datasets, but not for extensive field data. The variance component estimates obtained by directly solving the dyadic model equations in the above way are equivalent to MINQUE estimates if the fixed model is fitted with OLS, and are equivalent to bias-corrected ML estimates if the fixed model is fitted with GLS. The package includes a number of test data sets which are useful to demonstrate correctness of its calculations, and to illustrate its capabilities.

3 Introduction

Quantitative genetic analysis uses what we can infer from a pedigree about the genetic relationship between pairs of individuals, to analyse variation among a sample of individuals in metric traits, and to make inferences regarding the causes of variation in the population from which individuals have been sampled. In *dmm()* pairs of individuals are called *dyads*, and a dyadic model is used for variance component estimation. The *dmm* package sets up dyadic model equations (DME's) from relationship matrices and observations. DME's equate obserations on a dyad to their expectation (in terms of genetic and environmental (co)variance components). Estimates of (co)variance components are derived by solving the DME's directly (eg using the QR method). *dmm()* does not form Henderson's mixed model equations, and does not require inverses of relationship matrices. *dmm()* then computes genetic parameters (ie proportions of variance and correlations), from the estimated variance components. There is also a facility to post-compute genetic response to selection.

The approach taken by *dmm()* is suited to small multi-trait datasets; typically research data rather than field data, of less than around 10000 individuals. *dmm()* is less compute intensive than iterative likelihood-based techniques, such as REML, and extends easily to multiple traits and multiple (co)variance components, but has a high computer memory requirement.

Research datasets are likely to have a pedigree design which permits estimation of individual and maternal components for additive, non-additive, or sex-linked genetic (co)variances. *dmm()* allows estimation of any or all of the above genetic components, including all possible cross-effect genetic covariances. The number of components estimated has little effect on compute intensity. *dmm()* includes an experimental approach to dealing with multicollinearities among the variance components, using principal component regression.

Research datasets may also have one or more environmental factors. *dmm()* allows the usual estimation of fixed effects in a mixed model, either by OLS or GLS. *dmm()* also allows a specific environmental factor called *cohort* to be defined, and allows *cohort* to be fitted as an environmental variance component, and thus to be part of phenotypic variance. The *cohort* concept represents a different approach to the issue of 'common environmental variance'.

The pro's for *dmm()* are that it is simple (has a low compute requirement), flexible (will fit any combination of variance components), multi-trait (will handle more traits without increased complexity), and yields estimates which are unbiased and the same as *aov* estimates for balanced designs. Because *dmm()* uses standard regression techniques for variance component estimation, it has access to well known and tested methods for obtaining standard errors.

The con's for *dmm()* are that it is very demanding of computer memory (not suited to large datasets), and yields estimates which are MINQUE or bias-corrected-ML (depending on whether fixed effects are fitted by OLS or GLS respectively). These will only be the same as REML estimates for balanced designs, although they are likely to be similar if the degree of unbalance is not severe.

4 Getting started with dmm

Before using the *dmm* package in an R session its library must be loaded with the statement

```
> library(dmm)
```

If you get the message

```
Error in library(dmm) : there is no package called dmm
```

this indicates that the *dmm* package has not been installed on your system. You need to look at the instructions for downloading and installing an R package. Try the manual *R Installation and Administration* which deals with add-on packages in Chapter 6, and is located at any CRAN mirror site (start at *http://www.r-project.org* and choose CRAN, then choose a mirror site, then choose Packages).

To use dmm one first must put the dataset to be analysed into an R workspace as a dataframe object. The minimum requirement is for a dataframe with columns labelled :

Id Identifier for each individual

SId Identifier for the sire of each individual

DId Identifier for the dam of each individual

Sex Sex code for each individual

Fixed factors Codes for levels of each fixed factor

Observations Numeric values for each observation or trait

So we start by having a look at an example dataframe. We shall use the dataset *sheep.df* which is part of the *dmm* package. So load the dataset

```
> data(sheep.df)
```

and get an overview of its contents using the *str()* function

```
> str(sheep.df)
'data.frame': 42 obs. of  9 variables:
 $ Id  : Factor w/ 42 levels "0a4441","0a4712",..: 39 3 5 9 10 41 42 40 2 4 ...
 $ SId : Factor w/ 5 levels "0a4721","1a4123",..: NA NA NA NA NA NA NA NA NA NA ...
 $ DId : Factor w/ 15 levels "0a4712","0a4713",..: NA NA NA NA NA NA NA NA NA NA ...
 $ Year: int  1981 1982 1983 1983 1984 1981 1981 1981 1982 1982 ...
 $ Tb  : Factor w/ 2 levels "S","T": 1 1 1 2 1 1 1 2 1 1 ...
 $ Sex : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 1 1 1 1 ...
 $ Cww : num  NA NA NA 4.2 4.7 4.1 4.4 3.8 5.1 4.9 ...
 $ Diam: num  NA NA NA 21.7 21.1 20 21.6 20.1 22 21.1 ...
 $ Bwt : num  NA NA NA 50 45 51 53 43 45 48 ...
>
```

So it has the columns Id,SId,DId,Sex noted above and it has Year and Tb which are potential fixed effects, and it has three traits Cww,Diam,Bwt.

Another way of getting an overview is to print the first few rows

```
> sheep.df[1:5,]
      Id  SId  DId Year Tb Sex Cww Diam Bwt
1 9a4003 <NA> <NA> 1981  S   M  NA   NA  NA
2 0a4721 <NA> <NA> 1982  S   M  NA   NA  NA
3 1a4123 <NA> <NA> 1983  S   M  NA   NA  NA
4 1a4371 <NA> <NA> 1983  T   M 4.2 21.7  50
5 2a4127 <NA> <NA> 1984  S   M 4.7 21.1  45
>
```

The first few individuals have unknown parents, coded as *NA*.

There are two immediately obvious problems with the way this dataset has been setup

- The pedigree columns (Id,SId,DId) contain alphanumeric codes. For *dmm* the codes must be numeric and the Id code must be sequential.

- The Year column is not a factor. If we want to use it as a fixed effect with discrete levels, it should be a factor.

So we first need to use the *mdf()* function to convert this dataframe to a suitable format. We may also wish to use the *mdf()* function to setup relationship matrices required to estimate various genetic (co)variance components: it depends on our genetic model - if we just require the additive genetic relationship matrix, *dmm()* will calculate this 'on the fly', but if we require non-additive or sex-linked relationship matrices, these must be pre-calculated with function *mdf()*. In the present example we will use just the additive relationship matrix and let *dmm()* calculate it *inline*.

To learn to use *mdf()* we need to look at its help page

```
> help(mdf)
   ...   help page should appear on screen here
>
```

We need to define the pedigree columns with the *pedcols* argument, define Sex, Year and Tb as factors with the *factorcols* argument, and define the traits with the ycols argument. We also need to say how Sex is coded, using the *sexcode* argument. We do not want to make relationship matrices , so the *relmat* argument is left NULL.

```
> sheep.mdf <- mdf(sheep.df,pedcols=c(1:3),factorcols=c(4:6),
                   ycols=c(7:9),sexcode=c("M","F"))
Pedigree Id check:
No of rows with Id in original dataframe =  42
No of sex codes not in sexcode[] so changed to NA =  0
```

```
No of rows with Sex == NA removed from dataframe =   0
No of rows with Id == NA removed from dataframe =   0
No of rows with duplicated Id removed from dataframe =   0
No of rows remaining after duplicates and NA's removed =   42
No of DId's with no matching Id =   2
Length of dataframe with base Id's added =   44
Renumber pedigree Id's:
Add matrix of multivariate traits:
Return mdf as a normal dataframe:

> str(sheep.mdf)
'data.frame': 44 obs. of  7 variables:
 $ Id  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ SId : int  NA NA NA NA NA NA NA NA NA NA ...
 $ DId : int  NA NA NA NA NA NA NA NA NA NA ...
 $ Year: Factor w/ 8 levels "1981","1982",..: NA NA 1 2 3 3 4 1 1 1 ...
 $ Tb  : Factor w/ 2 levels "S","T": NA NA 1 1 1 2 1 1 1 2 ...
 $ Sex : Factor w/ 2 levels "F","M": 1 1 2 2 2 2 2 2 1 1 1 ...
 $ Ymat: num [1:44, 1:3] NA NA NA NA NA 4.2 4.7 4.1 4.4 3.8 ...
   ..- attr(*, "dimnames")=List of 2
   .. ..$ : chr  "0a4713" "2a4247" "9a4003" "0a4721" ...
   .. ..$ : chr  "Cww" "Diam" "Bwt"
>
```

So now the pedigree columns (Id, SId, DId) are coded as integers, Year is a factor, and the three traits (Cww, Diam, Bwt) are assembled into a matrix called Ymat. Also note that there are now 44 observations (originally there were 42); two 'base' individuals have been added because there were two DId's which did not occur as an Id.

We are now ready to use function $dmm()$. We wish to adjust for the fixed effects by first fitting a fixed model of the form

$$Y_{ijkl} = \mu + Sex_i + Year_j + Tb_k + r_{ijkl}$$

where Y_{ijkl} is an observation of one trait on the lth individual and r_{ijkl} is the corresponding residual.

Assume that we just wish to partition r_{ijkl} into an individual environmental variance and an individual genetic additive variance. To do this we fit a dyadic model of the form

$$Cov(r_{ijkl,i^{`}j^{`}k^{`}l^{`}}) = e_{ll^{`}}\sigma^2_{E(I)} + a_{ll^{`}}\sigma^2_{G(Ia)} + \delta_{ijkl,i^{`}j^{`}k^{`}l^{`}}$$

where $Cov(r_{ijkl,i^{`}j^{`}k^{`}l^{`}})$ is the dyadic observation for dyad $(l, l^{`})$ and $\delta_{ijkl,i^{`}j^{`}k^{`}l^{`}}$ is the corresponding dyadic residual.

These are single trait models, but extension to multi-trait case is trivial. In dmm the partitioned variance components $\sigma^2_{E(I)}$ and $\sigma^2_{G(Ia)}$ are labelled $VarE(I)$ and $VarG(Ia)$ respectively, even when they may refer to cross-trait covariances.

8

The notation used by *dmm* to label (co)variance components was designed to use only ASCII symbols. A few examples will make the meaning clear

VarE(I) variance environmental individual

VarG(Ia) variance genetic individual additive

VarG(Ma) variance genetic maternal additive

CovE(I,M) covariance environmental individual x maternal

VarGs(Ia) variance genetic sexlinked individual additive

There is a complete definition of all components in Section 6.

So we now call function *dmm()* to do the above model fitting, and save its output for later display. To learn to use *dmm()* look at its help page

```
>help(dmm)
 ...    help page should appear on screen here
```

We need to give the name of the dataframe, the fixed model with the *fixform* argument, and the variances to be partitioned with the *components* argument. The components argument in this case is actually the default, but we shall specify it for clarity. Note that *dmm()* is different from most other variance component programs in that it requires the individual environmental variance ("VarE(I)") to be explicitly fitted.

```
>sheep.fit1 <- dmm(sheep.mdf, Ymat ~ 1 + Sex + Year + Tb,
                   components = c("VarE(I)","VarG(Ia)"))
Dyadic mixed model fit for datafile: sheep.mdf
Data file is a normal dataframe:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 2
OLS-b step:
no of fixed effect df (k) =  9
no of traits (l) =  3
Setup antemodel matrices:
no of individuals in pedigree (m) =  44
no of individuals with data and X codes (n) =  36
Rank of X: 9   No of Fixed Effects: 9
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
>
```

So what are all these lines displayed on the screen? Well most are just reporting the stage of computation; a large problem can take some time and it

helps to know the stage reached. This is particularly important if the iterative GLS-b step is used.

If the fixed effect model is singular the process will stop with a message

```
"Rank of X .ne. k:"
```

If the dyadic model equations are singular the process will stop with a message

```
"Dyadic model equations not of full rank:
   either omit some components or try dmeopt='pcr'"
```

Even if there are no singularities, the dyadic model equations may have serious collinearities. We can check this by looking at the *dme.correl* attribute of the *sheep.fit1* object.

```
> sheep.fit1$dme.correl
          VarE(I)   VarG(Ia)
VarE(I)   1.0000000 0.8125828
VarG(Ia)  0.8125828 1.0000000
>
```

In this case the correlation between "VarE(I)" and "VarG(Ia)" is 0.81 which should sound warning bells. If this were a serious analysis it would be best to obtain some more data with a structure which would better separate the two components of interest. As this is only a tutorial, we shall proceed to look at estimates of fixed effects and variance components. The object *sheep.fit1* returned by *dmm()* is an object of class *dmm* and there is a print method for this class which gives a brief view of the 'fit' object as follows:

```
> print(sheep.fit1)  # or just sheep.fit1 will imply print
Call:
dmm.default(mdf = sheep.mdf, fixform = Ymat ~ 1 + Sex + Year +
    Tb, components = c("VarE(I)", "VarG(Ia)"))
Fixed formula:
Ymat ~ 1 + Sex + Year + Tb
Cohort formula:
NULL
Var/Covariance components:
[1] "VarE(I)"  "VarG(Ia)" "VarP(I)"
Traits:
[1] "Cww"  "Diam" "Bwt"
Fitted OLS fixed effects:
                   Cww         Diam       Bwt
(Intercept) 4.08994992 20.68664709 49.706459
SexM        0.21661950  0.30933131  3.978543
Year1982    0.76666667  0.73333333 -1.666667
Year1983    0.03328253 -0.26902396 -3.111563
```

```
Year1984     0.38164017 -0.38468523 -5.782813
Year1985     0.63093009  0.03065326 -3.203542
Year1986     0.95145594  0.85505128 -4.948870
Year1987     0.46112683  0.18022290 -8.992848
TbT          0.03015024 -0.35994127 -2.119376
Var/covariance components partitioned by DME after OLS fit:
            Cww:Cww    Cww:Diam    Cww:Bwt    Diam:Cww Diam:Diam Diam:Bwt
VarE(I)   0.02624452 -0.04600793 0.2737025 -0.04600793 0.2670173 1.166502
VarG(Ia) 0.35379492  0.46601661 1.7435379  0.46601661 0.7903752 2.231398
          Bwt:Cww Bwt:Diam   Bwt:Bwt
VarE(I)   0.2737025 1.166502 17.397968
VarG(Ia) 1.7435379 2.231398  8.616402
Observed (residual) var/covariance after OLS fit:
           Cww        Diam        Bwt
Cww   0.2219876 0.3489707   1.824223
Diam 0.3489707 0.9569698   3.134368
Bwt  1.8242230 3.1343678  24.994537
>
```

In the above view, there are no standard errors, and we always get all fixed
effects across all traits, and all components across all traitpairs. The matrix
labelled 'Observed (residual) var/covariance after OLS fit:' is covariance ma-
trix the residual terms r_{ijkl} from the fitted fixed model. It is *not* the pheno-
typic covariance matrix in most cases, because the individuals are related. The
print.dmm method is documented on its help page:

```
>help(print.dmm)
 ...
```

A more extensive view, with standard errors and confidence limits is given
by the summary method:

```
> summary(sheep.fit1)
Call:
summary.dmm(object = sheep.fit1)


Coefficients fitted by OLS for fixed effects:

            Trait Estimate StdErr  CI95lo CI95hi
(Intercept)   Cww   4.0899  0.278  3.5451  4.635
SexM          Cww   0.2166  0.186 -0.1482  0.581
Year1982      Cww   0.7667  0.385  0.0127  1.521
Year1983      Cww   0.0333  0.368 -0.6878  0.754
Year1984      Cww   0.3816  0.347 -0.2991  1.062
Year1985      Cww   0.6309  0.331 -0.0170  1.279
Year1986      Cww   0.9515  0.335  0.2943  1.609
Year1987      Cww   0.4611  0.339 -0.2031  1.125
```

```
TbT            Cww   0.0302  0.172 -0.3066  0.367

            Trait Estimate StdErr CI95lo CI95hi
(Intercept)  Diam  20.6866  0.577 19.555 21.818
SexM         Diam   0.3093  0.386 -0.448  1.067
Year1982     Diam   0.7333  0.799 -0.832  2.299
Year1983     Diam  -0.2690  0.764 -1.766  1.228
Year1984     Diam  -0.3847  0.721 -1.798  1.029
Year1985     Diam   0.0307  0.686 -1.315  1.376
Year1986     Diam   0.8551  0.696 -0.509  2.220
Year1987     Diam   0.1802  0.704 -1.199  1.559
TbT          Diam  -0.3599  0.357 -1.059  0.339

            Trait Estimate StdErr  CI95lo CI95hi
(Intercept)  Bwt   49.71   2.95   43.925  55.49
SexM         Bwt    3.98   1.97    0.108   7.85
Year1982     Bwt   -1.67   4.08   -9.667   6.33
Year1983     Bwt   -3.11   3.90  -10.763   4.54
Year1984     Bwt   -5.78   3.69  -13.006   1.44
Year1985     Bwt   -3.20   3.51  -10.078   3.67
Year1986     Bwt   -4.95   3.56  -11.922   2.02
Year1987     Bwt   -8.99   3.60  -16.041  -1.94
TbT          Bwt   -2.12   1.82   -5.693   1.45


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

          Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)    Cww:Cww   0.0262 0.0533 -0.0782  0.131
VarG(Ia)   Cww:Cww   0.3538 0.0491  0.2576  0.450
VarP(I)    Cww:Cww   0.3800 0.0316  0.3181  0.442

          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Cww:Diam  -0.046 0.1121 -0.266  0.174
VarG(Ia)   Cww:Diam   0.466 0.1033  0.264  0.668
VarP(I)    Cww:Diam   0.420 0.0665  0.290  0.550

          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Cww:Bwt    0.274  0.575 -0.853   1.40
VarG(Ia)   Cww:Bwt    1.744  0.530  0.705   2.78
VarP(I)    Cww:Bwt    2.017  0.341  1.349   2.69

          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Diam:Cww  -0.046 0.1121 -0.266  0.174
VarG(Ia)   Diam:Cww   0.466 0.1033  0.264  0.668
VarP(I)    Diam:Cww   0.420 0.0665  0.290  0.550
```

```
          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Diam:Diam    0.267  0.232 -0.187  0.721
VarG(Ia)  Diam:Diam    0.790  0.213  0.372  1.208
VarP(I)   Diam:Diam    1.057  0.137  0.788  1.326

          Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)    Diam:Bwt     1.17  1.201 -1.1874   3.52
VarG(Ia)   Diam:Bwt     2.23  1.107  0.0625   4.40
VarP(I)    Diam:Bwt     3.40  0.712  2.0023   4.79

          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)     Bwt:Cww    0.274  0.575 -0.853   1.40
VarG(Ia)    Bwt:Cww    1.744  0.530  0.705   2.78
VarP(I)     Bwt:Cww    2.017  0.341  1.349   2.69

          Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)    Bwt:Diam     1.17  1.201 -1.1874   3.52
VarG(Ia)   Bwt:Diam     2.23  1.107  0.0625   4.40
VarP(I)    Bwt:Diam     3.40  0.712  2.0023   4.79

          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)     Bwt:Bwt    17.40   6.07   5.49   29.3
VarG(Ia)    Bwt:Bwt     8.62   5.60  -2.35   19.6
VarP(I)     Bwt:Bwt    26.01   3.60  18.96   33.1

>
```

So now we get one parameter estimate per line with standard errors and confidence limits.With lots of traits and parameters this can get to be rather voluminous, so the summary method for *dmm* objects has arguments for choosing subsets of traits or components. There is also an argument to change the ordering from bytrait to byparameter. Consult the help page as follows

```
>help(summary.dmm)
 ... help page should appear here
```

We may wish to view genetic parameter estimates instead of variance components. In this case there is a gprint.dmm method which produces an abbreviated output (analagous to print.dmm) and a gsummary.dmm method which produces a full output with standard errors (analagous to summary.dmm). These functions have help pages

```
>help(gprint.dmm)
 ... help page should appear here
>help(gsummary.dmm)
 ... help page should appear here
```

In this case we shall view the gsummary output for just 2 traits

```
>gsummary(sheep.fit1,traitset=c("Diam","Bwt"))
Call:
gsummary.dmm(dmmobj = sheep.fit1, traitset = c("Diam", "Bwt"))

Proportion of phenotypic var/covariance partitioned by DME:
 to each component (OLS-b):

         Trait Estimate StdErr CI95lo CI95hi
VarE(I)   Diam    0.253  0.199 -0.138  0.643
VarG(Ia)  Diam    0.747  0.207  0.341  1.154
VarP(I)   Diam    1.000  0.000  1.000  1.000


         Trait Estimate StdErr  CI95lo CI95hi
VarE(I)    Bwt    0.669  0.209  0.2586  1.079
VarG(Ia)   Bwt    0.331  0.210 -0.0806  0.743
VarP(I)    Bwt    1.000  0.000  1.0000  1.000



Correlation corresponding to each var/covariance component:
 partitioned by DME (OLS-b):

         Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)  Diam:Diam        1      0      1      1
VarG(Ia) Diam:Diam        1      0      1      1
VarP(I)  Diam:Diam        1      0      1      1


         Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)   Diam:Bwt    0.541 0.3166 -0.0794  1.162
VarG(Ia)  Diam:Bwt    0.855 0.2646  0.3365  1.374
VarP(I)   Diam:Bwt    0.648 0.0923  0.4669  0.829


         Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)   Bwt:Diam    0.541 0.3166 -0.0794  1.162
VarG(Ia)  Bwt:Diam    0.855 0.2646  0.3365  1.374
VarP(I)   Bwt:Diam    0.648 0.0923  0.4669  0.829


         Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Bwt:Bwt        1      0      1      1
VarG(Ia)   Bwt:Bwt        1      0      1      1
VarP(I)    Bwt:Bwt        1      0      1      1



Phenotypic var/covariance from components partitioned by DME (OLS-b):
```

```
   Traitpair Estimate StdErr CI95lo CI95hi
1 Diam:Diam     1.06  0.137  0.788   1.33
2  Diam:Bwt     3.40  0.712  2.002   4.79
3  Bwt:Diam     3.40  0.712  2.002   4.79
4   Bwt:Bwt    26.01  3.601 18.957  33.07

>
```

The section labelled 'Proportion of phenotypic var/covariance ....' gives for each trait an estimate of each variance component as a proportion of "VarP(I)", the phenotpic variance. So for component "VarG(Ia)" the proportion 0.747 is the additive genetic heritability estimate for trait "Diam".

The section labelled 'Correlation ...' gives for each trait pair an estimate of the cross-trait correlation corresponding to each variance component. So for component "VarG(Ia)" the estimate 0.855 is the additive genetic correlation for traitpair "Diam:Bwt".

Note that the confidence limits are not constrained to the bounds of a proportion or a correlation. This will not usually be a problem with reasonable sized datasets.

Let us now change the dyadic model to include maternal as well as individual genetic and environmental effects. This is done by simply adding components "VarE(M)" and "VarG(Ma)" to the *components* argument, representing maternal environmental and maternal additive genetic effects. However we also need to add some covariances, because it is possible for individual and maternal effects to be correlated at both the environmental and genetic levels. So we add "CovE(I,M)" and "CovG(Ia,Ma)", and we also add their reciprocals "CovE(M,I)" and "CovG(Ma,Ia)". It is a useful convention in *dmm()* to always include reciprocal covariances, for two reasons

- It makes the variances and covariances sum correctly to phenotypic variance

- It allows for cross-trait-cross-effect covariances to differ between reciprocals, which is commonly the case

So let us do another fit of the new model including maternal effects. We will leave out the environmental covariances, but include the genetic ones. We still only need the additive relationship matrix so we can use the same dataframe.

```
>sheep.fit2 <- dmm(sheep.mdf, Ymat ~ 1 + Sex + Year + Tb,
        components=c("VarE(I)","VarG(Ia)","VarE(M)","VarG(Ma)",
        "CovG(Ia,Ma)","CovG(Ma,Ia)"))
 ...
```

Again we look at the column correlations of the dyadic model equations

```
> sheep.fit2$dme.corre
            VarE(I)  VarG(Ia)  VarE(M)  VarG(Ma) CovG(Ia,Ma) CovG(Ma,Ia)
```

```
VarE(I)       1.0000000 0.8125828 0.5302941 0.5074266   0.3504908   0.3504908
VarG(Ia)      0.8125828 1.0000000 0.6238947 0.6397454   0.6573003   0.6573003
VarE(M)       0.5302941 0.6238947 1.0000000 0.9646691   0.6613528   0.6613528
VarG(Ma)      0.5074266 0.6397454 0.9646691 1.0000000   0.7067437   0.7067437
CovG(Ia,Ma) 0.3504908 0.6573003 0.6613528 0.7067437   1.0000000   0.5009414
CovG(Ma,Ia) 0.3504908 0.6573003 0.6613528 0.7067437   0.5009414   1.0000000
>
```

For a small dataset, the column correlations are reasonable, except for the 0.96 between "VarE(M)" and "VarG(Ma)" and the 0.81 which was there in the previous analysis. We will just view the genetic parameters for two traits

```
>gsummary(sheep.fit2, traitset = c("Diam", "Cww"))
Call:
gsummary.dmm(dmmobj = sheep.fit2, traitset = c("Diam", "Cww"))

Proportion of phenotypic var/covariance partitioned by DME:
 to each component (OLS-b):


            Trait Estimate StdErr CI95lo CI95hi
VarE(I)      Diam   0.0411 0.0903 -0.136  0.218
VarG(Ia)     Diam   0.5381 0.1722  0.201  0.876
VarE(M)      Diam   0.0958 0.1427 -0.184  0.376
VarG(Ma)     Diam   1.1164 0.2163  0.693  1.540
CovG(Ia,Ma)  Diam  -0.3957 0.1266 -0.644 -0.148
CovG(Ma,Ia)  Diam  -0.3957 0.1266 -0.644 -0.148
VarP(I)      Diam   1.0000 0.0000  1.000  1.000


            Trait Estimate StdErr  CI95lo   CI95hi
VarE(I)       Cww   0.1851 0.0886  0.0114   0.3587
VarG(Ia)      Cww   0.5301 0.1613  0.2140   0.8461
VarE(M)       Cww   0.0475 0.1329 -0.2131   0.3081
VarG(Ma)      Cww   0.7445 0.1812  0.3893   1.0998
CovG(Ia,Ma)   Cww  -0.2536 0.1094 -0.4680  -0.0392
CovG(Ma,Ia)   Cww  -0.2536 0.1094 -0.4680  -0.0392
VarP(I)       Cww   1.0000 0.0000  1.0000   1.0000



Correlation corresponding to each var/covariance component:
 partitioned by DME (OLS-b):


            Traitpair Estimate    StdErr CI95lo CI95hi
VarE(I)     Diam:Diam    1.000 2.11e-08  1.000  1.000
VarG(Ia)    Diam:Diam    1.000 0.00e+00  1.000  1.000
VarE(M)     Diam:Diam    1.000 1.49e-08  1.000  1.000
VarG(Ma)    Diam:Diam    1.000 0.00e+00  1.000  1.000
CovG(Ia,Ma) Diam:Diam   -0.511 8.39e-02 -0.675 -0.346
```

16

```
CovG(Ma,Ia) Diam:Diam    -0.511 8.39e-02 -0.675 -0.346
VarP(I)     Diam:Diam     1.000 0.00e+00  1.000  1.000

            Traitpair Estimate    StdErr    CI95lo   CI95hi
VarE(I)      Diam:Cww   0.0229    0.0312   -0.0382    0.084
VarG(Ia)     Diam:Cww   0.9999    0.1283    0.7484    1.251
VarE(M)      Diam:Cww  -0.8569  253.5523 -497.8195  496.106
VarG(Ma)     Diam:Cww   0.9960    0.0900    0.8197    1.172
CovG(Ia,Ma)  Diam:Cww  -0.4762    0.1222   -0.7157   -0.237
CovG(Ma,Ia)  Diam:Cww  -0.5528    0.1008   -0.7505   -0.355
VarP(I)      Diam:Cww   0.6596    0.0537    0.5544    0.765

            Traitpair Estimate    StdErr    CI95lo   CI95hi
VarE(I)      Cww:Diam   0.0229    0.0312   -0.0382    0.084
VarG(Ia)     Cww:Diam   0.9999    0.1283    0.7484    1.251
VarE(M)      Cww:Diam  -0.8569  253.5523 -497.8195  496.106
VarG(Ma)     Cww:Diam   0.9960    0.0900    0.8197    1.172
CovG(Ia,Ma)  Cww:Diam  -0.5528    0.1008   -0.7505   -0.355
CovG(Ma,Ia)  Cww:Diam  -0.4762    0.1222   -0.7157   -0.237
VarP(I)      Cww:Diam   0.6596    0.0537    0.5544    0.765

            Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)      Cww:Cww   1.000   0.000   1.000    1.0
VarG(Ia)     Cww:Cww   1.000   0.000   1.000    1.0
VarE(M)      Cww:Cww   1.000   0.000   1.000    1.0
VarG(Ma)     Cww:Cww   1.000   0.000   1.000    1.0
CovG(Ia,Ma)  Cww:Cww  -0.404   0.104  -0.607   -0.2
CovG(Ma,Ia)  Cww:Cww  -0.404   0.104  -0.607   -0.2
VarP(I)      Cww:Cww   1.000   0.000   1.000    1.0


Phenotypic var/covariance from components partitioned by DME (OLS-b):

  Traitpair Estimate StdErr CI95lo CI95hi
1 Diam:Diam    3.003 0.2743  2.466   3.54
2  Diam:Cww    0.981 0.1330  0.720   1.24
3  Cww:Diam    0.981 0.1330  0.720   1.24
4   Cww:Cww    0.736 0.0633  0.612   0.86

>
```

Notice that the covariances between Ia and Ma are negative and contribute negatively to phenotypic variance, so that the proportions still sum to 1.0 and it is thus possible for the proportions due to some components to exceed 1.0. This is actually the case for "VarG(Ma)" for trait "Diam" where the proportion is 1.116.

Notice also that the correlations for a trait with itself are always 1.0, except for those components which are cross-effect covariances. For example the same-trait-cross-effect correlation between Ia and Ma for trait "Diam" is -0.511. This represents the genetic correlation between individual additive and maternal additive effects.

Notice also that the cross-trait-cross-effect correlations are not equal, as mentioned above. For example the correlation corresponding to "CovG(Ia,Ma)" for traits "Diam:Cww" is -0.476, while the reciprocal correlation corresponding to "CovG(Ma,Ia)" for traits "Diam:Cww" is -0.552.

Let us now assume that the above model is our final choice for these data. Up until now all $dmm()$ runs have defaulted to what we term $OLS\text{-}b$ estimates, that is the dyadic model equations are setup using ordinary least squares estimates of the fixed effects. This leads to variance component estimates which are equivalent to MINQUE estimates.

We may now wish, for the final model, to obtain the more desirable $GLS\text{-}b$ estimates, where the dyadic model equations are setup using generalised least squares estimates of the fixed effects. This leads to variance component estimates which are equivalent to 'bias-corrected ML' estimates. The disadvantage here is that the procedure becomes iterative, so it makes sense to reserve it for the final analysis. We simply add the $gls=T$ argument as follows

```
sheep.fit2g <- dmm(sheep.mdf, Ymat ~ 1 + Sex + Year ,
                   components=c("VarE(I)","VarG(Ia)","VarE(M)","VarG(Ma)",
                   "CovG(Ia,Ma)","CovG(Ma,Ia)"),gls=T)
 ...
OLS-b step completed:

GLS-b step:
Warning: Multivariate GLS is not same as multiple univariate GLS's
Round =  1  Stopcrit =  1.000393
Round =  2  Stopcrit =  6.171278
Round =  3  Stopcrit =  3.063188
Round =  4  Stopcrit =  3.975422
Round =  5  Stopcrit =  3.353479
Round =  6  Stopcrit =  4.5497
Round =  7  Stopcrit =  17.41195
Round =  8  Stopcrit =  17.14175
   .....
Round =  199  Stopcrit =  9.455716
Round =  200  Stopcrit =  4.414117
Iteration completed - count =   200
Failed to converge
GLS-b step abandoned:
>
```

So the $GLS\text{-}b$ step fails to converge. We cannot get $GLS\text{-}b$ estimates in this case. Not surprising given the small dataset and the collinearities among the

components. So let us go back to the simpler model and do *GLS-b* estimates
there:

```
> sheep.fitg <- dmm(sheep.mdf, Ymat ~ 1 + Sex + Year + Tb,
                    components = c("VarE(I)","VarG(Ia)"),gls=T)
 ...
OLS-b step completed:

GLS-b step:
Warning: Multivariate GLS is not same as multiple univariate GLS's
Round =  1  Stopcrit =  0.4485067
Round =  2  Stopcrit =  0.1680281
Round =  3  Stopcrit =  0.1077827
Round =  4  Stopcrit =  0.06148315
Round =  5  Stopcrit =  0.05731837
Round =  6  Stopcrit =  0.02106319
Round =  7  Stopcrit =  0.02599891
Round =  8  Stopcrit =  0.0157828
Round =  9  Stopcrit =  0.01001712
Round =  10  Stopcrit =  0.007371433
Iteration completed - count =  10
Convergence achieved
Components to genetic parameters and SE's:
GLS-b step completed successfully:
>
```

So in this case the iteration works and converges quite rapidly, despite the
high correlation between the two components. Note that this run does both
OLS-b and *GLS-b* estimates, in fact the *OLS-b* step provides intital estimates
of the variance components for the *GLS-b* step. The saved object *sheep.fitg*
contains the results of both steps. We can see the structure of *sheep.fitg* as
follows

```
>attributes(sheep.fitg)
$names
 [1] "aov"                    "mdf"                "fixform"
 [4] "b"                      "seb"                "vara"
 [7] "totn"                   "degf"               "dme.mean"
[10] "dme.var"                "dme.correl"         "dmeopt"
[13] "siga"                   "sesiga"             "vard"
[16] "degfd"                  "component"          "correlation"
[19] "correlation.variance"   "correlation.se"     "fraction"
[22] "fraction.variance"      "fraction.se"        "variance.components"
[25] "variance.components.se" "phenotypic.variance" "phenotypic.variance.se"
[28] "observed.variance"      "gls"                "call"

$class
```

```
[1] "dmm"

> attributes(sheep.fitg$gls)
$names
 [1] "b"                     "seb"                    "siga"
 [4] "sesiga"                "vard"                   "msr"
 [7] "msrdf"                 "msa"                    "component"
[10] "correlation"           "correlation.variance"   "correlation.se"
[13] "fraction"              "fraction.variance"      "fraction.se"
[16] "variance.components"   "variance.components.se" "phenotypic.variance"
[19] "phenotypic.variance.se" "observed.variance"     "dmeopt"

>
```

So *sheep.fitg* now contains results of the *OLS-b* step plus an attribute "gls" which itself contains results of the *GLS-b* step.

All the S3 methods (print(),summary(),gprint(),gsummary()) will report the *GLS-b* results as well as the *OLS-b* results, if given the argument *gls=T*. For example, let us view the variance components and fixed effect estimates as follows

```
> summary(sheep.fitg,gls=T,traitset=c("Cww","Diam"))
Call:
summary.dmm(object = sheep.fitg, traitset = c("Cww", "Diam"),
    gls = T)

Coefficients fitted by OLS for fixed effects:

            Trait Estimate StdErr  CI95lo CI95hi
(Intercept)  Cww    4.0899  0.278  3.5451  4.635
SexM         Cww    0.2166  0.186 -0.1482  0.581
Year1982     Cww    0.7667  0.385  0.0127  1.521
Year1983     Cww    0.0333  0.368 -0.6878  0.754
Year1984     Cww    0.3816  0.347 -0.2991  1.062
Year1985     Cww    0.6309  0.331 -0.0170  1.279
Year1986     Cww    0.9515  0.335  0.2943  1.609
Year1987     Cww    0.4611  0.339 -0.2031  1.125
TbT          Cww    0.0302  0.172 -0.3066  0.367


            Trait Estimate StdErr CI95lo CI95hi
(Intercept)  Diam  20.6866  0.577 19.555 21.818
SexM         Diam   0.3093  0.386 -0.448  1.067
Year1982     Diam   0.7333  0.799 -0.832  2.299
Year1983     Diam  -0.2690  0.764 -1.766  1.228
Year1984     Diam  -0.3847  0.721 -1.798  1.029
Year1985     Diam   0.0307  0.686 -1.315  1.376
```

```
Year1986    Diam   0.8551  0.696 -0.509  2.220
Year1987    Diam   0.1802  0.704 -1.199  1.559
TbT         Diam  -0.3599  0.357 -1.059  0.339


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

          Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)    Cww:Cww   0.0262 0.0533 -0.0782  0.131
VarG(Ia)   Cww:Cww   0.3538 0.0491  0.2576  0.450
VarP(I)    Cww:Cww   0.3800 0.0316  0.3181  0.442


          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Cww:Diam  -0.046 0.1121 -0.266  0.174
VarG(Ia)   Cww:Diam   0.466 0.1033  0.264  0.668
VarP(I)    Cww:Diam   0.420 0.0665  0.290  0.550


          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Diam:Cww  -0.046 0.1121 -0.266  0.174
VarG(Ia)   Diam:Cww   0.466 0.1033  0.264  0.668
VarP(I)    Diam:Cww   0.420 0.0665  0.290  0.550


          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Diam:Diam   0.267  0.232 -0.187  0.721
VarG(Ia)  Diam:Diam   0.790  0.213  0.372  1.208
VarP(I)   Diam:Diam   1.057  0.137  0.788  1.326



Coefficients fitted by GLS for fixed effects:

            Trait Estimate StdErr CI95lo CI95hi
(Intercept)  Cww    4.363  0.549  3.288  5.439
SexM         Cww    0.232  0.316 -0.388  0.852
Year1982     Cww    0.354  0.758 -1.131  1.840
Year1983     Cww   -0.414  0.706 -1.798  0.969
Year1984     Cww    0.716  0.621 -0.502  1.934
Year1985     Cww    0.606  0.614 -0.597  1.810
Year1986     Cww    0.405  0.652 -0.874  1.683
Year1987     Cww    0.197  0.654 -1.084  1.478
TbT          Cww    0.101  0.308 -0.502  0.705

            Trait Estimate StdErr CI95lo CI95hi
(Intercept) Diam  21.1631  0.915 19.370 22.956
SexM        Diam   0.3396  0.521 -0.681  1.361
Year1982    Diam   0.0224  1.263 -2.453  2.498
Year1983    Diam  -1.0643  1.174 -3.366  1.237
```

```
Year1984    Diam   0.1941  1.028 -1.821  2.209
Year1985    Diam  -0.0109  1.019 -2.008  1.987
Year1986    Diam  -0.1152  1.086 -2.244  2.014
Year1987    Diam  -0.2972  1.087 -2.428  1.834
TbT         Diam  -0.2187  0.509 -1.216  0.779


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

          Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)    Cww:Cww    0.0661 0.0457 -0.0234  0.156
VarG(Ia)   Cww:Cww    0.9807 0.0398  0.9027  1.059
VarP(I)    Cww:Cww    1.0468 0.0226  1.0025  1.091

          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Cww:Diam   -0.0429 0.0899 -0.219  0.133
VarG(Ia)  Cww:Diam    1.6542 0.0783  1.501  1.808
VarP(I)   Cww:Diam    1.6113 0.0445  1.524  1.699

          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Diam:Cww   -0.0429 0.0899 -0.219  0.133
VarG(Ia)  Diam:Cww    1.6542 0.0783  1.501  1.808
VarP(I)   Diam:Cww    1.6113 0.0445  1.524  1.699

          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Diam:Diam   0.0389 0.1757 -0.305  0.383
VarG(Ia)  Diam:Diam   2.8223 0.1530  2.522  3.122
VarP(I)   Diam:Diam   2.8612 0.0871  2.691  3.032

>
```

Notice that the *GLS-b* estimates differ from the *OLS-b* estimates, both for fixed effects and for variance components, and that the *GLS-b* estimates generally have smaller standard errors. That is as would be expected.

That is enough for a tutorial. Here is a list of further aspects which the user may wish to investigate

- Non additive genetic (co)variance components

- Sexlinked genetic variance components

- Calculation of response to selection

- Plotting the dyadic model residuals

- Using alternative regression methods to solve the dyadic model equations. Options are robust regression and principal components regression

- Defining a cohort effect

5 Some datasets with 'known' results

It is important that users be able to build confidence in the correctness of any R function by testing its results with some datasets for which the answers are 'known'. In the case of pedigree data, this is by no means easy achieve. The options are

- Use very simple small datasets
- Use simulated data, sampled from some population with known parameters
- Use real data and compare results with other programs

All three approaches have issues. Tiny datasets are artificial. Simulated data is a sample and will never give the exact same result as the parameters of the population from which it was drawn. Real data is messy and other programs (as well as one's own) may be in error. We use all three approaches here.

5.1 A balanced dataset: comparison with anova method

The dataset *dt8bal.df* is a simple balanced design with 8 observations of individuals from 4 sire families, with 2 individuals per family. We wish to use it to estimate additive genetic variance and to show that the results are the same as those obtained by estimating the sire and within sire variance components using the analysis of variance method.

First the *dmm()* analysis. We will just analyse the trait "CWW", so we can work straight from the dataframe, without preprocessing with function *mdf()*.

```
>data(dt8bal.df)
>dt8bal.fit <- dmm(dt8bal.df,CWW ~ 1, gls=T)
 ...
>summary(dt8bal.fit,gls=T)
Call:
summary.dmm(dmmobj = dt8bal.fit, gls = T)

Coefficients fitted by OLS for fixed effects:

  Trait Estimate StdErr CI95lo CI95hi
1   CWW     4.97  0.128   4.72   5.23


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

          Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)     CWW:CWW   0.0713  0.163 -0.2490  0.392
VarG(Ia)    CWW:CWW   0.0617  0.164 -0.2601  0.383
VarP(I)     CWW:CWW   0.1329  0.041  0.0525  0.213
```

```
Coefficients fitted by GLS for fixed effects:

  Trait Estimate StdErr CI95lo CI95hi
1  CWW     4.97  0.136   4.71   5.24



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

          Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)     CWW:CWW   0.0713  0.163 -0.2490  0.392
VarG(Ia)    CWW:CWW   0.0617  0.164 -0.2601  0.383
VarP(I)     CWW:CWW   0.1329  0.041  0.0525  0.213


>
```

So the OLS-b and GLS-b estimates of variance components are identical. That is as expected for balanced data.

Now we wish to do an analysis of variance between and within sire families, for the same data.

```
>dt8.fit <- aov(CWW ~ 1 + SId, dt8bal.df)
> summary(dt8.fit)
            Df Sum Sq Mean Sq F value Pr(>F)
SId          3  0.445  0.1483   1.262    0.4
Residuals    4  0.470  0.1175
12 observations deleted due to missingness
```

We now have the between and within sire mean squares, and we need to equate these to their expectations in terms of variance components as follows:

$$
\begin{aligned}
MS(B) &= \sigma_W^2 + 2.0 * \sigma_B^2 \\
MS(W) &= \sigma_W^2
\end{aligned}
$$

and solve these two equations to obtain $\sigma_B^2 = 0.0154$ and $\sigma_W^2 = 0.1175$. We then need to equate these two components to their expectaions in terms of causal components as follows:

$$
\begin{aligned}
\sigma_B^2 &= 0.25 * \sigma_{G(Ia)}^2 \\
\sigma_W^2 &= 0.75 * \sigma_{G(Ia)}^2 + \sigma_{E(I)}^2
\end{aligned}
$$

and solve these two equations to obtain $\sigma_{G(Ia)}^2 = 0.0616$ and $\sigma_{E(I)}^2 = 0.0713$.

These agree exactly with the analysis using function *dmm()* above. Most methods of variance component estimation agree with the analysis of variance

method for balanced designs, and this agreement extends to pedigree based causal components. This shows that *dmm* belongs to this family, and that its arithmetic is substantially correct.

5.2 The quercus program's demo2 dataset

The dataset *quercus.df* is a demonstrtation dataset from the program package QUERCUS, developed by Ruth G. Shaw and Frank H. Shaw. It is known to QUERCUS as the 'demo2' dataset. The QUERCUS program package is available from http://www.cbs.umn.edu/academics/departments/eeb/quercus.

We are going to use the program 'nf3.p' from within the QUERCUS package to compute both ML and REML estimates of components "VarE(I)", "VarG(Ia)", and "VarG(Id)", for the dataset 'demo2' which has a 2 generation pedigree with full-sib and half-sib families. The program 'nf3.p' requires a Pascal compiler - here we used the GNU gpc compiler. To run 'nf3.p' the data must be in a file called *sibships* as follows

```
    2 2 0 0 0
    0
    1    181    201     0.0214     0.4917
    2    181    201     0.8036     2.4103
    3    181    201    -0.1645    -1.1545
    4    181    202    -1.2467    -3.2364
    5    181    202    -1.1167    -2.4520
    ...
    256    0    0       -99        -99
    257    0    0       -99        -99
    258    0    0       -99        -99
    259    0    0       -99        -99
    260    0    0       -99        -99
0
```

and output appears in a file called 'Analysis' as follows

```
  This is an REML analysis.
LogLike       -272.6264
LogLike       -238.4432
LogLike       -238.4432
    At iteration    3

  *** The unconstrained analysis converged with the following results ***
  The log likelihood is -238.4432
  The mean of each trait is
        0.081236
       -0.085470
  The effect of the fixed factors is
  (in the order given, levels within factors)

  The estimates of the components are:
   Additive
```

```
        0.196601      0.134016
                      0.813206

   Environmental

        0.034755      0.134609
                      0.974764

   Dominance

        0.938617     -0.113085
                      0.046997


   Large-sample var-cov matrix of the estimates
        0.122560      0.021440      0.004091      0.093991      0.010331      0.00
1413     -0.207028     -0.028068     -0.004611
                      0.105442      0.037818      0.010331      0.065235      0.01
5525     -0.028068     -0.157275     -0.045913
                                    0.356861      0.001413      0.015525      0.18
3679     -0.004611     -0.045913     -0.482813
                                                  0.228582      0.021530      0.00
2238     -0.336299     -0.031249     -0.003320
                                                                0.153474      0.03
0090     -0.031249     -0.221470     -0.043116
                                                                              0.41
5866     -0.003320     -0.043116     -0.587751

        0.565659      0.057480      0.007096

                      0.382374      0.083193

                                    1.046082
```

The test statistic comparing two likelihoods is given by twice
their difference and is compared to Chi-square with df given by the
number of parameters specified by the hypothesis.

The above is the REML analysis. The same analysis using ML is a s follows

```
  This is an ML analysis.
LogLike       -267.4334
LogLike       -234.2705
LogLike       -234.2705
    At iteration   3
```

*** The unconstrained analysis converged with the following results ***

```
    The log likelihood is -234.2705
    The mean of each trait is
          0.081236
         -0.085470
    The effect of the fixed factors is
    (in the order given, levels within factors)

    The estimates of the components are:
     Additive

        0.149251      0.124371
                      0.726729


     Environmental

        0.022917      0.132197
                      0.953145


     Dominance

        0.985967     -0.103439
                      0.133474


      Large-sample var-cov matrix of the estimates
          0.109096      0.018697      0.003532      0.090625      0.009646      0.00
1273     -0.193564     -0.025326     -0.004052
                        0.092867      0.032809      0.009646      0.062091      0.01
4273     -0.025326     -0.144700     -0.040904
                                      0.311952      0.001273      0.014273      0.17
2452     -0.004052     -0.040904     -0.437904
                                                    0.227741      0.021358      0.00
2203     -0.332933     -0.030563     -0.003180
                                                                  0.152688      0.02
9776     -0.030563     -0.218326     -0.041863
                                                                                0.41
3060     -0.003180     -0.041863     -0.576523

           0.552195      0.054738      0.006537

                         0.369800      0.078184

                                       1.001173
```

The test statistic comparing two likelihoods is given by twice
their difference and is compared to Chi-square with df given by the
number of parameters specified by the hypothesis.

To analyse these data with *dmm()* we need to load the data, and prepare it with *mdf()*. In particular, we need to add base animals to the pedigree and compute both the additive and dominance relationship matrices. We do this as follows

```
> data(quercus.df)
> quercus.mdf <- mdf(quercus.df,pedcols=c(1:3), factorcols=4, ycols=c(5:6),
                     sexcode=c(1,2), relmat=c("E","A","D"))
Pedigree Id check:
No of rows with Id in original dataframe =  180
No of sex codes not in sexcode[] so changed to NA =  0
No of rows with Sex == NA removed from dataframe =  0
No of rows with Id == NA removed from dataframe =  0
No of rows with duplicated Id removed from dataframe =  0
No of rows remaining after duplicates and NA's removed =  180
No of SId's with no matching Id =  20
No of DId's with no matching Id =  60
Length of dataframe with base Id's added =  260
Renumber pedigree Id's:
Add matrix of multivariate traits:
Setup pedigree for nadiv():
Make relationship matrices:
starting to make D....done
Return mdf as an object of class mdf containing the dataframe as mdf$df and
  the relationship matrices as mdf$rel:
>
```

Note that *mdf()* has added 20 base sires and 60 base dams. We now do the *dmm()* analysis, with only the mean as a fixed effect

```
> quercus.fit <- dmm(quercus.mdf, Ymat ~ 1,
                     components=c("VarE(I)","VarG(Ia)","VarG(Id)"),
                     relmat="withdf", gls=T)
Dyadic mixed model fit for datafile: quercus.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 3
OLS-b step:
no of fixed effect df (k) =  1
no of traits (l) =  2
Setup antemodel matrices:
no of individuals in pedigree (m) =  260
no of individuals with data and X codes (n) =  180
Rank of X: 1   No of Fixed Effects: 1
DME substep:
QR option on dyadic model equations:
DME substep completed:
```

```
OLS-b step completed:

GLS-b step:
Warning: Multivariate GLS is not same as multiple univariate GLS's
Round =  1  Stopcrit =  2.914335e-16
Iteration completed - count =  1
Convergence achieved
GLS-b step completed successfully:
>
> summary(quercus.fit,gls=T)
Call:
summary.dmm(dmmobj = quercus.fit, gls = T)

Coefficients fitted by OLS for fixed effects:

   Trait Estimate StdErr  CI95lo CI95hi
1 Trait1   0.0812 0.0804 -0.0764  0.239

   Trait Estimate StdErr CI95lo CI95hi
1 Trait2  -0.0855  0.101 -0.283  0.112


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

           Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait1:Trait1   0.0348 0.2940 -0.5414  0.611
VarG(Ia) Trait1:Trait1   0.1966 0.1430 -0.0836  0.477
VarG(Id) Trait1:Trait1   0.9386 0.3727  0.2082  1.669
VarP(I)  Trait1:Trait1   1.1700 0.0863  1.0008  1.339

           Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait1:Trait2    0.135  0.369 -0.5887  0.858
VarG(Ia) Trait1:Trait2    0.134  0.179 -0.2177  0.486
VarG(Id) Trait1:Trait2   -0.113  0.468 -1.0301  0.804
VarP(I)  Trait1:Trait2    0.156  0.108 -0.0568  0.368

           Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait2:Trait1    0.135  0.369 -0.5887  0.858
VarG(Ia) Trait2:Trait1    0.134  0.179 -0.2177  0.486
VarG(Id) Trait2:Trait1   -0.113  0.468 -1.0301  0.804
VarP(I)  Trait2:Trait1    0.156  0.108 -0.0568  0.368

           Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait2:Trait2    0.975  0.460  0.0726   1.88
VarG(Ia) Trait2:Trait2    0.813  0.224  0.3745   1.25
VarG(Id) Trait2:Trait2    0.047  0.584 -1.0967   1.19
```

```
VarP(I)  Trait2:Trait2    1.835  0.135  1.5701   2.10

Coefficients fitted by GLS for fixed effects:

   Trait Estimate StdErr CI95lo CI95hi
1 Trait1   0.0812  0.109 -0.132  0.294

   Trait Estimate StdErr CI95lo CI95hi
1 Trait2  -0.0855  0.147 -0.374  0.203


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

            Traitpair Estimate StdErr   CI95lo CI95hi
VarE(I)  Trait1:Trait1   0.0348 0.2082 -0.37332  0.443
VarG(Ia) Trait1:Trait1   0.1966 0.1012 -0.00184  0.395
VarG(Id) Trait1:Trait1   0.9386 0.2639  0.42128  1.456
VarP(I)  Trait1:Trait1   1.1700 0.0611  1.05018  1.290

            Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait1:Trait2    0.135 0.2609 -0.3768  0.646
VarG(Ia) Trait1:Trait2    0.134 0.1269 -0.1147  0.383
VarG(Id) Trait1:Trait2   -0.113 0.3308 -0.7615  0.535
VarP(I)  Trait1:Trait2    0.156 0.0766  0.0054  0.306

            Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait2:Trait1    0.135 0.2609 -0.3768  0.646
VarG(Ia) Trait2:Trait1    0.134 0.1269 -0.1147  0.383
VarG(Id) Trait2:Trait1   -0.113 0.3308 -0.7615  0.535
VarP(I)  Trait2:Trait1    0.156 0.0766  0.0054  0.306

            Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)  Trait2:Trait2    0.975 0.3260  0.336  1.614
VarG(Ia) Trait2:Trait2    0.813 0.1585  0.502  1.124
VarG(Id) Trait2:Trait2    0.047 0.4133 -0.763  0.857
VarP(I)  Trait2:Trait2    1.835 0.0957  1.647  2.023

>
```

So the *dmm OLS-b* and *dmm GLS-b* estimates of variance components are exactly the same, and are also exactly the same as the QUERCUS REML estimates. The QUERCUS ML estimates are different.

The var-cov matrix of the estimates given by QUERCUS leads to larger standard errors than those given by *dmm*. This is not surprising, QUERCUS uses large-sample covariances from the matrix of second derivatives, while *dmm()* uses standard errors from the regression technique used to fit the dyadic model.

The QUERCUS 'demo2' dataset is essentially a balanced design, even with a dominance variance fitted, because the pedigree has equal sized families and there are no fixed effects except mean. So this reinforces the concept of a family of methods that agree with 'anova' and with each other for balanced designs. REML and *dmm OLS-b* (ie MINQUE) and *dmm GLS-b* (ie bias-corrected ML) belong to that family, but ML and the classical method of moments do not.

We have also learnt that we seem to be doing the dominance relationship matrix and "VarG(Id)" correctly, and that the process of appending relationship matrices to the dataframe seems to be sound.

5.3 The warcolak dataset

In the R package *nadiv* there is a synthetic dataset with a 3 generation pedigree designed for testing estimation of nonadditive and sex-linked genetic variances. The package author (Matthew Wolak) gives the following specification on the *warcolak* help page.

> The dataset was simulated to have two un-correlated traits with different genetic architectures (see examples below). The trait means are both equal to 1 for males and 2 for females. The additive genetic, dominance genetic, and environmental (or residual) variances for both trait1 and trait2 are 0.4, 0.3, & 0.3, respectively. However, the additive genetic variance for trait2 can be further decomposed to autosomal additive genetic variance (0.3) and X-linked additive genetic variance (0.1; assuming the no global dosage compensation mechanism).

> Females and males have equal variances (except for sex-chromosomal additive genetic variance, where by definition, males have half of the additive genetic variance as females; Wolak 2013) and a between-sex correlation of one for all genetic and residual effects (except the cross-sex residual covariance=0). All random effects were drawn from multivariate random normal distributions [e.g., autosomal additive genetic effects: N (0, kronecker(A, G))] with means of zero and (co)variances equal to the product of the expected sex-specific (co)variances (e.g., G) and the relatedness (or incidence) matrix (e.g., A).

> The actual variance in random effects will vary slightly from the amount specified in the simulation, because of Monte Carlo error. Thus, the random effects have been included as separate columns in the dataset.

We are going to use these data to check *dmm()*, but first we have to slightly alter the column names of the *warcolak* dataframe to suit *dmm()* conventions. To this end the dmm package provides a function *warcolak.convert()* just for this task.

```
> library(dmm)
> data(warcolak)
> warcolak.df <- warcolak.convert(warcolak)
> str(warcolak.df)
'data.frame': 5400 obs. of  13 variables:
 $ Id    : Factor w/ 5400 levels "u10_d1c","u10_d1d",..: 737 738 739 740 733 734 735 736 789
 $ SId   : Factor w/ 600 levels "u10_gs1","u10_gs2",..: NA NA NA NA NA NA NA NA 81 81 ...
 $ DId   : Factor w/ 1200 levels "u10_d1c","u10_d1d",..: NA NA NA NA NA NA NA NA 173 173 ...
 $ Sex   : Factor w/ 2 levels "F","M": 2 2 2 2 1 1 1 1 2 2 ...
 $ Trait1: num  1.975 1.046 1.758 0.369 1.679 ...
 $ Trait2: num  1.132 0.767 2.325 1.317 0.784 ...
```

```
 $ t1_a  : num  -0.205 -0.427 0.198 0.391 0.58 ...
 $ t2_a  : num  -0.0489 0.7127 0.9415 -0.6312 -0.8016 ...
 $ t2_s  : num  -0.0982 -0.3253 0.2469 0.2616 -0.0257 ...
 $ t1_d  : num  1.109 -0.141 0.993 -0.774 -0.521 ...
 $ t2_d  : num  0.241 -0.506 0.011 0.376 -0.281 ...
 $ t1_r  : num  0.0707 0.6133 -0.4331 -0.2486 -0.3797 ...
 $ t2_r  : num  0.0383 -0.1144 0.1258 0.3105 -0.1069 ...
> rm(warcolak)
>
```

The columns labelled *t1_a* to *t2_r* are the random effects for the additive,
sexlinked, dominance, and residual components.

We now need to preprocess *warcolak.df* with *mdf()* to renumber the Id's
and to append the additive, dominance, and sex-linked relationship matrices.
Matthew Wolak (author of the *nadiv* package) has kindly assisted me with the
information that the sexlinked breeding vales in the *warcolak* dataset were gen-
erated with the assumption of no global dosage compensation. This means that
the appropriate model for dosage compensation should be "ngdc". Unfortu-
nately an earlier release of *dmm* (dmm_1.5-1) assumed "hopi" and the analysis
reported in this document for release (dmm_1.5-1) is wrong. It was corrected in
release (dmm_1.6-1). Also the warcolak dataset was resimulated in *nadiv* release
(nadiv_2.14.0), requiring a recompute of the *dmm* analyses of the warcolak data.
This recompute is included in release (dmm_1.6-3) and later, and is reported
below.

```
> warcolak.mdf <- 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)
Pedigree Id check:
No of rows with Id in original dataframe =  5400
No of sex codes not in sexcode[] so changed to NA =  0
No of rows with Sex == NA removed from dataframe =  0
No of rows with Id == NA removed from dataframe =  0
No of rows with duplicated Id removed from dataframe =  0
No of rows remaining after duplicates and NA's removed =  5400
Length of dataframe with base Id's added =  5400
Renumber pedigree Id's:
Add matrix of multivariate traits:
Setup pedigree for nadiv():
Make relationship matrices:
starting to make D....done
Assuming male heterogametic (e.g., XX/XY) sex chromosome system
S-inverse made: Starting to make S....done
Return mdf as an object of class mdf:
 containing the dataframe as mdf$df:
 and the relationship matrices as mdf$rel:
>
```

We can now run *dmm()* and we shall fit a mean and a Sex difference as the
only fixed effects. We shall do separate runs for each trait, as Trait1 should not
have a sexlinked additive variance component fitted. Firstly Trait1

```
> warcolak.fit1 <- dmm(warcolak.mdf, Trait1 ~ 1 + Sex,
    components = c("VarE(I)","VarG(Ia)", "VarG(Id)"), relmat = "withdf")
Dyadic mixed model fit for datafile: warcolak.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 3
]OLS-b step:
no of fixed effect df (k) =  2
no of traits (l) =  1
Setup antemodel matrices:
no of individuals in pedigree (m) =  5400
no of individuals with data and X codes (n) =  5400
Rank of X: 2   No of Fixed Effects: 2
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
>
> warcolak.fit1$dme.corre
          VarE(I)  VarG(Ia)  VarG(Id)
VarE(I)   1.0000000 0.4856324 0.9190639
VarG(Ia)  0.4856324 1.0000000 0.6255619
VarG(Id)  0.9190639 0.6255619 1.0000000
>
```

There seem to be some serious collinearities in these data, particularly be-
tween "VarE(I)" and "VarG(Id)". We shall proceed to view the results.

```
> summary(warcolak.fit1)
Call:
summary.dmm(object = warcolak.fit1)


Coefficients fitted by OLS for fixed effects:

            Trait Estimate StdErr CI95lo CI95hi
(Intercept) Trait1     2.06 0.0180   2.03  2.100
SexM        Trait1    -1.02 0.0269  -1.07 -0.968



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

            Traitpair Estimate  StdErr CI95lo CI95hi
VarE(I)   Trait1:Trait1    0.282 0.03487  0.213  0.350
```

```
VarG(Ia) Trait1:Trait1    0.391 0.00857  0.374  0.408
VarG(Id) Trait1:Trait1    0.295 0.03592  0.225  0.365
VarP(I)  Trait1:Trait1    0.968 0.01315  0.942  0.993
```

The estimated components VarE(I), VarG(Ia), VarG(Id) agree reasonably
well with the stated population values (0.3,0.4,0.3 respectively).The confidence
limits include the stated population value. The collinearities noted above do
not seem to have grossly affected the results.

Because the warcolak dataset now includes the actual simulated effects as
well as their sum which is the phenotypic value of the trait, we can compute the
sample values for VarE(I), VarG(Ia), and VarG(Id) by computing the variances
of t1_r, t1_a, and t1_d . These are shown below.

```
> var(warcolak$t1_r[warcolak$sex == "M"])
[1] 0.3016221
> var(warcolak$t1_r[warcolak$sex == "F"])
[1] 0.3028589
> var(warcolak$t1_a[warcolak$sex == "M"])
[1] 0.3634443
> var(warcolak$t1_a[warcolak$sex == "F"])
[1] 0.3711862
> var(warcolak$t1_d[warcolak$sex == "M"])
[1] 0.2962183
> var(warcolak$t1_d[warcolak$sex == "F"])
[1] 0.2873658
```

We had to compute separately for each sex, because there is a Sex difference.
The sample values are close to the stated population values, as expected for a
sample of size 5400. The confidence limits of the *dmm* estimates include the
sample values, except for the male additive genetic variance.

The above *dmm* analysis is OLS-b, that is fixed effects fitted by ordinary
least squares. There is no reason to do a GLS-b run in this case as the data
are balanced (there is only a mean and a balanced Sex effect to fit), and the
estimates of variance components after GLS-b would be identical to those after
OLS-b.

We now look at the analysis of Trait2.

```
> warcolak.fit2 <- dmm(warcolak.mdf,Trait2 ~ 1 + Sex,
    components=c("VarE(I)","VarG(Ia)","VarG(Id)","VarGs(Ia)"),relmat="withdf")
Dyadic mixed model fit for datafile: warcolak.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 4
OLS-b step:
no of fixed effect df (k) =  2
no of traits (l) =  1
Setup antemodel matrices:
```

```
no of individuals in pedigree (m) =  5400
no of individuals with data and X codes (n) =  5400
Rank of X: 2   No of Fixed Effects: 2
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
>
>warcolak.fit2$dme.corre
           VarE(I)  VarG(Ia)  VarG(Id) VarGs(Ia)
VarE(I)   1.0000000 0.4856324 0.9190639 0.4080959
VarG(Ia)  0.4856324 1.0000000 0.6255619 0.8268400
VarG(Id)  0.9190639 0.6255619 1.0000000 0.5254999
VarGs(Ia) 0.4080959 0.8268400 0.5254999 1.0000000
>
```

There is an additiona serious collinearitty between "VarG(Ia)" and "VarGs(Ia)".
We shall keep this in mind as we view the results

```
> summary(warcolak.fit2)
Call:
summary.dmm(object = warcolak.fit2)


Coefficients fitted by OLS for fixed effects:


            Trait Estimate StdErr CI95lo CI95hi
(Intercept) Trait2     1.99 0.0179   1.95  2.023
SexM        Trait2    -1.00 0.0269  -1.06 -0.951




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


            Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Trait2:Trait2   0.2954 0.0349 0.2271  0.364
VarG(Ia)   Trait2:Trait2   0.2750 0.0126 0.2502  0.300
VarG(Id)   Trait2:Trait2   0.3366 0.0359 0.2662  0.407
VarGs(Ia)  Trait2:Trait2   0.0776 0.0123 0.0535  0.102
VarP(I)    Trait2:Trait2   0.9845 0.0134 0.9582  1.011


>
```

The estimated components VarE(I), VarG(Ia), VarG(Id), and VarGs(Ia), agree
well with the stated population values (0.3,0.3,0.3,0.1 respectively). The confi-
dence limits include the stated population value. The collinearities noted above
do not seem to have affected the results.

We compute the sample values for VarE(I), VarG(Ia), VarG(Id), and VarGs(Ia)
by computing the variances of t2_r, t2_a, t2_d, and t2_s. These are shown below

```
> var(warcolak$t2_r[warcolak$sex == "M"])
[1] 0.29752
> var(warcolak$t2_r[warcolak$sex == "F"])
[1] 0.2988329
> var(warcolak$t2_a[warcolak$sex == "M"])
[1] 0.3066459
> var(warcolak$t2_a[warcolak$sex == "F"])
[1] 0.300511
> var(warcolak$t2_d[warcolak$sex == "M"])
[1] 0.3113859
> var(warcolak$t2_d[warcolak$sex == "F"])
[1] 0.2891137
> var(warcolak$t2_s[warcolak$sex == "M"])
[1] 0.04912207
> var(warcolak$t2_s[warcolak$sex == "F"])
[1] 0.09356907
>
```

The confidence limits of the *dmm* estimated components include the sample
values. In the case of the sexlinked additive component the computed sample
variance of effect t2_s for males is half of that for females. That is correct. The
estimated variance component VarGs(Ia) is adjusted to an all-female basis, so
its confidence limits should include the above sample variance for females, and
that is what we see.

We conclude that comparison of *dmm()* estimates of variance components
from the simulated *warcolak* data with the data's population values, and with
the data's sample values computed from effect values included in the *warcolak*
dataset, suggests that *dmm()* estimates are correct, within the limits expected
from sampling variation.

There is one other thing that can be done as a check, and that is to analyse
each of the effects instead of the phenotypic values. We will do this just for
effect t2_s

```
> warcolak.fit2s <- dmm(warcolak.mdf,t2_s ~ 1 + Sex,
    components=c("VarE(I)","VarG(Ia)","VarG(Id)","VarGs(Ia)"),relmat="withdf")
Dyadic mixed model fit for datafile: warcolak.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 4
OLS-b step:
no of fixed effect df (k) =  2
no of traits (l) =  1
Setup antemodel matrices:
no of individuals in pedigree (m) =  5400
no of individuals with data and X codes (n) =  5400
Rank of X: 2   No of Fixed Effects: 2
```

```
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
>
> summary(warcolak.fit2s)
Call:
summary.dmm(object = warcolak.fit2s)

Coefficients fitted by OLS for fixed effects:

            Trait Estimate  StdErr   CI95lo CI95hi
(Intercept)  t2_s  0.00109 0.00496 -0.00863 0.0108
SexM         t2_s -0.00324 0.00744 -0.01783 0.0113


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

          Traitpair Estimate   StdErr   CI95lo  CI95hi
VarE(I)    t2_s:t2_s 1.00e-09 0.002662 -0.00522 0.00522
VarG(Ia)   t2_s:t2_s 1.00e-09 0.000965 -0.00189 0.00189
VarG(Id)   t2_s:t2_s 2.97e-03 0.002742 -0.00241 0.00834
VarGs(Ia)  t2_s:t2_s 9.32e-02 0.000939  0.09138 0.09506
VarP(I)    t2_s:t2_s 9.62e-02 0.001025  0.09418 0.09820


>
```

What we expect to get here is near-zero for all of the components except
VarGs(Ia), which should be close to its sample value. We find that the estimate
of VarGs(Ia) is indeed very close to its sample value for females of 0.0935. Much
closer than the estimate of VarGs(Ia) of 0.0776 obtained above from analysis
of the phenotypic values for Trait 2. We conclude that there is some evidence
that the collinearities among columns of the W matrix for components has af-
fected estimation of components. Some consideration could be given to using
principal component regression in this case. We will not do that here, there is
a section 7.4 demonstrating principal component regression.

   As far as *dmm()* is concerned all of the analyses of the *warcolak* dataset
indicate that it is performing correctly. We conclude with a word of caution. The
above analyses ignore the possibility that genetic parameters are sex-specific.
While this is OK for *warcolak* because we know from the simulation that genetic
effects do not differ between the sexes, it should not be assumed in general. It is
possible to use *dmm* to perform sex-specific analyses. See the separate document
dmmClassSpecific.pdf.

5.4 The Harvey dataset

The *harv101.df* dataset comes originally from Harvey(1960) [11]. It is real data, average daily gain on 65 Hereford steers. It has been extensively used as a test example for many years. It is unbalanced in both the fixed effects and the pedigree families.

We are going to compare estimates of additive genetic and environmental variance components obtained from these data with two programs, *dmm()* and DFREML. The program DFREML was written by Karin Meyer, and does REML estimates only. We shall also compare with estimates from Harvey's original analysis by the fitting constants method, which is an analysis of variance method, commonly known as Henderson's Method-3.

The program DFREML (Meyer(1998) [19]) is written in Fortran. It requires separate files for pedigree information and data. It also requires some interactive input which under Unix can be redirected from *standard input*. DFREML is actually a package of several programs, and we are going to use the routine DFUNI, which is for univariate analyses, along with the routine DFPREP, which recodes the pedigree and data files. We are not going to give all the details, just the output file from the final run as follows

```
                         *** DFREML 3.1.000        ***
                         Last modified : May 11, 2001



********************************************************************************

                         PROGRAM " D F U N I"

        ESTIMATE VARIANCE COMPONENTS FOR AN INDIVIDUAL ANIMAL MODEL

********************************************************************************KM**


 Today is 17/05/2010 -- Time is 15:30
 Running on host : "not determined"
 ---------------------------------
 DESCRIPTION OF DATA SET
 ---------------------------------

 harv103df.dat with dfuni - no interaction


 ANALYSIS FOR TRAIT :             1       adg

 Data file used      : "
 Pedigree file used  : "
 Data directory      : "
```

```
----------------------------------
MODEL OF ANALYSIS & DATA STRUCTURE
----------------------------------


ANALYSIS FITTING MODEL NO.                       =              1

NO. OF RECORDS                                   =             65
NO. OF ANIMALS                                   =             74

NO. OF TRAITS                                    =              1
   1    adg          MEAN =    2.41138     SDEV =   0.277809

NO. OF FIXED EFFECTS                             =              2
... WITH TOTAL NO. OF LEVELS                     =              6
   1    line         NO. OF LEVELS =     3
   2    agedam       NO. OF LEVELS =     3

NO. OF COVARIABLES                               =              2
... WITH TOTAL NO. OF REGRESSION COEFFICIENTS    =              2
   1    age          ORDER FITTED =   1   MEAN =    176.646      SDEV =    1
4.7048
   2    wt           ORDER FITTED =   1   MEAN =    416.846      SDEV =    4
1.4149

NO. OF EQUATIONS IN TOTAL                        =             83


-----------------------------
SUMMARY OF PEDIGREE STRUCTURE
-----------------------------


NO. OF "BASE" ANIMALS                      =            74
NO. OF ANIMALS WITH RECORDS                =            65
 ... WITH UNKNOWN/PRUNED SIRE              =             0
 ... WITH UNKNOWN/PRUNED DAM               =            65
NO. OF SIRES WITH PROGENY RECORDS          =             9
NO. OF DAMS WITH PROGENY RECORDS           =             0
NO. OF GRAND-SIRES W. PROGENY RECORDS      =             0
NO. OF GRAND-DAMS W. PROGENY RECORDS       =             0
-------------------------------------
OPTIONS SET IN OPTIMIZATION ROUTINE
-------------------------------------


USE QUADRATIC APPROXIMATION OF LOG L
MAXIMUM NO. OF ITERATES ALLOWED                  =            500
```

```
----------------------
CHARACTERISTICS OF RUN :
----------------------


RUN WITH OPTION                           =                0

NO. OF NON-ZERO PIVOTS ENCOUNTERED        =               81
NO. OF DEGREES OF FREEDOM 1               =               58
NO. OF DEGREES OF FREEDOM 2               =              -16

NO. OF ITERATES CARRIED OUT               =                0
NO. OF LIKELIHOODS EVALUATED              =               11

PARAMETERS : STARTING VALUES AND ESTIMATES    =
    1    HERITABILITY       0.4000000000    0.7400498769
    0    LOG L              34.62023750     34.82083038


-------------------------------------------
ESTIMATES OF VARIANCES & GENETIC PARAMETERS
-------------------------------------------


UNIVARIATE ANALYSIS FOR TRAIT NO.         =                1

TOTAL SUMS OF SQUARES (Y'Y)               =    4.939375385
SUMS OF SQUARES FOR RESIDUAL (Y'PY)       =    1.033240267
LOG DETERMINANT OF COEFFICIENT MATRIX     =    47.24592509
LOG DETERMINANT OF NRM                    =   -18.69933471
LOG LIKELIHOOD (WITH NRM)                 =    34.82083038

ADDITIV-GENETIC (DIRECT) VARIANCE   1     =  0.5071591820E-01
ERROR VARIANCE                            =  0.1781448737E-01
PHENOTYPIC VARIANCE    1                  =  0.6853040557E-01
PHENOTYPIC STANDARD DEVIATION             =         0.2618
PHENOTYPIC COEFFICIENT OF VARIATION (%)   =        10.8561

HERITABILITY    1                         =         0.7400          0


------------------------------------
APPROXIMATION OF SAMPLING VARIANCES
------------------------------------


NO. OF LIKELIHOOD VALUES AVAILABLE        =               11
QUADRATIC APPROXIMATION OF LIKELIHOOD :
NO. OF PARAMETERS OF FUNCTION             =                2
NORM OF GRADIENT VECTOR                   =  0.2075215705E-02
"RANGE" PARAMETER (IN %)                  =   10.00000000
```

```
NO. OF POINTS USED                           =               4

RANK OF APPROXIMATE INFORMATION MATRIX       =               1
LOG DETERMINANT .....................        = 0.9746069797

PARAMETER ESTIMATES WITH THEIR APPROX. S.E.
  1 HERITABILITY                   :   0.740050        0.614281

CONSTANT                                     =   34.82083038
LINEAR    COEFFCIENT             1            = 0.20752157E-02
QUADRATIC COEFFICIENT           1   1        = -1.3250627

CUBIC APPROXIMATION OF LIKELIHOOD :
NO. OF PARAMETERS OF FUNCTION                =               3
NORM OF GRADIENT VECTOR                      = 0.1835470165E-05
"RANGE" PARAMETER (IN %)                     =   10.00000000
NO. OF POINTS USED                           =               4

RANK OF APPROXIMATE INFORMATION MATRIX       =               1
LOG DETERMINANT .....................        = 0.9875165352

PARAMETER ESTIMATES WITH THEIR APPROX. S.E.
  1 HERITABILITY                   :   0.740050        0.610328

CONSTANT                                     =   34.82083038
LINEAR    COEFFCIENT             1            = 0.18354702E-05
QUADRATIC COEFFICIENT           1   1        = -1.3422796
CUBIC COEFFICIENT               1   1   1 = 0.86430366

-----------------------------------------
SOLUTION FOR FIXED EFFECTS AND COVARIABLES
-----------------------------------------

COVARIABLE NO.   1    age

EQ.NO.                      ORIG.ID.  NREC    MEAN     GLS-SOLUTION      L
SQ-SOLUTION
    2 level/ORDER      1    1                           -0.8556266281E-02 -0.816
4370130E-02

COVARIABLE NO.   2    wt

EQ.NO.                      ORIG.ID.  NREC    MEAN     GLS-SOLUTION      L
SQ-SOLUTION
    3 level/ORDER      1    1                            0.2447670253E-02 0.251
8596921E-02
```

```
FIXED EFFECT NO.   1    line

EQ.NO.                     ORIG.ID.   NREC     MEAN     GLS-SOLUTION        L
SQ-SOLUTION
    4  LEVEL           1         1     21    2.4019       0.08054226
0.05061950
    5  LEVEL           2         2     15    2.5447       0.16935178
0.14488600
    6  LEVEL           3         3     29    2.3493       0.11588633
0.08635849


FIXED EFFECT NO.   2    agedam

EQ.NO.                     ORIG.ID.   NREC     MEAN     GLS-SOLUTION        L
SQ-SOLUTION
    7  LEVEL           1         3     12    2.4575       0.00000000
0.00000000
    8  LEVEL           2         4     16    2.4537      -0.06872572
-0.03049424
    9  LEVEL           3         5     37    2.3781      -0.17235061
-0.14196724


********************************************************************************
```

The above output is for an analysis of the trait 'adg' with fixed effects
fitted for 'line' and 'agedam' and covariates 'age' and 'weight'. There is no
interaction 'line x agedam' fitted. This differs from the the analysis publised by
Harvey(1960) [11] in which the above interaction is included.

The Harvey(1960) [11] analysis results in variance components for 'sire/line'
and 'residual' as follows

$$\sigma^2_{S/L} = 0.0166$$
$$\sigma^2_W = 0.0522$$

We need to equate these two components to their expectaions in terms of
causal components as follows:

$$\sigma^2_{S/L} = 0.25 * \sigma^2_{G(Ia)}$$
$$\sigma^2_W = 0.75 * \sigma^2_{G(Ia)} + \sigma^2_{E(I)}$$

and solve these two equations to obtain

$$\sigma^2_{G(Ia)} = 0.0656$$
$$\sigma^2_{E(I)} = 0.0030.$$

44

These sum to $\sigma_P^2 = 0.0686$. So the Harvey analysis by fitting constants results in a larger additive genetic variance and a smaller environmental variance, while their sum, the phenotypic variance is almost identical.

We now wish to analyse these data with *dmm()*. We start with data preparation

```
> library(dmm)
> data(harv101.df)
> str(harv101.df)
'data.frame': 139 obs. of  9 variables:
 $ Id    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ SId   : int  NA NA NA NA NA NA NA NA NA NA ...
 $ DId   : int  NA NA NA NA NA NA NA NA NA NA ...
 $ Line  : int  NA NA NA NA NA NA NA NA NA NA ...
 $ Agedam: int  NA NA NA NA NA NA NA NA NA NA ...
 $ Age   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ Weight: num  NA NA NA NA NA NA NA NA NA NA ...
 $ Adg   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ Sex   : num  1 1 1 1 1 1 1 1 1 1 ...
>
> harv101.mdf <- mdf(harv101.df, pedcols=c(1:3), factorcols=c(4:5,9),
                    ycols=8, sexcode=c(1,2),keep=T)
Loading required package: Matrix
Loading required package: lattice
Pedigree Id check:
No of rows with Id in original dataframe =  139
No of sex codes not in sexcode[] so changed to NA =  0
No of rows with Sex == NA removed from dataframe =  0
No of rows with Id == NA removed from dataframe =  0
No of rows with duplicated Id removed from dataframe =  0
No of rows remaining after duplicates and NA's removed =  139
Length of dataframe with base Id's added =  139
Renumber pedigree Id's:
Add matrix of multivariate traits:
Return mdf as a normal dataframe:
>
> str(harv101.mdf)
'data.frame': 139 obs. of  10 variables:
 $ Id    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ SId   : int  NA NA NA NA NA NA NA NA NA NA ...
 $ DId   : int  NA NA NA NA NA NA NA NA NA NA ...
 $ Line  : Factor w/ 3 levels "1","2","3": NA NA NA NA NA NA NA NA NA NA ...
 $ Agedam: Factor w/ 3 levels "3","4","5": NA NA NA NA NA NA NA NA NA NA ...
 $ Age   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ Weight: num  NA NA NA NA NA NA NA NA NA NA ...
 $ Adg   : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
$ Sex    : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
$ Ymat   : num [1:139, 1] NA NA NA NA NA NA NA NA NA NA ...
>
```

Notice that we are using argument *keep=T* with *mdf()* to force it to keep the numeric columns 'Age' and Weight', which we wish to use as covariates. We did not bother with relationship matrices, as we only need the additive relationship matrix, and *dmm()* can calculate it "inline".

First we will fit the model without interaction used with DFREML above and we can use the default components (environmental and additive genetic)

```
> harv101.fit1 <- dmm(harv101.mdf,
                   Adg ~ 1 + Line + Agedam + Age + Weight, gls=T)
Dyadic mixed model fit for datafile: harv101.mdf
Data file is a normal dataframe:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 2
OLS-b step:
no of fixed effect df (k) =  7
no of traits (l) =  1
Setup antemodel matrices:
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
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:

GLS-b step:
Round =  1  Stopcrit =  0.02352827
Round =  2  Stopcrit =  0.004211696
Iteration completed - count =  2
Convergence achieved
GLS-b step completed successfully:
>
>harv101.fit1$dme.corre
           VarE(I)  VarG(Ia)
VarE(I)  1.0000000 0.8755222
VarG(Ia) 0.8755222 1.0000000

>
>summary(harv101.fit1,gls=T)
Call:
summary.dmm(dmmobj = harv101.fit1, gls = T)
```

```
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-b fit:

         Traitpair Estimate  StdErr  CI95lo CI95hi
VarE(I)    Adg:Adg   0.0167 0.01550 -0.0136 0.0471
VarG(Ia)   Adg:Adg   0.0520 0.01475  0.0230 0.0809
VarP(I)    Adg:Adg   0.0687 0.00758  0.0538 0.0836


Coefficients fitted by GLS for fixed effects:

            Trait Estimate   StdErr      CI95lo    CI95hi
(Intercept)  Adg   2.97709 0.499218   1.998620   3.95555
Line2        Adg   0.08906 0.135158  -0.175846   0.35397
Line3        Adg   0.03535 0.112905  -0.185944   0.25664
Agedam4      Adg  -0.06694 0.093367  -0.249939   0.11606
Agedam5      Adg  -0.17092 0.083863  -0.335292  -0.00655
Age          Adg  -0.00854 0.002918  -0.014257  -0.00282
Weight       Adg   0.00245 0.000869   0.000748   0.00415


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

         Traitpair Estimate  StdErr  CI95lo CI95hi
VarE(I)    Adg:Adg   0.0183 0.01539 -0.0119 0.0484
VarG(Ia)   Adg:Adg   0.0498 0.01446  0.0215 0.0781
VarP(I)    Adg:Adg   0.0681 0.00756  0.0533 0.0829

>
```

So both the OLS-b and GLS-b estimates of VarE(I) and VarG(Ia) are close to the DFREML estimates 0.01781 and 0.05071, but they are not exactly equal. The OLS-b and GLS-b estimates are also close but not equal, as expected for an unbalanced dataset. The fixed effects for covariates 'Age' and 'Weight' also agree with DFREML, but the constants fitted for 'Line' and 'Agedam' differ

because DFREML uses a different set of contrasts to the default used in R.

We now do another fit with the 'Line x Agedam' interaction included

```
> harv101.fit2 <- dmm(harv101.mdf,
        Adg ~ 1 + Line + Agedam + Line:Agedam + Age + Weight, gls=T)
Dyadic mixed model fit for datafile: harv101.mdf
Data file is a normal dataframe:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 2
OLS-b step:
no of fixed effect df (k) =  11
no of traits (l) =   1
Setup antemodel matrices:
no of individuals in pedigree (m) =   139
no of individuals with data and X codes (n) =   65
Rank of X: 11   No of Fixed Effects: 11
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:

GLS-b step:
Round =  1  Stopcrit =  0.07444493
Round =  2  Stopcrit =  0.01458684
Round =  3  Stopcrit =  0.00358769
Iteration completed - count =  3
Convergence achieved
GLS-b step completed successfully:
>
> summary(harv101.fit2, gls = T)

Coefficients fitted by OLS for fixed effects:
```

	Trait	Estimate	StdErr	CI95lo	CI95hi
(Intercept)	Adg	2.72290	0.503339	1.73636	3.70945
Line2	Adg	0.14911	0.205286	-0.25325	0.55147
Line3	Adg	0.11449	0.186059	-0.25018	0.47917
Agedam4	Adg	-0.04942	0.189274	-0.42040	0.32156
Agedam5	Adg	-0.03449	0.166360	-0.36056	0.29157
Age	Adg	-0.00709	0.002986	-0.01295	-0.00124
Weight	Adg	0.00224	0.000903	0.00047	0.00401
Line2:Agedam4	Adg	-0.09554	0.270166	-0.62507	0.43399
Line3:Agedam4	Adg	0.11616	0.239031	-0.35234	0.58466
Line2:Agedam5	Adg	-0.02430	0.240672	-0.49602	0.44741
Line3:Agedam5	Adg	-0.20179	0.206156	-0.60586	0.20227

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

        Traitpair Estimate StdErr   CI95lo CI95hi
VarE(I)   Adg:Adg  0.00293 0.0147  -0.0259 0.0317
VarG(Ia)  Adg:Adg  0.06472 0.0139   0.0374 0.0920
VarP(I)   Adg:Adg  0.06764 0.0071   0.0537 0.0816


Coefficients fitted by GLS for fixed effects:

              Trait Estimate   StdErr    CI95lo    CI95hi
(Intercept)     Adg  2.87126 0.489738  1.911370   3.83114
Line2           Adg  0.21954 0.222333 -0.216228   0.65532
Line3           Adg  0.24867 0.199647 -0.142637   0.63998
Agedam4         Adg  0.01700 0.175097 -0.326191   0.36019
Agedam5         Adg  0.04031 0.157460 -0.268317   0.34893
Age             Adg -0.00779 0.002976 -0.013621  -0.00195
Weight          Adg  0.00204 0.000857  0.000363   0.00372
Line2:Agedam4   Adg -0.18609 0.248449 -0.673049   0.30087
Line3:Agedam4   Adg -0.05271 0.222334 -0.488489   0.38306
Line2:Agedam5   Adg -0.12788 0.223753 -0.566433   0.31068
Line3:Agedam5   Adg -0.36253 0.194124 -0.743015   0.01795


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

        Traitpair Estimate  StdErr  CI95lo CI95hi
VarE(I)   Adg:Adg  0.00191 0.01461 -0.0267 0.0305
VarG(Ia)  Adg:Adg  0.06593 0.01352  0.0394 0.0924
VarP(I)   Adg:Adg  0.06784 0.00713  0.0539 0.0818

>
```

So both the OLS-b and GLS-b estimates of VarE(I) and VarG(Ia) are now very close to those from Harvey's fitting constants analysis 0.0030 and 0.0656. The fixed effects differ from Harvey, and this is not surprising as Harvey's model includes a 'Sire' effect. It is also not surprising that the variance components are now a little different from the DFREML estimates, as the fixed effects model differs.

What we can conclude is that *dmm()* has produced reasonable estimates of variance components from an unbalanced dataset, and that they are close to REML estimates in this case. It is also clear that OLS-b and GLS-b estimates will differ for an unbalanced dataset; we do not expect MINQUE and bias-corrected-ML to be the same.

5.5 The DFREML dataset

The dataset *tstmo1.df* is an example dataset from program DFREML (Meyer(1998) [19]). Inside DFREML it is known as 'Example 1'. It is a univariate dataset with 282 individuals in a 3 generation pedigree with full-sib families.

Karin Meyer gives the following description of these data:

> .... The test data given is that used by Meyer(1989) to illustrate univariate REML estimation via a derivtive-free algorithm. They are simulated records for a trait with a phenotypic variance of 100, direct heritability of 0.40, maternal heritability of 0.20, maternal-direct covariance (divided by 100) of -0.05, and a "c-squared" effect of 0.15. Data were generated for 2 generations of animals with a heirarchical full-sib family structure, yielding a total of 282 records and 306 animals in the analysis with generations as the only fixed effect.

We are going to use the program DFUNI from the DFREML package to estimate environmental and additive genetic variance components, after fitting a fixed effect called 'Gen' which is the generation number of each individual. We will not give the runing details. The final output including estimates is as follows

```
                        *** DFREML 3.1.000        ***
                        Last modified : May 11, 2001



******************************************************************************

                        PROGRAM " D F U N I"

      ESTIMATE VARIANCE COMPONENTS FOR AN INDIVIDUAL ANIMAL MODEL

****************************************************************************KM**

 Today is 25/05/2010 -- Time is 18:59
 Running on host : "not determined"
 ---------------------------------
 DESCRIPTION OF DATA SET
 ---------------------------------

 Example 1 - tstmo1.d with dfuni


 ANALYSIS FOR TRAIT :            1      weight

 Data file used      : "
```

```
Pedigree file used  : "
Data directory      : "


------------------------------------
MODEL OF ANALYSIS & DATA STRUCTURE
------------------------------------


ANALYSIS FITTING MODEL NO.                     =            1

NO. OF RECORDS                                 =          282
NO. OF ANIMALS                                 =          306

NO. OF TRAITS                                  =            1
   1    weight          MEAN =    229.787      SDEV =    13.4939

NO. OF FIXED EFFECTS                           =            1
... WITH TOTAL NO. OF LEVELS                   =            2
   1    generation     NO. OF LEVELS =     2

NO. OF COVARIABLES                             =            0

NO. OF EQUATIONS IN TOTAL                      =          309


------------------------------
SUMMARY OF PEDIGREE STRUCTURE
------------------------------


NO. OF "BASE" ANIMALS                  =          24
NO. OF ANIMALS WITH RECORDS            =         282
 ... WITH UNKNOWN/PRUNED SIRE          =           0
 ... WITH UNKNOWN/PRUNED DAM           =           0
NO. OF SIRES WITH PROGENY RECORDS      =          12
NO. OF DAMS WITH PROGENY RECORDS       =          36
NO. OF GRAND-SIRES W. PROGENY RECORDS  =          10
NO. OF GRAND-DAMS W. PROGENY RECORDS   =          18
------------------------------------
OPTIONS SET IN OPTIMIZATION ROUTINE
------------------------------------


USE QUADRATIC APPROXIMATION OF LOG L
MAXIMUM NO. OF ITERATES ALLOWED                =          500


----------------------
CHARACTERISTICS OF RUN :
----------------------
```

```
RUN WITH OPTION                             =            0

NO. OF NON-ZERO PIVOTS ENCOUNTERED          =          308
NO. OF DEGREES OF FREEDOM 1                 =          280
NO. OF DEGREES OF FREEDOM 2                 =          -26

NO. OF ITERATES CARRIED OUT                 =            0
NO. OF LIKELIHOODS EVALUATED                =            8

PARAMETERS : STARTING VALUES AND ESTIMATES  =
   1   HERITABILITY      0.3000000000        0.4633476480
   0   LOG L            -760.7423178        -759.5034413


------------------------------------------
ESTIMATES OF VARIANCES & GENETIC PARAMETERS
------------------------------------------


UNIVARIATE ANALYSIS FOR TRAIT NO.           =            1

TOTAL SUMS OF SQUARES (Y'Y)                 =   51165.23404
SUMS OF SQUARES FOR RESIDUAL (Y'PY)         =   14262.75854
LOG DETERMINANT OF COEFFICIENT MATRIX       =   378.8445786
LOG DETERMINANT OF NRM                      =  -195.4675049
LOG LIKELIHOOD (WITH NRM)                   =  -759.5034413

ADDITIV-GENETIC (DIRECT) VARIANCE   1       =   43.98042526
ERROR VARIANCE                              =   50.93842336
PHENOTYPIC VARIANCE   1                     =   94.91884862
PHENOTYPIC STANDARD DEVIATION               =        9.7426
PHENOTYPIC COEFFICIENT OF VARIATION (%)     =        4.2398

HERITABILITY   1                            =        0.4633          0


----------------------------------
APPROXIMATION OF SAMPLING VARIANCES
----------------------------------


NO. OF LIKELIHOOD VALUES AVAILABLE          =            8
QUADRATIC APPROXIMATION OF LIKELIHOOD :
NO. OF PARAMETERS OF FUNCTION               =            2
NORM OF GRADIENT VECTOR                     = 0.1407278372E-03
"RANGE" PARAMETER (IN %)                    =   10.00000000
NO. OF POINTS USED                          =            3

RANK OF APPROXIMATE INFORMATION MATRIX      =            1
LOG DETERMINANT .......................     =   4.421234753
```

```
PARAMETER ESTIMATES WITH THEIR APPROX. S.E.
   1  HERITABILITY                      :   0.463348        0.109633

CONSTANT                                       =  -759.5034413
LINEAR    COEFFCIENT            1               =  0.14072784E-03
QUADRATIC COEFFICIENT           1    1          =  -41.599476

CUBIC APPROXIMATION OF LIKELIHOOD :
NO. OF PARAMETERS OF FUNCTION                  =             3
NORM OF GRADIENT VECTOR                        =  0.7416302902E-02
"RANGE" PARAMETER (IN %)                       =   30.00000000
NO. OF POINTS USED                             =             5

RANK OF APPROXIMATE INFORMATION MATRIX         =             1
LOG DETERMINANT ......................         =   4.429672096

PARAMETER ESTIMATES WITH THEIR APPROX. S.E.
   1  HERITABILITY                      :   0.463348        0.109171

CONSTANT                                       =  -759.5034413
LINEAR    COEFFCIENT            1               =  -0.74163029E-02
QUADRATIC COEFFICIENT           1    1          =  -41.951950
CUBIC COEFFICIENT               1    1    1  =   16.196033

------------------------------------------
SOLUTION FOR FIXED EFFECTS AND COVARIABLES
------------------------------------------


FIXED EFFECT NO.   1    generation

EQ.NO.                   ORIG.ID.   NREC    MEAN     GLS-SOLUTION       L
SQ-SOLUTION
   2  LEVEL         1         1    138  220.4638      -9.46612491
-9.32345865
   3  LEVEL         2         2    144  238.7222       6.90686757
8.93499546


********************************************************************************
```

So the phenotypic variance of 94.9 is not exactly the population value of 100 and the additive genetic variance of 43.9 is not exactly the population value of 40. This is a small sample for simulated data and the deviations are to be expected.

We now want to conduct the same analysis with *dmm()*. First prepare the data. This time we will do the relationship matrices during preparation and

append them to the dataframe.

```
> library(dmm)
> data(tstmo1.df)
> str(tstmo1.df)
'data.frame': 282 obs. of  6 variables:
 $ Id    : int  10025 10026 10027 10028 10029 10030 10031 10032 10033 10034 ...
 $ SId   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ DId   : int  2 2 2 2 2 2 2 2 3 3 ...
 $ Sex   : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 2 1 1 1 ...
 $ Gen   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Weight: num  220 212 221 207 218 201 214 229 214 198 ...
>
> tstmo1.mdf <- mdf(tstmo1.df, pedcols=c(1:3), factorcols=c(4:5), ycols=6,
                  sexcode=c(1,2), relmat=c("E","A"))
Pedigree Id check:
No of rows with Id in original dataframe =  282
No of sex codes not in sexcode[] so changed to NA =  0
No of rows with Sex == NA removed from dataframe =  0
No of rows with Id == NA removed from dataframe =  0
No of rows with duplicated Id removed from dataframe =  0
No of rows remaining after duplicates and NA's removed =  282
No of SId's with no matching Id =  6
No of DId's with no matching Id =  18
Length of dataframe with base Id's added =  306
Renumber pedigree Id's:
Add matrix of multivariate traits:
Setup pedigree for nadiv():
Make relationship matrices:
Return mdf as an object of class mdf:
 containing the dataframe as mdf$df:
 and the relationship matrices as mdf$rel:
>
>str(tstmo1.mdf)
List of 2
 $ df :'data.frame': 306 obs. of  6 variables:
  ..$ Id  : int [1:306] 1 2 3 4 5 6 7 8 9 10 ...
  ..$ SId : int [1:306] NA NA NA NA NA NA NA NA NA NA ...
  ..$ DId : int [1:306] NA NA NA NA NA NA NA NA NA NA ...
  ..$ Sex : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 2 2 2 2 ...
  ..$ Gen : Factor w/ 2 levels "1","2": NA NA NA NA NA NA NA NA NA NA ...
  ..$ Ymat: num [1:306, 1] NA NA NA NA NA NA NA NA NA NA ...
 $ rel:List of 8
 ...
>
```

Now we can do the *dmm* fit of the same simple model as used for DFREML

```
> tstmo1.fit1 <- dmm(tstmo1.mdf, Ymat ~ 1 + Gen,
                 components=c("VarE(I)","VarG(Ia)"), gls=T)
Dyadic mixed model fit for datafile: tstmo1.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 2
OLS-b step:
no of fixed effect df (k) =  2
no of traits (l) =  1
Setup antemodel matrices:
no of individuals in pedigree (m) =  306
no of individuals with data and X codes (n) =  282
Rank of X: 2   No of Fixed Effects: 2
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:

GLS-b step:
Round =  1  Stopcrit =  1.01928
Round =  2  Stopcrit =  0.2283226
Round =  3  Stopcrit =  0.04124407
Round =  4  Stopcrit =  0.007648931
Iteration completed - count =  4
Convergence achieved
GLS-b step completed successfully:
>
>tstmo1.fit1$dme.corre
           VarE(I)   VarG(Ia)
VarE(I)  1.0000000 0.4158424
VarG(Ia) 0.4158424 1.0000000
>
> summary(tstmo1.fit1,gls=T)
Call:
summary.dmm(dmmobj = tstmo1.fit1, gls = T)


Coefficients fitted by OLS for fixed effects:

            Trait Estimate StdErr CI95lo CI95hi
(Intercept) Ymat     220.5  0.846  218.8  222.1
Gen2        Ymat      18.3  1.184   15.9   20.6


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

        Traitpair Estimate StdErr CI95lo CI95hi
```

```
VarE(I)  Ymat:Ymat      32.7   6.41   20.1   45.3
VarG(Ia) Ymat:Ymat      70.0   2.82   64.5   75.6
VarP(I)  Ymat:Ymat     102.7   5.83   91.3  114.2


Coefficients fitted by GLS for fixed effects:

             Trait Estimate StdErr CI95lo CI95hi
(Intercept)  Ymat    220.3   2.21  216.0  224.6
Gen2         Ymat     15.8   1.30   13.3   18.4


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

          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)  Ymat:Ymat      33.6   6.54   20.8   46.4
VarG(Ia) Ymat:Ymat      78.4   2.86   72.8   84.0
VarP(I)  Ymat:Ymat     112.0   5.91  100.4  123.6


>
```

So the OLS-b and GLS-b estimates are not identical (as expected for an unbalanced dataset) but of more concern is that they differ substantially from the DFREML estimates. This requires some consideration. Firstly, they are not expected to agree exactly because *dmm()* estimates are not REML equivalent. Secondly, the 'Gen' fixed effect in these data is not independent of the pedigree; generation 1 are parents and generation 2 are their offspring. So there is a parent/offspring correlation between generations 1 and 2, and this means that the levels of the 'Gen' fixed effect are not independent. So the analyses violate one of the basic assumptions of the model, both for DFREML and *dmm()*. Correlated levels of an effect is as serious a problem as correlated residuals. It would be expected that DFREML and *dmm()* would be biassed in different ways in such a case. We should conclude that neither program is 'correct' in this case.

There remains the question of whether we can match the stated population parameters for this dataset, notwithstanding the problem with the 'Gen' effect. This requires another run with more variance components fitted, as follows

```
> tstmo1.fit2 <- dmm(tstmo1.mdf, Ymat ~ 1 + Gen,
      components=c("VarE(I)","VarG(Ia)","VarE(M)","VarG(Ma)","CovG(Ia,Ma)",
      "CovG(Ma,Ia)"), gls=T)
 ...
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
```

```
GLS-b step:
Fixed effects iterated -> GLS b
Partitioned variance components from DME after GLS b
Round =  1  Stopcrit =  7.486482
Round =  2  Stopcrit =  5.989145
Round =  3  Stopcrit =  6.283543
 ...
Round =  200  Stopcrit =  6.234484
Iteration completed - count =  200
Failed to converge
GLS-b step abandoned:
>
>tstmo1.fit2$dme.corre
              VarE(I)   VarG(Ia)   VarE(M)   VarG(Ma) CovG(Ia,Ma) CovG(Ma,Ia)
VarE(I)     1.0000000 0.4158424 0.3457454 0.2933538   0.2209086   0.2209086
VarG(Ia)    0.4158424 1.0000000 0.6412865 0.7312091   0.7658688   0.7658688
VarE(M)     0.3457454 0.6412865 1.0000000 0.8488271   0.6383430   0.6383430
VarG(Ma)    0.2933538 0.7312091 0.8488271 1.0000000   0.8376605   0.8376605
CovG(Ia,Ma) 0.2209086 0.7658688 0.6383430 0.8376605   1.0000000   0.7037866
CovG(Ma,Ia) 0.2209086 0.7658688 0.6383430 0.8376605   0.7037866   1.0000000
>
```

So the GLS-b iteration fails to converge, another sign of problems with the
model or dataset. The correlations of these components are rather high, but we
can still have a look at the OLS-b estimates

```
> summary(tstmo1.fit2)
Call:
summary.dmm(dmmobj = tstmo1.fit2)


Coefficients fitted by OLS for fixed effects:

            Trait Estimate StdErr CI95lo CI95hi
(Intercept) Ymat    220.5  0.846  218.8  222.1
Gen2        Ymat     18.3  1.184   15.9   20.6



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

            Traitpair  Estimate StdErr CI95lo CI95hi
VarE(I)     Ymat:Ymat  2.37e+01   6.77  10.47  37.00
VarG(Ia)    Ymat:Ymat  7.95e+01   5.36  68.94  89.96
VarE(M)     Ymat:Ymat  1.00e-09   4.58  -8.97   8.97
VarG(Ma)    Ymat:Ymat  3.41e+01   6.83  20.76  47.53
CovG(Ia,Ma) Ymat:Ymat -2.50e+01   6.23 -37.26 -12.82
CovG(Ma,Ia) Ymat:Ymat -2.50e+01   6.23 -37.26 -12.82
```

```
VarP(I)      Ymat:Ymat   8.72e+01    7.63   72.28 102.20

>
```

The population values for these data are VarE(I) = 35, VarG(Ia) = 40, VarE(M) = 15, VarG(Ma) = 20, and CovG(Ia,Ma) = -5. I am assuming that what Karin Meyer calls a 'c-squared' effect' is actually VarE(M). Only for two components do the confidence limits include the population value. It is a small sample and there are problems with the model noted above.

What we can conclude from this somewhat unsatisfactory exposition, is that *dmm()* will not always agree with REML, and that we should avoid trying to use a pathological dataset to compare results from different programs. The example is pathological because the given result for DFREML is with a model that omits important significant variance components. The 'omitted variable' effect is operating here - we do not know how the omitted maternal additive genetic variance will be redistributed among the other variances when it is omitted. A larger sample would also be desirable for simulated data.

6   Mathematical methods

6.1   Genetic models and definition of causal components

An observation or a measurement of an individual is known as the individual's phenotypic value ($\boldsymbol{P}(I)$). $\boldsymbol{P}(I)$ can be multivariate. It is usually expressed as a deviation from a population mean and may also be adjusted for fixed effects other than the mean.

6.1.1   Individual effects

There is a conceptual division of individual phenotypic value into individual genetic value ($\boldsymbol{G}(I)$) and individual environmental deviations ($\boldsymbol{E}(I)$), and these values are additive so that for any individual

$$\boldsymbol{P}(I) = \boldsymbol{G}(I) + \boldsymbol{E}(I) \tag{1}$$

although this partitioning is generally not achievable in practice.

What can be done is to estimate some specific components of $\boldsymbol{G}(I)$ and $\boldsymbol{E}(I)$ and assume that those not estimated are zero, so that these sum to $\boldsymbol{G}(I)$ and $\boldsymbol{E}(I)$ respectively, which in turn sum to $\boldsymbol{P}(I)$ as in equation 1.

Estimable components of the $\boldsymbol{G}(I)$ and $\boldsymbol{E}(I)$ values are defined in Table 1. A full explanation of gene effects on individual phenotype can be found in introductory texts, for example Falconer(1961) [7] , Kempthorne(1957) [15], or Lynch and Walsh(1998) [17].

All the components of value in Table 1 are independent and additive, so we can write, for example

$$\boldsymbol{G}(I) = \boldsymbol{G}(Ia) + \boldsymbol{G}(Id)$$

and

$$\boldsymbol{P}(I) = \boldsymbol{G}(Ia) + \boldsymbol{G}(Id) + \boldsymbol{E}(I)$$

if we postulate no epistatic effects and no cohort environment.

6.1.2   Maternal effects

Equation 1 is not the only conceptual division of individual phenotypic value. In mammals, the genotype and the environment of the mother can influence an individual's phenotype, in addition to its own genotype and environment. Willham(1963) [30] developed the following partitioning

$$\boldsymbol{P}(I) = \boldsymbol{G}(I) + \boldsymbol{E}(I) + \boldsymbol{G}(M) + \boldsymbol{E}(M) \tag{2}$$

Here we can estimate components of $\boldsymbol{G}(I)$ and $\boldsymbol{E}(I)$, as in Table 1, but we can also estimate components of $\boldsymbol{G}(M)$ and $\boldsymbol{E}(M)$, as defined in Table 2.

All the components of value in Table 2 are independent and additive, so we can write, for example

Table 1: Definition of estimable causal components of individual genetic value and individual environmental value for autosomal loci

Component	Name	Meaning
$\boldsymbol{G}(Ia)$	additive genetic effect or breeding value	sum of average effects on phenotype of an individual's genes
$\boldsymbol{G}(Id)$	dominance effect	sum of deviations from average effect due to non-additive combination of gene effects within a locus
$\boldsymbol{G}(Ia:a)$	additive x additive epistatic effect	non-additive combination of additive gene effects across pairs of loci
$\boldsymbol{G}(Ia:d)$	additive x dominance epistatic effect	non-additive combination of additive and dominance gene effects across pairs of loci
$\boldsymbol{G}(Id:d)$	dominance x dominance epistatic effect	non-additive combination of dominance gene effects across pairs of loci
$\boldsymbol{E}(I)$	environmental deviation	unexplained deviation of individual phenotype (assumed environmental, developmental, or measurement error)
$\boldsymbol{E}(C)$	cohort environmental effect	environmental deviation due to individual belonging to a specific cohort

Table 2: Definition of estimable causal components of maternal genetic value and maternal environmental value for autosomal loci

Component	Name	Meaning
$\boldsymbol{G}(Ma)$	maternal additive genetic effect or breeding value	sum of average effects on phenotype of an individual's dam's genes
$\boldsymbol{G}(Md)$	maternal dominance effect	sum of deviations from average effect due to non-additive combination of gene effects within a dam's locus
$\boldsymbol{G}(Ma:a)$	maternal additive x additive epistatic effect	non-additive combination of additive gene effects across pairs of dam's loci
$\boldsymbol{G}(Ma:d)$	maternal additive x dominance epistatic effect	non-additive combination of additive and dominance gene effects across pairs of dam's loci
$\boldsymbol{G}(Md:d)$	matermnal dominance x dominance epistatic effect	non-additive combination of dominance gene effects across pairs of dam's loci
$\boldsymbol{E}(M)$	maternal environmental deviation	unexplained deviation of individual phenotype (assumed maternal environmental, or maternal developmental)
$\boldsymbol{E}(M\&C)$	within cohort maternal environmental deviation	unexplained deviation for individuals with same dam and same cohort
$\boldsymbol{E}(M\&!C)$	cross cohort maternal environmental deviation	unexplained deviation for individuals with same dam and different cohort

Table 3: Definition of estimable causal components of individual and maternal genetic value for sex-linked loci

Component	Name	Meaning
$\boldsymbol{G}s(Ia)$	sex-linked additive genetic effect or breeding value	sum of average effects on phenotype of an individual's sex-linked genes
$\boldsymbol{G}s(Ma)$	sex-linked maternal additive genetic effect or breeding value	sum of average effects on phenotype of an individual's dam's sex-linked genes

$$\boldsymbol{G}(M) = \boldsymbol{G}(Ma) + \boldsymbol{G}(Ma:a)$$

if we postulate no maternal dominance effect and only maternal additive x additive epistasis. We do not define a $\boldsymbol{P}(M)$. Maternal effects are part of the individual's phenotype. The dam's own phenotype is something entirely different.

The two effects $\boldsymbol{E}(M\&C)$ and $\boldsymbol{E}(M\&!C)$ are a partitioning of $\boldsymbol{E}(M)$ along the lines suggested by Bijma(2006) [3].

6.1.3 Individual and maternal effects due to sex-linked genes

All of the individual and maternal genetic effects defined in Table 1 and Table 2 are from genes at autosomal loci. For genes located on the sex chromosome, effects are defined in the same way, but require different procedures for their estimation. Table 3 defines the individual and maternal additive sex-linked effects used by the current version of *dmm()*. It is possible to define dominance and epistatic sex-linked effects, but these are not currently used by *dmm()*. It is also possible for sex-linked genes to interact epistatically with autosomal genes.

The components defined in Table 3 are additive, but are not independent. In general, individual effects are not independent from maternal effects. We can write, for example

$$\boldsymbol{G}(M) = \boldsymbol{G}(Ma) + \boldsymbol{G}s(Ma)$$

if we postulate only additive genetic maternal effects that are both sex-linked and autosomal, or again

$$\boldsymbol{G}(I) = \boldsymbol{G}(Ia) + \boldsymbol{G}s(Ia)$$

for only individual additive genetic effects.

Table 4: Definition of estimable causal components of variation between maternal and paternal founder lines

Component	Name	Meaning
$\boldsymbol{G}lm(I)$	effect of maternal line	in mammals could be attributed to differences in cytoplasmic genetic material between maternal lines
$\boldsymbol{G}lp(I)$	effect of paternal line	in mammals could be attributed to differences in Y chromosome genes between paternal lines

6.1.4 Effects due to maternal founder line and paternal founder line

All of the effects defined in Table 1 and Table 2 are from genes at autosomal loci. The effects defined in Table 3 are from genes on the sex chromosome (X chromosome in mammals). There are other possible genetic effects. Other possibilities are cytoplasmic inheritance (via mitochondrial DNA in mammmals or plasmid DNA in plants), and Y chromosome genes (carried by males in mammals).

The way that these effects are analysed is to divide the pedigree into maternal (or paternal) founder lines; that is groups of individuals which trace back to one base generation female (or male). Table 4 defines maternal founder line and paternal founder line effects used in versions of $dmm()$ from dmm_2.1-4 onward.

The components defined in Table 4 are additive in their contribution to individual phenotype and are independent of all other effects.

6.1.5 Estimation of effects

There are techniques (commonly known as BLUP) for estimating any of the above genetic effects for each individual in a population, given adequate data, pedigree information, and estimates of genetic parameters. These techniques are not the province of $dmm()$. $dmm()$ is a procedure for estimation of genetic parameters. Genetic parameters depend on the variances (and sometimes covariances) of effects. So we proceed to these.

6.1.6 Variances of effects

For each of the effects defined in Tables 1, 2 , 3 and 4, there is a sample variance, and a population variance about which we might wish to make inferences. The statistical methods used by $dmm()$ to estimate population variance components are covered in section 6.2. Here we will just focus on the genetic aspects.

The dyadic model (equation 13) is a linear model with the variance components $\boldsymbol{\Gamma}$ as unknowns and the model matrix $\boldsymbol{W}$ defining the relationship of the dyadic observations $\boldsymbol{\Psi}$ to the unknown variance components. Each variance component corresponds to a column of model coefficients in $\boldsymbol{W}$. It is the set-

ting up of the column of coefficients for each variance component that requires a knowledge of genetics.

For any dyad (y_i, y_j) the expected value of its covariance $(y_i - X_i\alpha)(y_j - X_j\alpha)'$ is given by

$$\sum_c (r_c)_{i,j}\gamma_c \tag{3}$$

which is actually the sum of one row of $\boldsymbol{W}$ times one column of $\boldsymbol{\Gamma}$. So one element of one column of $\boldsymbol{W}$ is $(r_c)_{i,j}$; but not quite, there are statistical considerations requiring introduction of an $\boldsymbol{M}$ matrix, and we also need a $\boldsymbol{Z}$ matrix to map individuals observed to individuals in the pedigree (see equations 9).

The quantity $(r_c)_{i,j}$ is a relationship coefficient between individuals i and j. The type of relationship coefficient required depends on the variance component γ_c. For example if γ_c is individual additive genetic variance VarG(Ia) the relationship coefficient required is the additive genetic relationship between individuals i and j.

For the original definition of relationship coefficients see Wright(1922) [32]. For a modern account see one of the texts, Falconer(1961) [7] , Kempthorne(1957) [15], or Lynch and Walsh(1998) [17]. A matrix of relationship coefficients for all pairs of individuals in a pedigree is termed a relationship matrix. Relationship coefficients have a dual definition and interpretation, as a probability of identity by descent of certain paires of genes, and as a correlation between the genes present in gametes from two individuals. It is the correlation interpretation which enables us to write $(r_c)_{i,j}\gamma_c$ in equation 3. Expressions for what is commonly termed the *Covariance between Relatives* treat the relationship coefficient as a correlation. Equation 3 is actually an example of an expression for the covariance between relatives for a given set of components γ_c. For calculation of relationship matrices from pedigrees see Wolak(2012) [31]. The *dmm()* package uses Matthew Wolak's excellent package *nadiv* to compute relationship matrices.

Given the required relationship matrices, it is a simple matter to setup the $\boldsymbol{W}$ matrix for the dyadic model equations 13. The calculation of the column elements of $\boldsymbol{W}$ for each type of variance component is documented in Table 5.

The component names in column 1 of Table 5 are in the form used by *dmm()* to label output; their equivalent in the more familiar subscripted notation (eg $\sigma^2_{a_I}$ for the first item) should be obvious. The formulae in column 2 of Table 5 produce matrices each of which must be vectorized into a column of $\boldsymbol{W}$. The symbols in column 2 of Table 5 are defined as follows

$\boldsymbol{Z}_I$ Matrix mapping individuals with observations to individuals in the pedigree

$\boldsymbol{Z}_M$ Matrix mapping individuals with observations to their dams in the pedigree

$\boldsymbol{Z}_C$ Matrix mapping individuals with observations to cohorts

$\boldsymbol{Z}_{Lm}$ Matrix mapping individuals with observations to maternal founder line groups

Table 5: Calculation of $\boldsymbol{W}$ matrix column elements for variance components defined as the variance of the effects documented in Tables 1, 2 and 3

Component	Column of $\boldsymbol{W}$ matrix
VarG(Ia)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_A\boldsymbol{Z}_I'\boldsymbol{M}$
VarG(Id)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_D\boldsymbol{Z}_I'\boldsymbol{M}$
VarG(Ia:a)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_{AA}\boldsymbol{Z}_I'\boldsymbol{M}$
Varg(Ia:d)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_{AD}\boldsymbol{Z}_I'\boldsymbol{M}$
Varg(Id:d)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_{DD}\boldsymbol{Z}_I'\boldsymbol{M}$
VarE(I)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_E\boldsymbol{Z}_I'\boldsymbol{M}$
VarE(C)	$\boldsymbol{M}\boldsymbol{Z}_C\boldsymbol{Z}_C'\boldsymbol{M}$
VarG(Ma)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{R}_A\boldsymbol{Z}_M'\boldsymbol{M}$
VarG(Md)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{R}_D\boldsymbol{Z}_M'\boldsymbol{M}$
VarG(Ma:a)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{R}_{AA}\boldsymbol{Z}_M'\boldsymbol{M}$
VarG(Ma:d)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{R}_{AD}\boldsymbol{Z}_M'\boldsymbol{M}$
VarG(Md:d)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{R}_{DD}\boldsymbol{Z}_M'\boldsymbol{M}$
VarE(M)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{Z}_M'\boldsymbol{M}$
VarE(M&C)	$\boldsymbol{M}((\boldsymbol{Z}_M\boldsymbol{Z}_M')\&(\boldsymbol{Z}_C\boldsymbol{Z}_C'))\boldsymbol{M}$
VarE(M&!C)	$\boldsymbol{M}((\boldsymbol{Z}_M\boldsymbol{Z}_M')\&!(\boldsymbol{Z}_C\boldsymbol{Z}_C'))\boldsymbol{M}$
VarGs(Ia)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_S\boldsymbol{Z}_I'\boldsymbol{M}$
VarGs(Ma)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{R}_S\boldsymbol{Z}_M'\boldsymbol{M}$
VarGlm(I)	$\boldsymbol{M}\boldsymbol{Z}_{Lm}\boldsymbol{Z}_{Lm}'\boldsymbol{M}$
VarGlp(I)	$\boldsymbol{M}\boldsymbol{Z}_{Lp}\boldsymbol{Z}_{Lp}'\boldsymbol{M}$

$\boldsymbol{Z}_{Lp}$ Matrix mapping individuals with observations to paternal founder line groups

$\boldsymbol{R}_A$ Matrix of additive relationship coefficients

$\boldsymbol{R}_D$ Matrix of dominance relationship coefficients

$\boldsymbol{R}_{AA}$ Matrix of additive x additive epistatic relationship coefficients

$\boldsymbol{R}_{AD}$ Matrix of additive x dominance epistatic relationship coefficients

$\boldsymbol{R}_{DD}$ Matrix of dominance x dominance epistatic relationship coefficients

$\boldsymbol{R}_S$ Matrix of sex-linked additive relationship coefficients

$\boldsymbol{R}_E$ Matrix of environmental correlations for individuals

$\boldsymbol{M}$ Matrix which transforms observations to residuals, see section 6.2.2

When *dmm()* labels components as variances, as in Table 5, it means either single-trait-variances or same-effect-cross-trait covariances. This convention extends thruout the *dmm()* output, for example genetic parameters are labelled by the component from which they derive.

6.1.7 Covariances of effects

What we mean by covariances of effects is a covariance between two different effects, commonly called a cross-effect-covariance. There can be single-trait-cross-effect-covariances and cross-trait-cross-effect-covariances.

Given the effects defined in Tables 1, 2 and 3, the only covariances which have non-zero expectation are those between an individual effect and a maternal effect of the same type (eg CovG(Ia,Ma)). This is because additive, dominance, and epistatic effects are defined in such a way as to be independent.

The columns of $\boldsymbol{W}$ for cross-effect-covariances depend on relationship matrices in the same way as the variances of effects, and are defined (for all non-zero covariances) in Table 6.

Notice that in Table 6 the cross-effect-covariances occur in symmetric pairs (eg CovG(Ia,Ma) and CovG(Ma.Ia)). These pairs are identical for a single-trait-cross-effect-covariance, but usually differ for a cross-trait-cross-effect-covariance. Users of *dmm()* are therefore encouraged to put covariances into the dyadic model in symmetric pairs. This practice also ensures that components sum to phenotypic (co)variance.

When *dmm()* labels components as covariances, as in Table 6, it means either single-trait-cross-effect-covariances or cross-trait-cross-effect-covariances. This convention extends thruout the labelling of *dmm()* output, for example genetic correlations are labelled by the covariance component from which they derive.

Table 6: Calculation of $\boldsymbol{W}$ matrix column elements for covariance components defined as the covariance of the effects documented in Tables 1, 2 and 3

Component	Column of $\boldsymbol{W}$ matrix
CovG(Ia,Ma)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_A\boldsymbol{Z}'_M\boldsymbol{M}$
CovG(Ma,Ia)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{R}_A\boldsymbol{Z}'_I\boldsymbol{M}$
CovG(Id,Md)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_D\boldsymbol{Z}'_M\boldsymbol{M}$
CovG(Md,Id)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{R}_D\boldsymbol{Z}'_I\boldsymbol{M}$
CovG(Ia:a,Ma:a)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_{AA}\boldsymbol{Z}'_M\boldsymbol{M}$
CovG(Ma:a,Ia:a)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{R}_{AA}\boldsymbol{Z}'_I\boldsymbol{M}$
CovG(Ia:d,Ma:d)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_{AD}\boldsymbol{Z}'_M\boldsymbol{M}$
CovG(Ma:d,Ia:d)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{R}_{AD}\boldsymbol{Z}'_I\boldsymbol{M}$
Covg(Id:d,Md:d)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_{DD}\boldsymbol{Z}'_M\boldsymbol{M}$
Covg(Md:d,Id:d)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{R}_{DD}\boldsymbol{Z}'_I\boldsymbol{M}$
CovGs(Ia,Ma)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_S\boldsymbol{Z}'_M\boldsymbol{M}$
CovGs(Ma,Ia)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{R}_S\boldsymbol{Z}'_I\boldsymbol{M}$
CovE(I,M)	$\boldsymbol{M}\boldsymbol{Z}_I\boldsymbol{R}_E\boldsymbol{Z}'_M\boldsymbol{M}$
CovE(M,I)	$\boldsymbol{M}\boldsymbol{Z}_M\boldsymbol{R}_E\boldsymbol{Z}'_I\boldsymbol{M}$
CovE(I,M&!C)	$\boldsymbol{M}((\boldsymbol{Z}_I\boldsymbol{Z}'_M)\&!(\boldsymbol{Z}_C\boldsymbol{Z}'_C))\boldsymbol{M}$
CovE(M&!C,I)	$\boldsymbol{M}((\boldsymbol{Z}_M\boldsymbol{Z}'_I)\&!(\boldsymbol{Z}_C\boldsymbol{Z}'_C))\boldsymbol{M}$
CovE(I,M&C)	$\boldsymbol{M}((\boldsymbol{Z}_I\boldsymbol{Z}'_M)\&(\boldsymbol{Z}_C\boldsymbol{Z}'_C))\boldsymbol{M}$
CovE(M&C,I)	$\boldsymbol{M}((\boldsymbol{Z}_M\boldsymbol{Z}'_I)\&(\boldsymbol{Z}_C\boldsymbol{Z}'_C))\boldsymbol{M}$

6.2   Statistical models and variance component estimation

The models used by *dmm()* are heirarchical. At the first level we model observations on the ith individual or monad (y_i). At the second level we model observations on pairs of individuals or dyads (y_i, y_j). The first level models fixed effects and puts everything else into residual. The second level models residual (co)variances from the first level and breaks them into causal components corresponding to nominated genetic and environmental effects. We call the first level model a monadic model and the second level model a dyadic model. Some workers call the second level model a derived model.

6.2.1   First level - Monadic model

If $\mathbf{Y}$ is an $n \times l$ matrix of observations on n individuals with l traits observed on each, we write the mixed model

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\alpha} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon} \tag{4}$$

where

$\mathbf{X}$ is a design matrix $(n \times k)$

$\boldsymbol{\alpha}$ is a matrix of fixed effects $(k \times l)$ for k fixed effects and l traits

$\mathbf{Z}$ is an incidence matrix mapping n individuals with observations to m individuals in a pedigree

$\boldsymbol{\gamma}$ is a matrix of random effects associated with individuals in the pedigree $(m \times l)$

$\boldsymbol{\epsilon}$ is a matrix of residuals $(n \times l)$

We proceed by contracting equation 4 to

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\alpha} + \mathbf{R} \tag{5}$$

where

$$\mathbf{R} = \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

and we have $\mathcal{E}(\mathbf{R}) = \mathbf{0}$ and $\mathrm{Cov}(\mathbf{R}) = \mathcal{E}(\mathbf{RR'}) = \mathcal{E}(\mathbf{Z}\boldsymbol{\gamma}\boldsymbol{\gamma}'\mathbf{Z} + \boldsymbol{\epsilon}\boldsymbol{\epsilon}')$

We can estimate $\boldsymbol{\alpha}$ from equation 5 either by OLS ignoring the fact that the elements of $\mathbf{R}$ are correlated using $\hat{\boldsymbol{\alpha}}_{OLS} = (\mathbf{X'X})^{-1}\mathbf{X'y}$, or by GLS using estimates of $\boldsymbol{\gamma}\boldsymbol{\gamma}'$ and $\boldsymbol{\epsilon}\boldsymbol{\epsilon}'$ to approximate $\mathrm{Cov}(\mathbf{R})$

Before we can use GLS we need to get some estimates of $\boldsymbol{\gamma}\boldsymbol{\gamma}'$ and $\boldsymbol{\epsilon}\boldsymbol{\epsilon}'$. We do this at the second level .

6.2.2 Second level - Dyadic model

We rewrite equation 4 as

$$Y - X\alpha = Z\gamma + \epsilon = R \tag{6}$$

but we now wish to model dyads (i, j). The defining observation for a dyad is the covariance between the two individuals (i, j), which turns out to be just the product of the observations on the two individuals, appropriately adjusted for mean and fixed effects. So we model the dyadic observations as

$$(Y - X\alpha) \otimes (Y - X\alpha) = (Z\gamma + \epsilon) \otimes (Z\gamma + \epsilon) + \delta \tag{7}$$

where δ is a matrix of dyadic residuals. If we take expectations from equation 7 we get a set of equations

$$\mathcal{E}((Y - X\alpha) \otimes (Y - X\alpha)) = Z\mathcal{E}(\gamma\gamma')Z' + \mathcal{E}(\epsilon\epsilon') \tag{8}$$

which is a set of linear equations in $\mathcal{E}(\gamma\gamma')$ and $\mathcal{E}(\epsilon\epsilon')$. Typically, in the univariate case, $\mathcal{E}(\epsilon\epsilon')$ will be $I\sigma_E^2$ where σ_E^2 is the (unknown) variance of residuals and $\mathcal{E}(\gamma\gamma')$ might be $R_A\sigma_{Ia}^2$ where R_A is an additive genetic relationship matrix and σ_{Ia}^2 is the (unknown) individual additive genetic variance. In the multivariate case, $\mathcal{E}(\epsilon\epsilon')$ will be $I \otimes \Sigma_E$ where Σ_E is the (unknown) covariance matrix of residuals for multiple traits, and $\mathcal{E}(\gamma\gamma')$ might be $R_A \otimes \Sigma_{Ia}$ where R_A is an additive genetic relationship matrix and Σ_{Ia} is the (unknown) additive genetic covariance matrix for multiple traits.

If we substitute $\hat{\alpha}$ (from the monadic model) for α in equations 8, we need to allow for the estimate $\hat{\alpha}$ not being exactly the true α, and we do this by defining a matrix M such that

$$Y - X\hat{\alpha} = MY$$

We can compute M as $(I - H)$ where H is the 'hat' matrix from estimating $\hat{\alpha}$ from model 5. In the OLS case H is defined as $H = X(X'X)^-X'$ and can be obtained from the QR decomosition of X. In the GLS case M and H are more complex and are discussed later.

We can then write equations 8 as

$$(MY) \otimes (MY) = Vec(MZ\mathcal{E}(\gamma\gamma')Z'M') + Vec(M\mathcal{E}(\epsilon\epsilon')M') \tag{9}$$

Equations 9 are like 8, that is linear in $\mathcal{E}(\gamma\gamma')$ and $\mathcal{E}(\epsilon\epsilon')$, but unlike 8 they can be solved for $\mathcal{E}(\gamma\gamma')$ and $\mathcal{E}(\epsilon\epsilon')$ because we know M and Z.

We can break $\mathcal{E}(\gamma\gamma')$ and $\mathcal{E}(\epsilon\epsilon')$ into a correlation matrix and a (co)variance as in equation 3. This leads to a rewrite of equation 9 as

$$(MY) \otimes (MY) = Vec(MZ_I R_{Ia} Z_I' M') \otimes \Sigma_{Ia} + Vec(MIM') \otimes \Sigma_E \tag{10}$$

so we now have the two unknown components $\boldsymbol{\Sigma_{Ia}}$ and $\boldsymbol{\Sigma_E}$ separated. We can generalize to any number of components (v) by writing

$$(MY) \otimes (MY) = \sum_{c=1}^{v} Vec(MZ_cR_cZ_c'M') \otimes \Sigma_c \qquad (11)$$

The term $(MY) \otimes (MY)$ sets up a matrix $\boldsymbol{\Psi}$ of order $\boldsymbol{n^2 \times l^2}$ with each column consisting of all the squares and products of adjusted $\boldsymbol{Y}$ values, for one traitpair. There is a column for each pair of traits so we model same-trait-dyadic covariances and cross-trait-dyadic covariances.

The terms $\boldsymbol{Vec(MZ_cR_cZ_c'M') \otimes \Sigma_c}$ can be setup as an $\boldsymbol{n^2 \times v}$ matrix of coefficients $\boldsymbol{W}$ ($\boldsymbol{v}$ being the number of components), and a $\boldsymbol{v \times l^2}$ matrix $\boldsymbol{\Gamma}$ of unknowns, collapsing $\boldsymbol{\Sigma_c}$ into $\boldsymbol{\Gamma}$, so we can write

$$\boldsymbol{\Psi = W\Gamma} \qquad (12)$$

as equations 10 or 11 in matrix form, which we call the dyadic model equations (DME's) , or

$$\boldsymbol{\Psi = W\Gamma + \Delta} \qquad (13)$$

as the corresponding dyadic model, $\boldsymbol{\Delta}$ being the dyadic residuals.

Equations 12 or model 13 are quite general as $\boldsymbol{W}$ and $\boldsymbol{\Gamma}$ can encompass any set of (co)variance components $\boldsymbol{\Sigma_c}$ (not just $\boldsymbol{\mathcal{E}(\gamma\gamma')}$ and $\boldsymbol{\mathcal{E}(\epsilon\epsilon')}$), each component having a column of coefficients in $\boldsymbol{W}$ and a row of unknowns in $\boldsymbol{\Gamma}$.

The actual method of setting up $\boldsymbol{W}$ for these equations for each of the components available in $dmm()$ is documented in section 6.1.

Equations 12 or model 13 show that (co)variance component estimation reduces to a linear regression problem with the causal components as the estimated regression coefficients. We can therefore solve equations 12 by OLS, which amounts to a least squares fitting of the dyadic model 13 to the data embodied in all the dyadic covariances in $\boldsymbol{\Psi}$.

This procedure is similar to that proposed by Pukelsheim(1976) [24], which he termed a dispersion-mean model, and it it a happy coincidence that our package name dmm is also an acronym for Pukelsheim's terminology. Searle(1979) [25] shows that if $\boldsymbol{\hat{\alpha}}$ (or $\boldsymbol{M}$) are obtained by OLS, and equations 12 solved by OLS, then the resulting(co)variance component estimates are equivalent to MINQUE estimates.

6.2.3   First level - second level iteration

We can now go back to the monadic model at level 1 of the heirarchy, substituting for $\boldsymbol{\mathcal{E}(RR')}$ in model 5 using the estimates of $\boldsymbol{\Gamma}$ obtained from fitting model 13 as follows

$$\hat{\Omega} = \mathcal{E}(RR') = \sum_i Z_iR_iZ_i'\sigma_i^2$$

where R_i is the relevant relationship matrix for the ith component (or I in the case of σ_E^2).

Having an estimate of Ω (the covariance matrix of residuals R) we can estimate α using GLS by

$$\hat{\alpha_{GLS}} = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}Y$$

using $\hat{\Omega}$ in place of the unknown Ω.

Given $\hat{\alpha_{GLS}}$, we return to level 2 and refine the estimate of Γ by re-fitting model 13, but we need to obtain a new M matrix (say M_{GLS}) which differs from the M in equation 9 because using GLS has effectively changed $X'X$ into $X'\Omega^{-1}X$. The equation for M_{GLS} is

$$M_{GLS} = I - H_{GLS} = I - X(X'V^{-1}X)^{-}X'V^{-1}$$

where

$$H_{GLS} = H_{OLS}VH_{OLS}V^{-1}$$

and given this we can reconstruct equations 12 and re-estimate Γ.

This iteration can be repeated until convergence of estimates is achieved. Convergence is defined as achieving a state where the current round of estimates of $\hat{\alpha_{GLS}}$ and $\hat{\Gamma}$ differ by only a small amount from the previous round. Convergence may fail to occur; this indicates that the model is a poor representation of the data. Convergence is normally rapid (ie in less than 10 rounds). It is necessary to constrain the matrix $\hat{\Omega}$ to be positive definite, and this is achieved by constraining all component matrices ($\hat{\Gamma}$) to be positive definite.

Anderson(1984) [1] has shown that if $\hat{\alpha}$ (or M) are obtained by GLS, and then equations 12 solved by OLS, then the resulting (co)variance components are equivalent to ML estimates. However Anderson's derivation did not include the M matrix, ie he used equation 7 rather than equation 9. In $dmm()$ we use equation 9 and this effectively corrects the ML estimates of Γ for bias. That is it allows for the degrees of freedom used in fitting $\hat{\alpha}_{GLS}$ at the first level, before fitting $\hat{\Gamma}$ at the second level. We therefore call the GLS-b estimates from $dmm()$ 'bias-corrected-ML estimates'.

It is, in theory, possible to obtain REML estimates from $dmm()$. Searle, et al (1992) [26] show that if we use $\hat{\alpha}_{GLS}$ and then solve the resulting equations 12 by GLS instead of OLS, the resulting estimates of Γ are equivalent to REML or I-MINQUE. However, using GLS on equations 12 requires the covariance matrix of the dyadic residuals Δ, and this is an enormous array of the order of the fourth power of the number of individuals, so it is not computationally feasible to attempt REML estimates using the dmm approach.

One can show that the GLS-b estimates are bias corrected very simply. Make a simple dataset of 3 observations

```
> y <- c(1,2,3)
> Id <- y
> y.df <- data.frame(cbind(Id,y))
```

and use *dmm()* to obtain just a mean and a residual variance

```
> fit <- dmm(y.df,y ~ 1, components=c("VarE(I)"),gls=T)
> summary(fit,gls=T)
Call:
summary.dmm(dmmobj = fit, gls = T)


Coefficients fitted by OLS for fixed effects:

  Trait Estimate StdErr CI95lo CI95hi
1     y        2  0.577  0.868   3.13



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

        Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)       y:y        1  0.354  0.307   1.69
VarP(I)       y:y        1  0.354  0.307   1.69



Coefficients fitted by GLS for fixed effects:

  Trait Estimate StdErr CI95lo CI95hi
1     y        2  0.577  0.868   3.13



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

        Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)       y:y        1  0.354  0.307   1.69
VarP(I)       y:y        1  0.354  0.307   1.69

>
```

and we do indeed get a mean of **2** and a variance of **1**, not **0.66**, which is the uncorrected ML estimate of variance.

The M matrix for this simple case is

$$
\begin{pmatrix}
\frac{N-1}{N} & \frac{-1}{N} & \frac{-1}{N} \\[2mm]
\frac{-1}{N} & \frac{N-1}{N} & \frac{-1}{N} \\[2mm]
\frac{-1}{N} & \frac{-1}{N} & \frac{N-1}{N}
\end{pmatrix}
$$

and it can be shown that the quadratic form $y'My$ expands to $\sum_i y_i^2 - (1/N)\sum_{i,j} y_i y_j$ which is the dyadic version of the familiar formula for variance.

It is clear from this that the M matrix can be used to compute a correction for mean.

In more complex cases the M matrix simply adjusts the degrees of freedom appropriately, given the replication for all fixed effects embodied in the X matrix. This adjustment is not the same as the adjustment made by REML. The bias-corrected ML estimates obtained from *dmm()* will agree with REML estimates in some cases, but not always. This is because REML accounts for the covariances of dyadic residuals (Δ in equation 13), while *dmm()* assumes these residuals are uncorrelated (ie it uses OLS on equation 13).

The approach taken by *dmm()* also has similarities with the 'symmetric sums method' of Koch(1967) [16]. Koch was probably the first worker to equate dyadic covariances to their expectations, as in equation 7. The difference is that Koch summed his expectations (he could hardly have done anything else, given computing facilities at the time), while *dmm()* writes all expectations as a set of equations 12 to be solved directly.

There is some more detail of the setting up and solving of dyadic model equations in the document *solvedme.pdf* [12]

6.2.4 Standard errors of variance component estimates

Using a dyadic model reduces variance component estimation to a multiple linear regression problem, the components being estimated as regression coefficients. Therefore standard errors for variance component estimates can be computed using expressions for the standard error of a regresssion coefficient. In terms of model 13 the unknowns Γ are defined (if estimated by OLS) as

$$\Gamma = (W'W)^{-1}W'\Psi$$

but are not actually computed that way, and their standard errors are obtained from

$$\mathrm{Cov}(\Gamma) = (W'W)^{-1} \otimes \Sigma$$

where $\Sigma = \Delta\Delta'/(n^2 - v)$ is the $(l \times l)$ covariance matrix of residuals from model 13. Again $(W'W)^{-1}$ is not computed that way, but is obtained from the QR decomposition of W.

Because these expressions are exact, given the usual assumptions in regression models, the standard errors of component estimates from *dmm()* are likely to be smaller than those obtained other procedures which employ numerical approximations. This is born out in the examples.

6.2.5 Problems with variance component estimation

A common problem in partitioning residual variance after fixed effects are fitted into a number of variance components is to encounter the following message from *dmm()*.

73

```
Error in dyad.am.expect(am, gls, dmeopt) :
  Dyadic model equations not of full rank:
   either omit some components or try dmeopt='pcr'
```

This means that the set of components fitted is more than can be estimated from the data. There are a number of ways of dealing with this situation

- omit some of the components, rerun dmm() and see if the problem disappears

- start with a minimal set of components (eg components=c("VarE(I)","VarG(Ia)")), and sequentially add components until the problem occurs

- use dmeopt="pcr" to force dmm() to produce an output object, then look at the correlations among the columns of the dyadic model equations matrix (W matrix) using *print(dmmobject$dme.correl)*

- as from dmm_2.1-3 a dmm output object will be returned when the 'not of full rank' message occurs, so there is no need to use dmeopt="pcr", you can inspect the correlations of the columns of the W matrix directly with *print(dmmobject$dme.correl)*. As from dmm_2.1-5 the print method for objects of class *dmm* will print the above correlations of the columns of the W matrix, at the end of its usual output.

Deciding which components can be estimated from a particular pedigree is not always easy. There are some general rules

- to partition individual variation into VarE(I) and VarG(Ia) there must be some individuals in the pedigree which are related (ie have a relationship coefficient other than zero). If the relationship matrix is an identity (all individuals unrelated) the above partitioning is impossible. In general this means that at least a 2 generation pedigree is required to partition individual effects.

- to partition maternal variation into VarE(M) and VarG(Ma) there must be some dams in the pedigree which are related. In general this means that at least a 3 generation pedigree is required to partition maternal effects.

- to partition sexlinked genetic variation the must be individuals of both sexex present in the pedigree and having observations.

The easiest way to see if a particular pedigree design will estimate the components one desires is to construct an artificial dataset with that pedigree and attempt to analyse it.

I would like to thank Paloma Moran Martinez for bringing these issues to my attention.

6.3 Genetic parameters and their standard errors

Strictly speaking, any set of parameters which quantify genetic and environmental variation in a set of traits can be called genetic parameters. So causal variance and covariance components are genetic parameters. However, there is a convention that 'genetic parameters' refers to heritabilities and genetic correlations (and for completeness phenotypic (co)variance). Geneticists seem to prefer standardized unitless quantities to describe their populations.

When people say 'heritability' , without qualification, they imply individual additive genetic variation as a proportion of total phenotypic variation. Similarly an unqualified 'genetic correlation' implies the correlation based on the individual additive genetic covariance. There are, of coarse, equivalent parameters for the non-additive, sex-linked, and maternal components. We avoid using qualifiers in *dmm()* by using the terms *proportion* for each variance component as a proportion of the total or phenotypic variance, and *correlation* for each covariance component scaled by the appropriate variances to a correlation. We use the component names defined in Tables 5 and 6 as qualifying labels.

So *proportion* is defined as

$$\frac{VarX(Yz)}{VarP(I)}$$

for component X(Yz), and *correlation* for traits T1 and T2 is defined as

$$\frac{CovX(Yz[T1], Yz[T2])}{\sqrt{VarX(Yz[T1]VarX(Yz[T2])}}$$

for cross-trait-same-effect-covariances, and

$$\frac{CovX(Yz[T1], Wv[T1])}{\sqrt{VarX(Yz[T1]VarX(Wv[T1])}}$$

for same-trait-cross-effect-covariances for effects X(Yz) and X(Wv), and

$$\frac{CovX(Yz[T1], Wv[T2])}{\sqrt{VarX(Yz[T1]VarX(Wv[T2])}}$$

for cross-trait-cross-effect-covariances.

So genetic parameter estimates are a simple (but nonlinear) transform of variance and covariance component estimates. Their sampling variances (and hence standard errors) can therefore be obtained from the sampling variances of the components. The usual method of obtained these is known as the *delta method* [2]. This approach computes the sampling variance of a polynomial approximation to the transform. Typically a Taylor series expansion truncated to two terms is used as as the approximation. In *dmm(()* we develop a different approach, using logarithms to linearize the function as outlined below.

We need the following identities

$$Var(\ln X) = \ln(1 + CV_X^2)$$

which is exact if the distribution of X is lognormal, and approximate otherwise, and is also approximated by CV_X^2 if CV^2 is small.

We also need the bivariate equivalent

$$Cov(\ln X, \ln Y) = \ln(1 + CC_{X,Y}^2)$$

where $CC_{X,Y}^2 = \frac{Cov(X,Y)}{\overline{XY}} = r_{X,Y} CV_X CV_Y$

In the present application X and Y are variances, which have a χ^2 distribution. This does not necessarily mean that the above identities are being misapplied, because the lognormal distribution can be used as an approximation to the χ^2 distribution, for example see Jouini etal (2011) [14].

6.3.1 Sampling variance of a proportion

Let $Z = X/Y$ be the proportion. Apply the log transform to linearize

$$\ln Z = \ln X - \ln Y$$

and write as

$$z = x - y$$

Do the variance of the linear transform

$$
\begin{aligned}
Var(z) &= Var(x - y) \\
&= Var(x) + Var(y) - 2Cov(x, y) \\
&= \ln(1 + CV_X^2) + \ln(1 + CV_Y^2) + 2\ln(1 + CC_{X,Y}^2)
\end{aligned}
$$

then reverse the transform with the exp function

$$
\begin{aligned}
Var(Z) &= \bar{Z}^2(\exp[Var(lnZ) - 1) \\
&= \bar{Z}^2(\exp[Var(z) - 1) \\
&= (\frac{\bar{X}}{Y})^2(\exp[\ln(1 + CV_X^2) + \ln(1 + CV_Y^2) + 2\ln(1 + CC_{X,Y}^2)])
\end{aligned}
$$

This looks complex but is readily programmed in R as a set of nested function calls.

6.3.2   Sampling variance of a correlation

Let $Z = W/(XY)^{0.5}$ be the correlation. Apply the log transform to linearize

$$\ln Z = \ln W - 0.5\ln X - 0.5\ln Y$$

nd write as

$$z = w - 0.5x - 0.5y$$

Do the variance of the linear transform

$$
\begin{aligned}
Var(z) &= Var(w) - 0.25Var(x) - 0.25Var(y)\\
&\quad -2Cov(w,z) - 2Cov(w,y) + 0.5Cov(x,y)\\
&= \ln(1 + CV_W^2) + 0.25\ln(1 + CV_X^2) + 0.25\ln(1 + CV_Y^2)\\
&\quad - \ln(1 + CC_{W,X}^2) - \ln(1 + CC_{W,Y}^2) + 0.5\ln(1 + CC_{X,Y}^2)
\end{aligned}
$$

then reverse the transform with the exp function

$$
\begin{aligned}
Var(Z) &= \bar{Z}^2(\exp[Var(lnZ)] - 1)\\
&= \bar{Z}^2(\exp[Var(z)] - 1)\\
&= (\frac{\bar{W}}{(XY)^{0.5}})^2(\exp[\ln(1 + CV_W^2) + 0.25\ln(1 + CV_X^2) + 0.25\ln(1 + CV_Y^2)\\
&\quad - \ln(1 + CC_{W,X}^2)]) - \ln(1 + CC_{W,Y}^2)]) + 0.5\ln(1 + CC_{X,Y}^2)])
\end{aligned}
$$

6.3.3   Problems with genetic parameter calculation

To calculate the genetic parameters corresponding to a cross-effect covariance, *dmm(()* requires that the corresponding variance components are also estimated. For example, for CovG(Ia,Ma) the corresponding variances (VarG(Ia) and VarG(Ma)) must also be estimated, and so must appear in the *components=* argument for *dmm()*. If this is not done, you will get error messages similar to

```
....
DME substep completed:
Error in varcomp[var2, ii] : subscript out of bounds
>
```

nd execution will terminate. No output will be returned.

It does not make sense to estimate a cross-effect-covariance without simultaneously estimating the corresponding variances. If two effects have a covariance, they must also have variances. All variances and covariances that are significantly different from zero should be in the dyadic model used to partition residual variance. If a large and significant variance or covariance is omitted, estimates for all the other included variances and covariances are biased. This is known as the 'omitted variable effect'. It is well known is a regression context, but it applies also to partitioning of variance components.

6.4 Genetic response to selection

The function *gresponse()* converts phenotypic selection differentials to genetic selection differentials. It requires a set of genetic parameter estimates, in the form of a *dmm* object as computed by *dmm()*.

6.4.1 Understanding selection differentials

Consider the mechanics of selection. The selection process chooses a subset of individuals which are to be parents of the next generation. The selected subgroup will presumably have some differences from the whole group of individuals from which they were chosen. The difference may be univariate (ie one measured trait differs) or multivariate (any number of traits differ). Such a difference is referred to as a *phenotypic selection differential*, and thus may be a single value or a vector of differences. It is called 'phenotypic' because it is the average observed difference of selected from unselected individuals.

The function *gresponse()* assumes that phenotypic selection differentials are known and are specified in the same units of measurement as the traits used to compute genetic parameters with *dmm()*. It is outside of the scope of *gresponse()* to calculate expected selection differentials for various types and intensities of selection.

When selected parents are mated and produce progeny there will be an observable performance (for one or more traits) of the progeny of selected parents. If we also mated (experimantally) some unselected parents (ie chosen at random from the whole population) their progeny would also have an observable performance.. The average observed difference between the progeny of selected parents and the progeny of unselected parents, would be a *genetic selection differential*. or more precisely a realised genetic selection differential. It is called 'genetic' because it represents the part of the parent's phenotypic selection differential which is inherited by their progeny.

One of the main uses of genetic parameters is to predict what the genetic selection differential would be, for a given phenotypic selection differential, without having to carry out the above experimental matings. This is what *gresponse()* does. The term 'response to selection' refers to achieving a certain genetic selection differential, given a specific phenotypic selection differential.

Most of the predictable response to selection is due to additive genetic variation. Therefor *gresponse()* deals with two important cases, response due to individual additive genetic variation, and response due to maternal additive genetic variation. In each case the additive genetic variation can be autosomal or sexlinked or both, leading to four additive genetic effects which can be considered in any combination. The following four sections deal with a gradual development of the response prediction equations leading up to the general case with all four effects.

6.4.2 Response due to individual additive genetic variation

If there are two traits, X and Y, the equation for response due to individual additive genetic variation is

$$
\left[\begin{array}{c} d_{G_X} \\ d_{G_Y} \end{array}\right] = \left[\begin{array}{cc} \sigma^2_{G_X} & \sigma_{G_{XY}} \\ \sigma_{G_{XY}} & \sigma^2_{G_Y} \end{array}\right] \left[\begin{array}{cc} \sigma^2_{P_X} & \sigma_{P_{XY}} \\ \sigma_{P_{XY}} & \sigma^2_{P_Y} \end{array}\right]^{-1} \left[\begin{array}{c} d_{P_X} \\ d_{P_Y} \end{array}\right]
$$

where

d_{P_X} is the phenotypic selection differential for trait x

d_{G_X} is the genetic selection differential for trait x

$\sigma^2_{G_X}$ is the additive genetic variance for trait X

$\sigma_{G_{XY}}$ is the additive genetic covariance for traits X and Y

$\sigma^2_{G_Y}$ is the additive genetic variance for trait y

$\sigma^2_{P_X}$ is the phenotypic variance for trait X

$\sigma_{P_{XY}}$ is the phenotypic covariance for traits X and Y

$\sigma^2_{P_Y}$ is the phenotypic variance for trait Y

For multiple traits this is readily generalized to

$$
d_G = G P^{-1} d_P \tag{14}
$$

where

d_G is a vector of genetic selection differentials

d_P is a vector of phenotypic selection differentials

G is an additive genetic covariance matrix

P is a phenotypic covariance matrix

and for a single trait it reduces to the familiar breeder's equation

$$
d_G = d_P h^2
$$

where h^2 is additive genetic heritability

6.4.3 Response due to individual and maternal additive genetic variation

We start by rewriting equation 2 using a slightly different notation and a 'prime' superscript to indicate parent generation.

$$p = g_{I_a} + e_I + g'_{M_a} + e'_M \tag{15}$$

because we have to be clear now that the genetic components of phenotype p are $g_{I_a} + g'_{M_a}$, but the individual's own genotypic value is $g_{I_a} + g_{M_a}$

If the above equation is for a progeny generation, we have for the parent generation

$$p' = g'_{I_a} + e_I + g''_{M_a} + e''_M \tag{16}$$

and we are looking at predicting $\begin{bmatrix} d_{g_{I_a}} \\ d_{g'_{M_a}} \end{bmatrix}$ from $d_{p'}$.

For a single trait, this prediction can be written

$$\begin{bmatrix} d_{g_{I_a}} \\ d_{g'_{M_a}} \end{bmatrix} = \begin{bmatrix} G_{I_a I_a} & G_{I_a M'_a} \\ G_{I_a M'_a} & G_{M'_a M'_a} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} [\, P \,]^{-1} [\, d_{p'} \,]$$

but because covariance component estimates from $dmm()$ are of $G_{I_a M_a}$ and $G_{M_a M_a}$ rather than $G_{I_a M'_a}$ and $G_{M'_a M'_a}$ we rewrite it as

$$\begin{bmatrix} d_{g_{I_a}} \\ d_{g'_{M_a}} \end{bmatrix} = \begin{bmatrix} G_{I_a I_a} & G_{I_a M_a} \\ G_{I_a M'_a} & G_{M_a M_a} \end{bmatrix} \begin{bmatrix} R_{II} \\ R_{IM} \end{bmatrix} [\, P \,]^{-1} [\, d_{p'} \,]$$

where R_{II} is 1, and R_{IM} is the relationship between individual and parent, that is 0.5.

This single trait case reduces to Dickerson's(1947) [5] formula

$$d_G = (h_I^2 + \frac{3}{2} r_{G_{IM}} h_I h_M + \frac{1}{2} h_M^2) d_P$$

where d_G in Dickerson's notation is the sum of our $d_{g_{I_a}}$ and $d_{g'_{M_a}}$.

For two traits, X and Y, we can write

$$\begin{bmatrix} d_{g_{I_X}} \\ d_{g_{I_Y}} \\ d_{g'_{M_X}} \\ d_{g'_{M_Y}} \end{bmatrix} = \begin{bmatrix} \sigma^2_{G_{I_X}} & \sigma_{G_{I_{XY}}} & \sigma_{G_{I_X M_X}} & \sigma_{G_{I_X M_Y}} \\ \sigma_{G_{I_{XY}}} & \sigma^2_{G_{I_Y}} & \sigma_{G_{I_Y M_X}} & \sigma_{G_{I_Y M_Y}} \\ \sigma_{G_{I_X M_X}} & \sigma_{G_{I_X M_Y}} & \sigma^2_{G_{M_X}} & \sigma_{G_{M_{XY}}} \\ \sigma_{G_{I_Y M_X}} & \sigma_{G_{I_Y M_Y}} & \sigma_{G_{M_{XY}}} & \sigma^2_{G_{M_Y}} \end{bmatrix} \begin{bmatrix} R_{II} \\ R_{IM} \end{bmatrix} \begin{bmatrix} \sigma^2_{P_X} & \sigma_{P_{XY}} \\ \sigma_{P_{XY}} & \sigma^2_{P_Y} \end{bmatrix}^{-1} \begin{bmatrix} d_{p'_X} \\ d_{p'_Y} \end{bmatrix}$$

where here R_{II} is I_2 (an identity matrix of order 2), and R_{IM} is $0.5 I_2$.

For multiple traits this generalizes to

$$d_G = GRP^{-1} d_P \tag{17}$$

80

where G is the combined genetic covariance matrix of individual and maternal effects, and R is a $2l \times l$ matrix $\begin{bmatrix} I_l \\ 0.5I_l \end{bmatrix}$

where l is number of traits.

There remains the issue of phenotypic response in the progeny generation not being the same as the genetic response, when there are maternal effects (Mueller and James(1985) [23]). This is not currently considered by *gresponse()*. It involves the "CovE(I,M)" environmental covariance of individual and maternal effects.

6.4.4 Response due to individual additive autosomal and sexlinked genetic variation

If some of the genes affecting a trait are on the X chromosome, and some are autosomal, the appropriate effects model is

$$p = g_{I_a} + g_{s_{I_a}} + e_I \tag{18}$$

In this case there are two sorts of individual additive genetic variance (VarG(Ia) and VarGs(Ia) in our R notation). Their respective contributions to response to selection was first obtained by Griffin(1966) [10]. We first look at Griffin's equations, in his original notation. Change in female progeny mean due to mass selection among males is

$$(\bar{i}_m / {}_m\sigma_{ind})[{}_{fm}\sigma_{As} + 0.5({}_{fm}\sigma_{Aa}) + 0.5({}_{fm}\sigma_{AA})] \tag{19}$$

where

$\bar{i}_m$ is the standardised phenotypic selection differential for males

${}_m\sigma_{ind}$ is the phenotypic standard deviation for males

${}_{fm}\sigma_{As}$ is the additive sexlinked genetic covariance between females and males

${}_{fm}\sigma_{Aa}$ is the additive autosomal genetic covariance between females and males

${}_{fm}\sigma_{AA}$ is the epistatic genetic covariance between females and males (both autosomal and sexlinked)

Note the unusual leading subscript notation.

Similarly the change in female progeny mean due to mass selection among females is

$$(\bar{i}_f / {}_f\sigma_{ind})[0.5({}_{ff}\sigma_{As}) + 0.5({}_{ff}\sigma_{Aa}) + 0.25({}_{ff}\sigma_{AA})] \tag{20}$$

with obvious notation.

Changes in the male progeny mean due to mass selection among males are

$$(\bar{i}_m / {}_m\sigma_{ind})[0.5({}_{mm}\sigma_{Aa})] \tag{21}$$

81

so the sexlinked additive genetic covariance $(\boldsymbol{\sigma_{As}})$ does not enter into response in males to selection of males, because a male parent cannot pass an X chromosome onto its male progeny.

And, finally, change in male progeny mean, due to mass selection among females is

$$(\bar{i}_f/_f\sigma_{ind})[(_{fm}\sigma_{As}) + 0.5(_{fm}\sigma_{Aa}) + 0.5(_{fm}\sigma_{AA})] \qquad (22)$$

that is, the same as the male to female path, except for the selection differential.

There are a number of points to note, before we adapt these equations to our current purpose.

- Griffin is using sex-specific genetic parameters. His $_{ff}\boldsymbol{\sigma_{Aa}}$ is autosomal additive genetic variance in females, and $_{fm}\boldsymbol{\sigma_{Aa}}$ is a cross-sex additive genetic covariance. We use the same parameters for both sexes so we write

$$_{mm}\boldsymbol{\sigma_{Aa}} =_{ff} \boldsymbol{\sigma_{Aa}} =_{fm} \boldsymbol{\sigma_{Aa}} = VarG(Ia)$$

  and

$$_{mm}\boldsymbol{\sigma_{As}} =_{ff} \boldsymbol{\sigma_{As}} =_{fm} \boldsymbol{\sigma_{As}} = VarGs(Ia)$$

  and

$$_{m}\boldsymbol{\sigma_{ind}} =_{f} \boldsymbol{\sigma_{ind}} = (VarP(I))^{0.5} = \sigma_P$$

- We will not use the epistatic $\boldsymbol{\sigma_{AA}}$ component.

- Our $\boldsymbol{d_P}$ is a phenotypic selection differential in real units (ie not standardized). So

$$\overline{i_m}/(_{m}\sigma_{ind}) = d_{P_{he}}/\sigma_P^2$$

$$\overline{i_f}/(_{f}\sigma_{ind}) = d_{P_{ho}}/\sigma_P^2$$

  The $\boldsymbol{he}$ and $\boldsymbol{ho}$ subscripts refer to heterogametic and homogametic sexes (see below). We will take the $\boldsymbol{\sigma_P^2}$ denominator inside the square brackets and use it to turn the genetic variance components into heritabilities, for example the female to male equation can be written

$$d_{P_{ho}}[0.5h_{As}^2 + 0.5h_{Aa}^2]$$

- Griffin used $\boldsymbol{m}$ and $\boldsymbol{f}$ for male and female and assumed the male sex to be hetrogametic, as in mammals. We will generalize slightly, as foreshadowed above, and use $\boldsymbol{he}$ for the heteogametic sex and $\boldsymbol{ho}$ for the homogametic sex.

- Griffin's four equations represent four paths of genetic change. We will write these as $\boldsymbol{he.he}$, $\boldsymbol{ho.he}$, $\boldsymbol{he.ho}$, and $\boldsymbol{ho.ho}$, the first ($\boldsymbol{he}$ or $\boldsymbol{ho}$) representing the parent sex, and the last the progeny sex.

So taking taking all the above points together we rewrite Griffin's 4 equations as

$$
\begin{aligned}
d_{G_{he.he}} &= d_{P_{he}}[0h^2_{As} + 0.5h^2_{Aa}] \\
d_{G_{ho.he}} &= d_{P_{ho}}[h^2_{As} + 0.5h^2_{Aa}] \\
d_{G_{he.ho}} &= d_{P_{he}}[h^2_{As} + 0.5h^2_{Aa}] \\
d_{G_{ho.ho}} &= d_{P_{ho}}[0.5h^2_{As} + 0.5h^2_{Aa}]
\end{aligned}
$$

where

$$
h^2_{As} = \frac{VarGs(Ia)}{VarP(I)}
$$

$$
h^2_{Aa} = \frac{VarG(Ia)}{VarP(I)}
$$

These are, of course single trait equations. To get multivariate equations we simply replace each heritability by GP^{-1}, where G is the relevant genetic covariance matrix and P is the phenotypic covariance matrix. This leads to

$$
\begin{aligned}
d_{G_{he.he}} &= [OG_{S_{Ia}} + 0.5IG_{Ia}]P^{-1}d_{P_{he}} \\
d_{G_{ho.he}} &= [IG_{S_{Ia}} + 0.5IG_{Ia}]P^{-1}d_{P_{ho}} \\
d_{G_{he.ho}} &= [IG_{S_{Ia}} + 0.5IG_{Ia}]P^{-1}d_{P_{he}} \\
d_{G_{ho.ho}} &= [0.5IG_{S_{Ia}} + 0.5IG_{Ia}]P^{-1}d_{P_{ho}}
\end{aligned}
$$

where O is a matrix of zeros of order $l \times l$ and I is an identity matrix of order $l \times l$, l being the number of traits. The phenotypic selection differentials d_P and genetic selection differentials d_G, are now vectors of length l.

If we form the coefficients into matrices

$$
S_{he.he} = \begin{bmatrix} 0.5I \\ O \end{bmatrix}
$$

$$
S_{ho.he} = \begin{bmatrix} 0.5I \\ I \end{bmatrix}
$$

$$
S_{he.ho} = \begin{bmatrix} 0.5I \\ I \end{bmatrix}
$$

$$
S_{ho.ho} = \begin{bmatrix} 0.5I \\ 0.5I \end{bmatrix}
$$

and form a partitioned G matrix

$$
G = \begin{bmatrix} G_{Ia} & O \\ O & G_{S_{Ia}} \end{bmatrix}
$$

we can then write

$$
\begin{aligned}
d_{G_{he.he}} &= GS_{he.he}P^{-1}d_{P_{he}} \\
d_{G_{ho.he}} &= GS_{ho.he}P^{-1}d_{P_{ho}} \\
d_{G_{he.ho}} &= GS_{he.ho}P^{-1}d_{P_{he}} \\
d_{G_{ho.ho}} &= GS_{ho.ho}P^{-1}d_{P_{ho}}
\end{aligned}
\tag{23}
$$

and we have 4 matrix equations predicting genetic selection differentials for each of the 4 paths separately. Each d_G is a vector with partitions containing the responses due to G_{Ia} and $G_{S_{Ia}}$, in all traits.

These path specific genetic selection differentials can be summed to obtain sex-specific genetic selection differentials

$$
\begin{aligned}
d_{G_{.he}} &= d_{G_{he.he}} + d_{G_{ho.he}} \\
d_{G_{.ho}} &= d_{G_{ho.ho}} + d_{G_{he.ho}}
\end{aligned}
\tag{24}
$$

and these sex specific responses can be averaged to obtain an overall genetic selection differential

$$
d_{G_{..}} = 0.5(d_{G_{.he}} + d_{G_{.ho}})
\tag{25}
$$

We can also sum the G_{Ia} and $G_{S_{Ia}}$ contributions within any of the above d_G's to obtain a total response due to both autosomal and sexlinked genetic variation, as is done in Griffin's equations.

6.4.5 Combined responses due to individual and maternal additive genetic variation, both autosomal and sexlinked

We need to put together a set of equations which will predict responses due to individual additive, individual additive sexlinked, maternal additive, and maternal additive sexlinked genetic variation. We now work with the following model for effects

$$
p = g_{Ia} + g_{S_{Ia}} + e_I + g'_{M_a} + g'_{S_{M_a}} + e'_M
$$

the prime indicating parent generation.

The concept of a partitioned G matrix needs to be extended to 4 partitions as follows

$$
G = \begin{bmatrix}
G_{IaIa} & O & G_{IaMa} & O \\
O & G_{S_{IaIa}} & O & G_{S_{IaMa}} \\
G_{MaIa} & O & G_{MaMa} & O \\
O & G_{S_{MaIa}} & O & G_{S_{MaMa}}
\end{bmatrix}
$$

We must also extend the $\boldsymbol{R}$ matrix if section 6.4.3 which deals with mother x offspring relationships, and the $\boldsymbol{S}$ matrix of section 6.4.4 which deals with autosomal and sexlinked inheritance. We put both the $\boldsymbol{R}$ and $\boldsymbol{S}$ matrix coefficients into a combined matrix $\boldsymbol{T}$ which is defined as

$$\boldsymbol{T}_{he.he} = \begin{bmatrix} \mathbf{0.5I} \\ \boldsymbol{O} \\ \mathbf{0.25I} \\ \boldsymbol{O} \end{bmatrix}$$

$$\boldsymbol{T}_{ho.he} = \begin{bmatrix} \mathbf{0.5I} \\ \boldsymbol{I} \\ \mathbf{0.25I} \\ \mathbf{0.5I} \end{bmatrix}$$

$$\boldsymbol{T}_{he.ho} = \begin{bmatrix} \mathbf{0.5I} \\ \boldsymbol{I} \\ \mathbf{0.25I} \\ \mathbf{0.5I} \end{bmatrix}$$

$$\boldsymbol{T}_{ho.ho} = \begin{bmatrix} \mathbf{0.5I} \\ \mathbf{0.5I} \\ \mathbf{0.25I} \\ \mathbf{0.25I} \end{bmatrix}$$

where $\boldsymbol{I}$ is an identity matrix of order $\boldsymbol{l}$, and $\boldsymbol{O}$ is a matrix of zeros of order $\boldsymbol{l}$, $\boldsymbol{l}$ being the number of traits.

We can then write the combined equations as

$$\begin{aligned} \boldsymbol{d_{G_{he.he}}} &= \boldsymbol{G}\boldsymbol{T}_{he.he}\boldsymbol{P}^{-1}\boldsymbol{d_{P_{he.he}}} \\ \boldsymbol{d_{G_{ho.he}}} &= \boldsymbol{G}\boldsymbol{T}_{ho.he}\boldsymbol{P}^{-1}\boldsymbol{d_{P_{ho.he}}} \\ \boldsymbol{d_{G_{he.ho}}} &= \boldsymbol{G}\boldsymbol{T}_{he.ho}\boldsymbol{P}^{-1}\boldsymbol{d_{P_{he.ho}}} \\ \boldsymbol{d_{G_{ho.ho}}} &= \boldsymbol{G}\boldsymbol{T}_{ho.ho}\boldsymbol{P}^{-1}\boldsymbol{d_{P_{ho.ho}}} \end{aligned} \qquad (26)$$

which look similar to equations 23 but $\boldsymbol{G}$ now has 4 partitions as defined above, and the $\boldsymbol{d_G}$'s each have 4 matching partitions for the 4 modes of inheritance. We have also defined a separate $\boldsymbol{d_P}$ for each of the 4 paths, instead of just for each sex selected, although this is only necessary if the population is structured.

It is necessary to keep the 4 paths as 4 separate equations, but the path specific $\boldsymbol{d_G}$'s can be summed/averages as indicated in equations 24 and 25 of the previous section.

The *gresponse()* function always uses this combined approach, simply setting to zero the $\boldsymbol{G}$ matrix components not present in any particular case, and always doing the 4 paths separately, even when there is no sexlinkage. Regardless

of how $\boldsymbol{d_P}$ is specified, it is always set up internally as 4 path specific pheno-
typic selection differentials, for use in the above equations, even if they are all
identical. That is the price of generality.

6.4.6   Limitations

Function *gresponse()* is experimental. It currently only does elementary predic-
tions of response to mass selection based on individual phenotype. It does a one
generation prediction assuming generations are not overlapping.

Sex specific genetic parameters and questions of population structure where
the male-to-male, male-to-female, female-to-male, and female-to-female path-
ways are considered separately, are not considered, althought there is provision
for path specific selection differentials.

Nonadditive genetic variation is not considered.

7 Some issues unique to *dmm()*.

Apart from using a dyadic model, *dmm()* has a number of facilities which might be termed experimental or novel. We want these to be both mathematically sound and useful. Because these have had limited scrutiny, they are grouped in this section under a general warning of let the user beware.

7.1 Cohorts

A cohort is a group of individuals reared together in a common environment, most likely up to the point of observation, or at least for a substantial period. It must be an environmental grouping. The following discussion will be wrong if it represents a genetic grouping. An example would be a group of animals born at the same time and reared together such as a drop of lambs, or a group of plants grown in the same plot.

So why give special attention to cohort? Is it not just another fixed effect? The answer depends on our intended use of the genetic parameter estimates. If we want parameter estimates that are relevant to selection within a cohort, then leaving cohort as a fixed effect in the model is the appropriate course of action. Genetic parameters will then be a summary of within cohort variation and will be appropriate for selection among individuals within a cohort.

If we want parameter estimates that are relevant to selection across cohorts, or ignoring cohorts, then cohort must be a random effect in the model, and its variance must be included in summing components to obtain phenotypic variance. Genetic parametrs will then include between cohort variation as an environmental variance "VarE(C)", and will be appropriate for selection of individuals across cohorts.

To setup the latter case in *dmm()* we need to use the argument *cohortform* to define a cohort formula, and we also need to include "VarE(C)" in the *component* argument. Any dataframe columns which appear in *cohortform* should not also appear in the *fixform* argument.

An example will make this clear

```
> library(dmm)
> data(sheep.df)
> sheep.mdf <- mdf(sheep.df,pedcols=c(1:3),factorcols=c(4:6),ycols=c(7:9),
```

In the *sheep.df* dataset there is a factor called *Year* which records the year of birth of each individual. We are going to let *Year* be the *cohort* grouping. We first run an analysis without a cohort effect

```
> sheep.fit <- dmm(sheep.mdf, Ymat ~ 1 + Year + Tb + Sex)
Dyadic mixed model fit for datafile: sheep.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 2
OLS-b step:
```

```
no of fixed effect df (k) =  9
no of traits (l) =  3
Setup antemodel matrices:
no of individuals in pedigree (m) =  44
no of individuals with data and X codes (n) =  36
Rank of X: 9   No of Fixed Effects: 9
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
>
```

Then we rerun with *Year* as a cohort instead of as a fixed effect

```
> sheep.fitc <- dmm(sheep.mdf, Ymat ~ 1 + Tb + Sex, cohortform = ~ Year,
                components=c("VarE(I)","VarE(C)","VarG(Ia)"))
Dyadic mixed model fit for datafile: sheep.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 3
OLS-b step:
no of fixed effect df (k) =  3
no of traits (l) =  3
Setup antemodel matrices:
ncohortcodes =  8
no of individuals in pedigree (m) =  44
no of individuals with data and X codes (n) =  36
Rank of X: 3   No of Fixed Effects: 3
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
>
```

Then let us compare the genetic parameter estimates from the two runs. We will look just at trait "Diam"

```
> gsummary(sheep.fit,traitset="Diam")
Call:
gsummary.dmm(dmmobj = sheep.fit, traitset = "Diam")

Proportion of phenotypic var/covariance partitioned by DME to each component (OLS-b):

         Trait Estimate StdErr  CI95lo CI95hi
VarE(I)   Diam    0.321  0.201 -0.0722  0.715
VarG(Ia)  Diam    0.679  0.207  0.2732  1.084
VarP(I)   Diam    1.000  0.000  1.0000  1.000
```

```
  .....

Phenotypic var/covariance from components partitioned by DME (OLS-b):

  Traitpair Estimate StdErr CI95lo CI95hi
1 Diam:Diam     1.01  0.132  0.748   1.27


>
>gsummary(sheep.fitc,traitset="Diam")
Call:
gsummary.dmm(dmmobj = sheep.fitc, traitset = "Diam")


Proportion of phenotypic var/covariance partitioned by DME to each component (OLS-b):

          Trait Estimate StdErr   CI95lo CI95hi
VarE(I)    Diam   0.0587 0.1240 -0.18428  0.302
VarG(Ia)   Diam   0.8542 0.1333  0.59289  1.116
VarE(C)    Diam   0.0871 0.0462 -0.00348  0.178
VarP(I)    Diam   1.0000 0.0000  1.00000  1.000


  .....


Phenotypic var/covariance from components partitioned by DME (OLS-b):

  Traitpair Estimate StdErr CI95lo CI95hi
1 Diam:Diam     1.63  0.148   1.34   1.92


>
```

The phenotypic variance increases, as expected when another variance is included in phenotypic variation. Also the estimates of 'Proportion' change when "VarE(C)" is added to the causal components, for two reasons, firstly the denominator of the 'Proportion' is different, and secondly "VarE(C)" is correlated with the other components so its presence in the model alters the fit.

In theory, changing from a fixed effect "Year" to a random effect "VarE(C)" should not alter the fit, but that only applies when all effects are fitted simultaneously. With a mixed model there is an heirarchy - fixed effects fitted first, data adjusted, then random effects estimated. That applies to *dmm* and to bias-corrected-ML and REML. The only procedure which fits all effects simultaneously in a mixed model is Henderson's Method 3.

7.2   Partitioning "VarE(M)"

If there is more than one offspring per dam, the component "VarE(M)", as fitted by *dmm()* may be a complex mixture of

$\boldsymbol{\sigma^2_{E_M}}$ maternal environmental effect

$\boldsymbol{\sigma^2_{E_C}}$ nonmaternal environmental effect due to littermates (the so-called "common environment" variance)

These can be separated only if dams have multiple litters and multiple offspring per litter. In that case, we can use the *cohort* facility in *dmm()*, where *cohort* is any grouping which separates multiple litters of a dam (eg Year of birth), to define the following components

VarE(C) variance between cohorts

VarE(M&C) variance of individuals with the same dam and same cohort - ie littermates

VarE(M&!C) variance of individuals with the same dam and not the same cohort - ie maternal half sibs across cohorts

With this partitioning we can clearly obtain $\boldsymbol{\sigma^2_{E_M}}$ annd $\boldsymbol{\sigma^2_{E_C}}$ if desired, by simple subtraction

$$\begin{aligned}
\boldsymbol{\sigma^2_{E_C}} &= VarE(M\&C) - VarE(M\&!C) \\
\boldsymbol{\sigma^2_{E_M}} &= VarE(M\&!C)
\end{aligned}$$

The above discussion is well presented by Bijma(2006)[3].

We do not of course fit VarE(M) at the same time as its two components VarE(M&C) and VarE(M&!C). The relationship between VarE(M) and its components is actually a weighted average

$$VarE(M) = \frac{n_1 VarE(M\&C) + n_2 VarE(M\&!C)}{n_1 + n_2}$$

where $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$ are sums of the relavant columns of the $\boldsymbol{W}$ matrix.

We continue the example of the cohort section, fitting maternal environmental variance and then splitting it as follows

```
> ....
>sheep.fitm <- dmm(sheep.mdf,Ymat ~ 1 + Tb + Sex,
          components=c("VarE(I)","VarE(C)", "VarG(Ia)", "VarE(M)", "VarG(Ma)"),
          cohortform = ~ Year)
Dyadic mixed model fit for datafile: sheep.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 5
OLS-b step:
no of fixed effect df (k) =  3
no of traits (l) =  3
```

```
Setup antemodel matrices:
ncohortcodes =  8
no of individuals in pedigree (m) =  44
no of individuals with data and X codes (n) =  36
Rank of X: 3   No of Fixed Effects: 3
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
>
> summary(sheep.fitm,traitset=c("Cww"))
Call:
summary.dmm(dmmobj = sheep.fitm, traitset = c("Cww"))

Coefficients fitted by OLS for fixed effects:

            Trait Estimate StdErr  CI95lo CI95hi
(Intercept)  Cww     4.593  0.127  4.3452  4.841
TbT          Cww    -0.104  0.173 -0.4420  0.234
SexM         Cww     0.426  0.183  0.0675  0.784


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

          Traitpair Estimate StdErr   CI95lo CI95hi
VarE(I)     Cww:Cww   0.0645 0.0637 -0.06032 0.1894
VarG(Ia)    Cww:Cww   0.1215 0.0612  0.00165 0.2414
VarE(M)     Cww:Cww   0.2228 0.1239 -0.02004 0.4656
VarE(C)     Cww:Cww   0.0447 0.0204  0.00461 0.0848
VarG(Ma)    Cww:Cww   0.0744 0.1217 -0.16407 0.3129
VarP(I)     Cww:Cww   0.5279 0.0429  0.44381 0.6121


>
>sheep.fitm2 <- dmm(sheep.mdf,Ymat ~ 1 + Tb + Sex,
 components=c("VarE(I)","VarE(C)", "VarG(Ia)","VarE(M&C)","VarE(M&!C)","VarG(Ma)"),
 cohortform = ~ Year)
Dyadic mixed model fit for datafile: sheep.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 6
OLS-b step:
no of fixed effect df (k) =  3
no of traits (l) =  3
Setup antemodel matrices:
ncohortcodes =  8
no of individuals in pedigree (m) =  44
```

```
no of individuals with data and X codes (n) =  36
Rank of X: 3   No of Fixed Effects: 3
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
>
> summary(sheep.fitm2,traitset=c("Cww"))
Call:
summary.dmm(dmmobj = sheep.fitm2, traitset = c("Cww"))

Coefficients fitted by OLS for fixed effects:

            Trait Estimate StdErr  CI95lo CI95hi
(Intercept)  Cww     4.593  0.127  4.3452  4.841
TbT          Cww    -0.104  0.173 -0.4420  0.234
SexM         Cww     0.426  0.183  0.0675  0.784


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

           Traitpair Estimate StdErr    CI95lo CI95hi
VarE(I)      Cww:Cww   0.0939 0.0737 -0.05061 0.2384
VarG(Ia)     Cww:Cww   0.1056 0.0624 -0.01666 0.2280
VarE(C)      Cww:Cww   0.0484 0.0209  0.00738 0.0894
VarE(M&!C)   Cww:Cww   0.2613 0.1258  0.01475 0.5078
VarE(M&C)    Cww:Cww   0.1461 0.1328 -0.11410 0.4063
VarG(Ma)     Cww:Cww   0.0692 0.1218 -0.16958 0.3080
VarP(I)      Cww:Cww   0.7246 0.1349  0.46027 0.9889


>
```

so in this small demonstration case VarE(M&!C) turns out larger than
VarE(M) so the estimate of $\sigma^2_{E_C}$ by subtraction will be negative. This often
happens if we overfit a small dataset with many components.

7.3 Using robust regression to solve DME's

One of the advantages of turning variance component estimation into a regres-
sion is that one can use regression techniques other than least squares. One
option is robust regression. There are several packages for robust regression
available in R. All of them are only capable of multiple regression with a uni-
variate response. In *dmm()* the function *lmrob()* from package *robustbase* can
be used instead of QR to solve the dyadic model equations, if the argument
dmeopt="lmrob" is specified.

There is no information on the properties of the variance component esti-

mates obtained by *dmm()* using *lmrob()*. Their standard errors are obtained as the standard errors of coefficients returned by *lmrob()*.

Experience has shown that "lmrob" estimates do not exactly agree with QR estimates for a balanced dataset. Apparently robust algorithms will always remove some perceived outlier data points, and this will naturally lead to slightly different estimates of regression coefficients. They also usually lead to smaller standard errors.

One possible use for the robust regression option is to check whether the usual regression estimates obtained by ordinary least squares (*qr* or *lm* options) have been unduly affected by outliers. If the robust regression estimates of variance components are vastly different from the OLS estimates, then some serious outliers are indicated. In this case the robust regression estimates are probably superior. The best course of action would be to find the reason for the outliers and remove or correct them. The only limitation to using *lmrob* option in this way is that robust regression is a univariate procedure, so you can only assess one trait at a time.

7.4   Using principal component regression to solve DME's

If there are collinearities in the dyadic model equations (check the column correlations) then one of the basic assumptions of multiple regression is not met - the independent variables (in this case the columns of the $\boldsymbol{W}$ matrix) are not independent. In practice any correlation exceeding about 0.5 is thought to be a serious violation.

This is by no means an unusual situation in quantitative genetic data, particularly if non-additive genetic components are being estimated. For example, the following correlations are obtained for our test data sets

```
> quercus.fit$dme.correl
            VarE(I)   VarG(Ia)  VarG(Id)
VarE(I)   1.0000000 0.7314919 0.9428127
VarG(Ia) 0.7314919 1.0000000 0.8610611
VarG(Id) 0.9428127 0.8610611 1.0000000
> warcolak.fitg$dme.correl
            VarE(I)   VarG(Ia)  VarG(Id) VarGs(Ia)
VarE(I)    1.0000000 0.4856324 0.9190639 0.7494688
VarG(Ia)   0.4856324 1.0000000 0.6255619 0.7473142
VarG(Id)   0.9190639 0.6255619 1.0000000 0.8105875
VarGs(Ia)  0.7494688 0.7473142 0.8105875 1.0000000
> tstmo1.fit1$dme.correl
            VarE(I)   VarG(Ia)
VarE(I)   1.0000000 0.4158424
VarG(Ia) 0.4158424 1.0000000
> harv.fit$dme.correl
            VarE(I)   VarG(Ia)
VarE(I)   1.0000000 0.8780947
```

```
VarG(Ia) 0.8780947 1.0000000
>
```

Note the correlations exceeding **0.9** involving dominance variance.

There is a way of circumventing this issue, which may be helpful in the current context. The *dmm()* argument *dmeopt="pcr"* invokes a principal component regression in place of QR, using the *method="svdpc"* in the function *mvr()* of package *pls*. What this does is to transform the variance components to a set of independent variables (which are linear combinations of the variance components) before doing the multiple regression, then transform the fitted coefficients back to the original variables. This avoids the assumption violation, but it requires a certain amount of user intervention. One needs to choose how many transformed independent variates to use, and the criteria for this are not readily automated.

With this in mind, the *dmeopt="pcr"* option is setup with some extra screen output to aid in user choices. Usually several runs will be required to arrive at a satisfactory number of components.

We will use the *quercus.df* dataset for an example. Three variance components are fitted, with correlations noted above. Component "VarG(Id)" is strongly correlated with the other two components. First a run allowing variable *ncomp*, which is the number of principal components to be included in the regression, to be its default value which is the rank of the W matrix - in this case **3**.

```
> data(quercus.df)
> quercus.mdf <- mdf(quercus.df,pedcols=c(1:3), factorcols=4, ycols=c(5:6),
                 sexcode=c(1,2), relmat=c("E","A","D"))
 .....
> quercus.fitpcr <- dmm(quercus.mdf,Ymat ~ 1,
     components = c("VarE(I)","VarG(Ia)", "VarG(Id)"),
     dmeopt="pcr",relmat = "withdf")
Dyadic mixed model fit for datafile: quercus.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 3
OLS-b step:
no of fixed effect df (k) =  1
no of traits (l) =  2
Setup antemodel matrices:
no of individuals in pedigree (m) =  260
no of individuals with data and X codes (n) =  180
Rank of X: 1   No of Fixed Effects: 1
DME substep:
PCR option on dyadic model equations:
Data:  X dimension: 32400 3
Y dimension: 32400 4
```

```
Fit method: svdpc
Number of components considered: 3

VALIDATION: RMSEP
Cross-validated using 10 random segments.

Response: Trait1:Trait1
        (Intercept)  1 comps  2 comps  3 comps
CV            1.158    1.154    1.154    1.155
adjCV         1.158    1.154    1.154    1.154

Response: Trait1:Trait2
        (Intercept)  1 comps  2 comps  3 comps
CV            1.449    1.449    1.449    1.449
adjCV         1.449    1.449    1.449    1.449

Response: Trait2:Trait1
        (Intercept)  1 comps  2 comps  3 comps
CV            1.449    1.449    1.449    1.449
adjCV         1.449    1.449    1.449    1.449

Response: Trait2:Trait2
        (Intercept)  1 comps  2 comps  3 comps
CV            1.813    1.808    1.808    1.808
adjCV         1.813    1.808    1.808    1.808

TRAINING: % variance explained
                 1 comps     2 comps     3 comps
X              89.286032   99.157279   1.000e+02
Trait1:Trait1   0.643912    0.648228   6.544e-01
Trait1:Trait2   0.008194    0.008364   8.699e-03
Trait2:Trait1   0.008194    0.008364   8.699e-03
Trait2:Trait2   0.652555    0.652902   6.557e-01
DME substep completed:
OLS-b step completed:
>
> quercus.fitpcr$dme.corre
           VarE(I)   VarG(Ia)  VarG(Id)
VarE(I)  1.0000000 0.7314919 0.9428127
VarG(Ia) 0.7314919 1.0000000 0.8610611
VarG(Id) 0.9428127 0.8610611 1.0000000
>
> summary(quercus.fitpcr)
Call:
summary.dmm(dmmobj = quercus.fitpcr)
```

```
Coefficients fitted by OLS for fixed effects:

   Trait Estimate StdErr  CI95lo CI95hi
1 Trait1   0.0812 0.0804 -0.0764  0.239

   Trait Estimate StdErr CI95lo CI95hi
1 Trait2  -0.0855  0.101 -0.283  0.112



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

            Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait1:Trait1   0.0348 0.2940 -0.5414  0.611
VarG(Ia) Trait1:Trait1   0.1966 0.1430 -0.0836  0.477
VarG(Id) Trait1:Trait1   0.9386 0.3727  0.2082  1.669
VarP(I)  Trait1:Trait1   1.1700 0.0863  1.0008  1.339

            Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait1:Trait2    0.135  0.369 -0.5887  0.858
VarG(Ia) Trait1:Trait2    0.134  0.179 -0.2177  0.486
VarG(Id) Trait1:Trait2   -0.113  0.468 -1.0301  0.804
VarP(I)  Trait1:Trait2    0.156  0.108 -0.0568  0.368

            Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait2:Trait1    0.135  0.369 -0.5887  0.858
VarG(Ia) Trait2:Trait1    0.134  0.179 -0.2177  0.486
VarG(Id) Trait2:Trait1   -0.113  0.468 -1.0301  0.804
VarP(I)  Trait2:Trait1    0.156  0.108 -0.0568  0.368

            Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait2:Trait2    0.975  0.460  0.0726   1.88
VarG(Ia) Trait2:Trait2    0.813  0.224  0.3745   1.25
VarG(Id) Trait2:Trait2    0.047  0.584 -1.0967   1.19
VarP(I)  Trait2:Trait2    1.835  0.135  1.5701   2.10

>
end{verbatim}
```

First let us note that "pcr" is using all 3 components

```
\begin{verbatim}
Fit method: svdpc
Number of components considered: 3
```

and that it actually does analyses with 1,2, and 3 components, and the
successive amounts of variance explained are

```
TRAINING: % variance explained
                 1 comps     2 comps     3 comps
X               89.286032   99.157279   1.000e+02
```

Now let us compare the results with those from the same model using
dmeopt="qr"

```
> summary(quercus.fit)
Call:
summary.dmm(dmmobj = quercus.fit)

Coefficients fitted by OLS for fixed effects:

   Trait Estimate StdErr  CI95lo CI95hi
1 Trait1   0.0812 0.0804 -0.0764  0.239

   Trait Estimate StdErr CI95lo CI95hi
1 Trait2  -0.0855  0.101 -0.283  0.112


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

            Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait1:Trait1   0.0348 0.2940 -0.5414  0.611
VarG(Ia) Trait1:Trait1   0.1966 0.1430 -0.0836  0.477
VarG(Id) Trait1:Trait1   0.9386 0.3727  0.2082  1.669
VarP(I)  Trait1:Trait1   1.1700 0.0863  1.0008  1.339


            Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait1:Trait2   0.135  0.369 -0.5887  0.858
VarG(Ia) Trait1:Trait2   0.134  0.179 -0.2177  0.486
VarG(Id) Trait1:Trait2  -0.113  0.468 -1.0301  0.804
VarP(I)  Trait1:Trait2   0.156  0.108 -0.0568  0.368


            Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait2:Trait1   0.135  0.369 -0.5887  0.858
VarG(Ia) Trait2:Trait1   0.134  0.179 -0.2177  0.486
VarG(Id) Trait2:Trait1  -0.113  0.468 -1.0301  0.804
VarP(I)  Trait2:Trait1   0.156  0.108 -0.0568  0.368


            Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait2:Trait2   0.975  0.460  0.0726   1.88
VarG(Ia) Trait2:Trait2   0.813  0.224  0.3745   1.25
VarG(Id) Trait2:Trait2   0.047  0.584 -1.0967   1.19
VarP(I)  Trait2:Trait2   1.835  0.135  1.5701   2.10

>
```

So the variance component estimates are identical ("pcr" with ncomp=3 versus "qr"), and the standard errors are also identical. This is surprising because "pcr" uses a jackknife technique to estimate the var/covariance matrix of estimates, while "qr" uses the exact procedure for regression coefficients estimated by least squares.

Now a rerun with ncomp=2, that is we will only use the first two principal components to fit the regressions, but will transform the result back to the 3 variance components

```
> quercus.fitpcr2 <- dmm(quercus.mdf,Ymat ~ 1,
        components = c("VarE(I)","VarG(Ia)", "VarG(Id)"),
        dmeopt="pcr",relmat = "withdf",ncomp.pcr=2)
Dyadic mixed model fit for datafile: quercus.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 3
OLS-b step:
no of fixed effect df (k) =  1
no of traits (l) =  2
Setup antemodel matrices:
no of individuals in pedigree (m) =   260
no of individuals with data and X codes (n) =   180
Rank of X: 1   No of Fixed Effects: 1
DME substep:
PCR option on dyadic model equations:
Data:  X dimension: 32400 3
Y dimension: 32400 4
Fit method: svdpc
Number of components considered: 2

VALIDATION: RMSEP
Cross-validated using 10 random segments.

Response: Trait1:Trait1
        (Intercept)  1 comps  2 comps
CV            1.158    1.154    1.155
adjCV         1.158    1.154    1.155

Response: Trait1:Trait2
        (Intercept)  1 comps  2 comps
CV            1.449    1.449    1.449
adjCV         1.449    1.449    1.449

Response: Trait2:Trait1
        (Intercept)  1 comps  2 comps
CV            1.449    1.449    1.449
```

```
adjCV          1.449    1.449    1.449

Response: Trait2:Trait2
      (Intercept)  1 comps  2 comps
CV          1.813    1.808    1.808
adjCV       1.813    1.808    1.808

TRAINING: % variance explained
                 1 comps    2 comps
X              89.286032  99.157279
Trait1:Trait1   0.643912   0.648228
Trait1:Trait2   0.008194   0.008364
Trait2:Trait1   0.008194   0.008364
Trait2:Trait2   0.652555   0.652902
DME substep completed:
OLS-b step completed:
>
> summary(quercus.fitpcr2)
Call:
summary.dmm(dmmobj = quercus.fitpcr2)


Coefficients fitted by OLS for fixed effects:

   Trait Estimate StdErr  CI95lo CI95hi
1 Trait1   0.0812 0.0804 -0.0764  0.239

   Trait Estimate StdErr CI95lo CI95hi
1 Trait2  -0.0855  0.101 -0.283  0.112



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

            Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)  Trait1:Trait1    0.430 0.1332 0.1685  0.691
VarG(Ia) Trait1:Trait1    0.338 0.1356 0.0724  0.604
VarG(Id) Trait1:Trait1    0.416 0.0674 0.2838  0.548
VarP(I)  Trait1:Trait1    1.184 0.0916 1.0041  1.363

            Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait1:Trait2   0.0193  0.102 -0.1800  0.218
VarG(Ia) Trait1:Trait2   0.0927  0.148 -0.1980  0.383
VarG(Id) Trait1:Trait2   0.0396  0.049 -0.0564  0.136
VarP(I)  Trait1:Trait2   0.1516  0.107 -0.0589  0.362

            Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait2:Trait1   0.0193 0.1368 -0.2489  0.287
```

```
VarG(Ia) Trait2:Trait1    0.0927 0.1282 -0.1585   0.344
VarG(Id) Trait2:Trait1    0.0396 0.0725 -0.1025   0.182
VarP(I)  Trait2:Trait1    0.1516 0.1063 -0.0568   0.360


             Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)  Trait2:Trait2     0.557  0.123  0.317  0.797
VarG(Ia) Trait2:Trait2     0.663  0.167  0.336  0.991
VarG(Id) Trait2:Trait2     0.600  0.067  0.469  0.731
VarP(I)  Trait2:Trait2     1.821  0.158  1.510  2.131

>
```

The resulting estimates are remarkably different and have much smaller standard errors. This is exactly what the textbooks say - regression coefficients from principal component regression may be biased, but will have a smaller mean square error, compared to least squares. One has to hope that the bias is small and the gain from reduced standard errors more than compensates.

What we have done, by excluding the third principal component, is to restrict the 3D space in which the three component estimates can vary. This is equivalent to applying a constraint equation to the three estimates. The constraint equation is a linear function of the form

$$a_1 VarE(I) + a_2 VarG(Ia) + a_3 VarG(Id) = 0$$

and we can get the a_i coefficients of this function from the third column of the right singular vector of the W matrix. The *pls* package calls the right singular vector "loadings" and provides a function to extract these from the fit object. The function *dmm()* returns the "loadings" in an object called *pcr.loadings* which is part of the returned object of class *dmm* whenever argument *dmeopt="pcr"* is used. So we can inspect the loadings as follows

```
> quercus.fitpcr$pcr.loadings)

Loadings:
          Comp 1 Comp 2 Comp 3
'VarE(I)'   0.485  0.646 -0.589
'VarG(Ia)'  0.680 -0.702 -0.211
'VarG(Id)'  0.550  0.298  0.780


              Comp 1 Comp 2 Comp 3
SS loadings    1.000  1.000  1.000
Proportion Var 0.333  0.333  0.333
Cumulative Var 0.333  0.667  1.000
>
```

The constraint coefficients we want in this case are given by the "Comp 3" column of the loadings matrix. Note that it is necessary to do the fit with all

3 principal components to get at the loadings. If you use **ncomp = 2** the required column "Comp 3" of loadings will be missing. So we have

$$-0.589 \times VarE(I) - 0.211 \times VarG(Ia) + 0.780 \times VarG(Id) = 0$$

so substituting the **ncomp = 2** estimates for trait "Trait1" we get

$$0.589 \times 0.430 - 0.211 \times 0.338 + 0.780 \times 0.416 = -0.000108$$

The estimates from a 2 principal component fit do indeed conform to the constraint.

The above constraint equation is actually the equation of a plane in the 3D space of the three variance component estimates. When we omit the third principal component, estimates of the three variance components are constrained to lie on this plane. When we include the third principal component, estimates of the three variance components can be anywhere in the 3d space.

If we were to omit more than one principal component (not feasible in the current example), there would be more than one constraint, given by the appropriate columns of the loadings matrix. Estimates would then be constrained to lie onlines of intersection of two or more hyperplanes.

The constraint(s) on the variance component estimates are the reason that principal component regression leads to biased estimates.

We might conclude by noting that there are three ways of dealing with colliniarities in the dyadic model equations

- use "qr" and ignore the issue, accept the higher standard errors and enjoy the unbiased guarantee. The results are only unbiased if the model is correct - ie if you have not omitted variance components which are nonzero.

- use "qr" and change the model omitting one or more variance components. This amounts to setting a constraint that the omitted component(s) are zero. So the results are biased unless the omitted component is actually zero.

- use "pcr" and omit one or more principal components. Enjoy the lower standard errors. This amounts to setting a constraint that some linear combination of the components is zero, or it can be rewritten as a constraint that one particular component is a linear combination of the others. If you are happy with the implied constraint, then you are happy with the bias. If the implied constraint is actually true - ie the true values of the components do actually lie on the constraint plane, then there is no bias.

For completeness, we should look at the results obtained with the second option above, that is omitting "VarG(Id)" which implies constraining it to zero

```
> summary(quercus.fit2)
Call:
summary.dmm(dmmobj = quercus.fit2)
```

```
Coefficients fitted by OLS for fixed effects:

   Trait Estimate StdErr  CI95lo CI95hi
1 Trait1    0.0812 0.0804 -0.0764  0.239

   Trait Estimate StdErr CI95lo CI95hi
1 Trait2  -0.0855  0.101 -0.283  0.112


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

           Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)  Trait1:Trait1    0.703 0.1266  0.455  0.951
VarG(Ia) Trait1:Trait1    0.468 0.0939  0.284  0.652
VarP(I)  Trait1:Trait1    1.171 0.0863  1.002  1.340

           Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait1:Trait2   0.0541  0.159 -0.2573  0.365
VarG(Ia) Trait1:Trait2   0.1013  0.118 -0.1297  0.332
VarP(I)  Trait1:Trait2   0.1554  0.108 -0.0569  0.368

           Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)  Trait2:Trait1   0.0541  0.159 -0.2573  0.365
VarG(Ia) Trait2:Trait1   0.1013  0.118 -0.1297  0.332
VarP(I)  Trait2:Trait1   0.1554  0.108 -0.0569  0.368

           Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)  Trait2:Trait2    1.008  0.198  0.620   1.40
VarG(Ia) Trait2:Trait2    0.827  0.147  0.539   1.11
VarP(I)  Trait2:Trait2    1.835  0.135  1.570   2.10

>
```

We get different estimates for the remaining two components and standard errors which are smaller for Trait1 but larger for Trait2, compared with the "pcr" with ncomp=2 estimates. It is hard to choose between these two results.

As we said at the outset, this facility is experimental.

There is an excellent presentation of principal component regression and the *pls* package in Mevik and Wehrens(2007) [18].

7.5 Constraining covariance matrices to be positive definite

If the option *posdef=T* is used each matrix of cross-trait (co)variances for each "Varxxx" component will be individually positive definite, and each cross-effect

covariance (if "Covxxx" components are defined) will be constrained such that the corresponding correlation is in the bounds -1 to 1. If option *gls=T* is used *posdef=T* is enforced.

The algorithm used takes the matrix of cross-trait covariances for each variance component at a time, first does an approximate procedure which ensures that its eigenvalues are positive, then calls the *nearPD()* function from package *Matrix*.This 2-step procedure ensures that *nearPD* does not fail when matrices from small test examples are wildly negative definite. When all variance components have been made positive definite, the algorithm looks at each covariance component in turn, and ensures that for each element of each covariance component matrix the corresponding correlation is not outside the bounds -1 to 1. If it is outside bounds, the covariance component is altered. The two variances involved in the correlation are left unchanged.

Whenever a constraint is applied, in any estimation procedure, the estimated parameters are biased and their standard errors are only approximate. If one wants the positive definite constraint one simply has to hope that the bias is not large.

In a multi-trait analysis with the positive definite constraint applied, if one trait has variance component estimates which are negative before the constraint is applied, the constraint algorithm will adjust the estimates for all traits. Then if the "rogue" trait is omitted and the analysis rerun, the estimates for all remaining traits will not be adjusted, and will therefore differ from the previous analysis. We simply need to be aware that in a multivariate analysis, the presence or absence of one trait may affect the results for all other traits.

7.6 Dealing with unequal numbers of observations for various traits

The number of observations availabble for each pair of traits in a dataset can be obtained with the *make.countarray()* function. See its *help()* page for details.

If there are missing observations for some traits in a multi-trait dataset, *dmm()* provides two special options which allow a comprehensive analysis of all the available data for each trait. These are specified as the arguments

traitspairwise=T Forces a separate *dmm()* run for each pair of traits and returns an object of class *dmmarray*, which is an array of class *dmm* objects, one for each pair of traits.

traitsblockwise=T,... Forces a separate *dmm()* run for each pair of blocks of traits. Blocks are specified in the ellipsis argument. Returns an object of class *dmmblockarray*, which is an array of class *dmm* objects, one for each pair of blocks of traits. Use this option if there are blocks of traits with equal replication, and other blocks with equal but different replication.

When using either of the above options, the fixed effect model must be appropriate for all pairs (or blocks) of traits, and the same variance components

will be estimated for all pairs (blocks). The number of levels of each fixed effect can vary between pairs (blocks) but at least one level must be present.

After running *dmm()* with either of the above options one would typically want to construct a single genetic variance/covariance matrix for all the traits for use with the *gresponse()* function. This can be readily accomplished with the *condense.dmmarray* and *condense.dmmblockarray()* functions which are described in their *help()* pages. These functions return an object of class *dmm* which can be passed to the *gresponse()* function or to any of the *print* , *gprint* or *plot* functions.

The statistical properties and positive definite status of matrices obtained by combining trait pair (or block pair) estimates with differing replication are not known. It is simply a matter of doing the best one can with available data to estimate each element of the variance/covariance matrices, then hoping that combining all the elements does not lead to matrices with undesirable properties.

7.7   Class-specific component estimates and genetic parameters

From version 2.1-1 of *dmm()* or later component estimates can be made specific to the classes of one or more specific factors. This feature is documented separately in dmmClassSpecific.pdf [13]

7.8   Maternal and paternal founderline groups

A founderline is a set of individuals all descended from one individual in the base generation. A maternal founderline is a set of individuals all descended from one female in the base generation. A paternal founderline is a set of individuals all descended from one male in the base generation.

We are interested in maternal founderline effects because their variation reflects variations which are inherited only from the female parent. This includes various forms of cytoplasmic inheritance, including mitochondrial DNA in animals, and plasmid DNA in plants. The contribution of these effects to phenotypic variance can be analysed by fitting a variance component for variance between maternal founder lines. We call this component VarGlm(I).

We are interested in paternal founderline effects because their variation reflects variations which are inherited only from the male parent. This includes Y-chromosome inheritance in mammals. The contribution of this effect to phenotypic variance can be analysed by fitting a variance component for variance between paternal founder lines. We call this component VarGlp(I).

The function *founderLine()* which is part of the *nadiv* package will compute a vector of maternal or paternal founderline codes. To use these founderLine codes with *dmm()* they need to be appended to the dataframe as columns called "MLine" or "PLine".

Alternatively there is internal code in *dmm()* which will set up the required founderline codes, as it executes. If the supplied dataframe does not contain the columns "MLine" or "PLine" *dmm()* will use its own internal code.

There is a slight difference in the above two approaches. The *nadiv* routine *founderLine()* gives an 'NA' code to base individuals of sex opposite to that of the founderline - for example for a maternal founderline male base individuals are coded 'NA'. The internal code in *dmm()* gives a numeric code to all individuals. This only makes a difference to the analysis if there are base individuals with data.

Apart from this, all that is needed to setup a founderline component in *dmm()* is to include either "VarGlm(I)" or "VarGlp(I)" in the *component* argument.

An example will make this clear

```
> library(dmm)
> data(sheep.df)
> sheep.mdf <- mdf(sheep.df,pedcols=c(1:3),factorcols=c(4:6),ycols=c(7:9),
      sexcode=c("M","F"),relmat=c("E","A"),keep=T)
> sheep.linefit <-  dmm(sheep.mdf,Cww ~ 1 + Sex + Year, components=c("VarE(I)","VarG(Ia)","V
Dyadic mixed model fit for datafile: sheep.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
OLS-b step:
no of fixed effect df (k) =  9
no of traits (l) =  1
Setup antemodel matrices:
No of factors with specific components: 0
No of non-specific components partitioned: 3
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 maternal line codes =   16
no of individuals in pedigree (m) =   44
no of individuals with data and X codes (n) =   37
Rank of X: 9   No of Fixed Effects: 9
DME substep:
No of components defined =  3
No of components estimable =  3
Checking dyadic model equations:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
>
> summary(sheep.linefit)
Call:
summary.dmm(object = sheep.linefit)


Coefficients fitted by OLS for fixed effects:
```

```
              Trait Estimate StdErr    CI95lo CI95hi
(Intercept)    Cww    4.1000  0.267   3.57615  4.624
SexM           Cww    0.2237  0.178  -0.12614  0.574
Year1982       Cww    0.7667  0.378   0.02583  1.508
Year1983       Cww    0.0441  0.356  -0.65442  0.743
Year1984       Cww    0.3881  0.339  -0.27687  1.053
Year1985       Cww    0.6361  0.323   0.00203  1.270
Year1986       Cww    0.9470  0.328   0.30315  1.591
Year1987       Cww    0.4588  0.333  -0.19334  1.111
Year1988       Cww   -0.2237  0.564  -1.32829  0.881


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

          Traitpair Estimate StdErr   CI95lo CI95hi
VarE(I)     Cww:Cww   0.0237 0.0558  -0.0857 0.1331
VarG(Ia)    Cww:Cww   0.1910 0.0785   0.0372 0.3449
VarGlm(I)   Cww:Cww   0.0185 0.0346  -0.0494 0.0863


> gsummary(sheep.linefit)
Call:
gsummary.dmm(dmmobj = sheep.linefit)

Proportion of phenotypic var/covariance partitioned by DME:
 to each component (OLS-b):

          Trait Estimate StdErr CI95lo CI95hi
VarE(I)     Cww   0.1016  0.213 -0.316  0.520
VarG(Ia)    Cww   0.8191  0.331  0.170  1.468
VarGlm(I)   Cww   0.0792  0.157 -0.229  0.388
VarP(I)     Cww   1.0000  0.000  1.000  1.000

.......

>
```

So the analysis found 16 founderlines in this small test dataset, and fitted three random effect variance components, the last of which is "VarGlm(I)". This component will convert to a proportion of phenotypic variance with *gsummary()* the same as all other components.

The founderline components "VarGlm(I)" and "VarGlp(I)" can be made class specific, in the same way as all other components. However we do not define any covariances between these founderline components and any other components. They are assumed to be independent of all other effects.

8 Computing strategy

The *dmm* package is written entirely in R. It does not directly use any of the available sparse matrix or parallel processing facilities. The R code is blatantly straightforward. All of the coding effort to date has been focused on ensuring correctness of the calculations. There may be room for some optimization later on.

It has been known for many years that solving the dyadic model equations leads to estimates of variance components (Pukelsheim(1976) [24]). No-one seems to have attempted to actually use it as an estimation method. Twenty years ago, it would not have been computationally feasible. Part of my goal in producing the *dmm* package is to show that it is a feasible approach with today's computers. In that it must be said the I have been only partly successful as currently implimented in R - it is feasible with datasets up to about 10000 individuals.

Most of the available variance component estimation software is compute intensive, but will run in modest amounts of memory. *dmm()* is different in that it is computationally quite efficient, but requires massive amounts of memory. An example will make this clear. The *warcolak* dataset has 5000 individuals and two traits. On my PC which has an Intel ® Core ™ i7 processor, *dmm()* runs on the *warcolak* data in about 15 mins and requires 22Gb of memory for the OLS-b step. The GLS-b step requires an additional 144 minutes and 40Gb of memory.

The efficiency of the QR algorithm is quite astounding. A dataset of 5000 individuals generates dyadic model equations consisting of 25 million simultaneous equations. QR can solve these in less than 5 minutes in a PC, without the slightest hint of loss of significant digits. Most of the compute time in *dmm()* is spent setting up the equations, not in solving them. There would seem to be no argument for turning the DME's into normal equations - this may reduce the number of equations, but it is usually recommended against on numerical grounds. The other DME solving options ("robust" or "pcr") are less efficient.

The critical memory limitation is in R, not in available physical RAM. R has an inbuilt limit of $\mathbf{2^{31} - 1 = 2147483648 - 1}$ or approximately $\mathbf{2 \times 10^9}$, for the size of any array. This limit applies even in a 64-bit installation of R. In other words 64-bit R has not yet been configured to fully exploit 64-bit addressing. I understand that this limitation is being worked on, but is unlikely to change in the near future. The only option to circumvent it is to rewrite in another language.

The memory requirements of any example are readily calculated. The arrays which most likely to encounter the above limit are $\boldsymbol{\Psi}$ and $\boldsymbol{W}$ in equation 13. Their sizes are

$\boldsymbol{W} \ \ \boldsymbol{n^2 \times c}$

$\boldsymbol{\Psi} \ \ \boldsymbol{n^2 \times l^2}$

where n is the number of individuals, c is the number of components fitted, and l is the numnber of traits analysed.

For example, in the case of the *warcolak* dataset, $n = 5000$, $c = 4$ and $l = 2$, so the size of both W and Ψ is 10^8, which is within the limit by one order of magnitude.

There may be a way to reduce the memory requirements of array W, because it is sparse, but array Ψ is fully populated.

There are, of coarse, other arrays in the R code, and the total amount of physical RAM used is likely to be 10 to 20 times the sizes of the above arrays, but that is less likely to be a real limitation than is the internal R limit.

When using the *dmeopt="pcr"* method for solving the DME's, the memory requirements are more demanding. The *pls* package uses three dimensional arrays of size $n^2 \times l^2 \times ncomp$ because it solves the DME's for all values of the number of components retained from 1 up to $ncomp$. This is a severe limitation and some effort will be made to circumvent it in future versions of *dmm*.

If the *gls=T* option is used there is an additional memory requirement which is more restrictive. The product $n^2 \times l^2 \times c$ must not exceed $2^{31} - 1$. This is very restrictive indeed. For example with the dataset *merino.df* we have $n = 2599$, $l = 11$, and $c = 2upto8$. Here we have $2599^2 \times 11^2 = 8 \times 10^8$ so there is only room for $c = 2$ unless the number of traits is reduced. There are several issues, including the above, suggesting a rethink of the *gls=T* part of the algorithm.

9 Analysis of a large multivariate dataset

The dataset *merino.df* is a set of real data from an Australian Merino sheep research flock with a multi-generation pedigree, eight fixed effects, and 11 traits related to wool production. It is used here as a display of the utility of *dmm()* for analysis of a multivariate dataset with many of the real world complications. It contains pedigree records for 4449 individuals, 1593 of which have missing data, leaving 2856 individuals with complete data.

9.1 Data preparation

We start with a preprocessing of the datafile using the *mdf()* function

```
>library(dmm)
>data(merino.df)
> str(merino.df)
'data.frame': 4449 obs. of  22 variables:
 $ Id     : Factor w/ 3831 levels "50-0001","50-0009",..: NA NA NA NA NA NA NA NA NA 1 ...
 $ SId    : Factor w/ 136 levels "47-1438","47-2093",..: NA NA NA NA NA NA NA NA NA NA ...
 $ DId    : Factor w/ 999 levels "42-3693","42-3725",..: 4 12 85 92 126 129 160 166 71 65 ...
 $ Sex    : Factor w/ 2 levels "M","F": NA NA NA NA NA NA NA NA 2 1 ...
 $ Yearbi : Factor w/ 18 levels "50","51","52",..: 1 1 1 1 1 1 1 1 1 1 1 ...
 $ YearSbi: Factor w/ 18 levels "47","48","50",..: NA NA NA NA NA NA NA NA NA NA ...
 $ YearDbi: Factor w/ 22 levels "42","43","44",..: 1 1 5 5 5 5 5 5 4 4 ...
 $ Mob    : Factor w/ 2 levels "1","2": NA NA NA NA NA NA NA NA NA NA ...
 $ Agem   : Factor w/ 2 levels "3","9": NA NA NA NA NA NA NA NA NA 1 ...
 $ Birwt  : num  NA NA NA NA NA NA NA NA NA 3.6 ...
 $ Weanwt : num  NA NA NA NA NA NA NA NA NA 22.7 ...
 $ Birls  : Factor w/ 2 levels "1","2": NA NA NA NA NA NA NA NA NA 1 1 ...
 $ Weanls : Factor w/ 2 levels "1","2": NA NA NA NA NA NA NA NA NA 1 ...
 $ Crimp  : num  NA NA NA NA NA NA NA NA NA 12.3 ...
 $ Densty : num  NA NA NA NA NA NA NA NA NA 37.8 ...
 $ Diamtr : num  NA NA NA NA NA NA NA NA NA 21 ...
 $ Yield  : num  NA NA NA NA NA NA NA NA NA 55.2 ...
 $ Bodywt : num  NA NA NA NA NA NA NA NA NA 39.3 ...
 $ Wrinkl : int  NA NA NA NA NA NA NA NA NA 4 ...
 $ Length : num  NA NA NA NA NA NA NA NA NA 8.7 ...
 $ Flcwt  : num  NA NA NA NA NA NA NA NA NA 4.49 ...
 $ Woolwt : num  NA NA NA NA NA NA NA NA NA 2.48 ...
>
> merino.mdf <- mdf(merino.df,pedcols=c(1:3),factorcols=c(4:9,12:13),
    ycols=c(10:11,14:22),sexcode=c("M","F"),
    relmat=c("E","A","AA","AD","DD","D","S.hopi"))
Loading required package: Matrix
Loading required package: lattice
Pedigree Id check:
```

```
No of rows with Id in original dataframe =  4449
No of sex codes not in sexcode[] so changed to NA =  642
No of rows with Sex == NA removed from dataframe =  642
No of rows with Id == NA removed from dataframe =  1
No of rows with duplicated Id removed from dataframe =  8
No of rows remaining after duplicates and NA's removed =  3798
No of SId's with no matching Id =  5
No of DId's with no matching Id =  211
Length of dataframe with base Id's added =  4014
Renumber pedigree Id's:
Add matrix of multivariate traits:
Setup pedigree for nadiv():
Make relationship matrices:
starting to make D....done
starting to make D....done
starting to make D....done
S-inverse made: Starting to make S....done
Return mdf as an object of class mdf:
 containing the dataframe as mdf$df:
 and the relationship matrices as mdf$rel:
>
> str(merino.mdf$df)
'data.frame': 4014 obs. of  12 variables:
 $ Id     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ SId    : int  NA NA NA NA NA NA NA NA NA NA ...
 $ DId    : int  NA NA NA NA NA NA NA NA NA NA ...
 $ Sex    : Factor w/ 2 levels "F","M": 2 2 2 2 2 1 1 1 1 1 ...
 $ Yearbi : Factor w/ 18 levels "50","51","52",..: NA NA NA NA NA NA NA NA NA NA ...
 $ YearSbi: Factor w/ 18 levels "47","48","50",..: NA NA NA NA NA NA NA NA NA NA ...
 $ YearDbi: Factor w/ 22 levels "42","43","44",..: NA NA NA NA NA NA NA NA NA NA ...
 $ Mob    : Factor w/ 2 levels "1","2": NA NA NA NA NA NA NA NA NA NA ...
 $ Agem   : Factor w/ 2 levels "3","9": NA NA NA NA NA NA NA NA NA NA ...
 $ Birls  : Factor w/ 2 levels "1","2": NA NA NA NA NA NA NA NA NA NA ...
 $ Weanls : Factor w/ 2 levels "1","2": NA NA NA NA NA NA NA NA NA NA ...
 $ Ymat   : num [1:4014, 1:11] NA NA NA NA NA NA NA NA NA NA ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr  "47-1438" "48-1430" "48-1846" "48-1149" ...
  .. ..$ : chr  "Birwt" "Weanwt" "Crimp" "Densty" ...
>
```

We can see that *mdf()* has renumbered the Id's, deleted some individua;s with missing codes, added som base sires and dams, and made up the traits into a matrix called 'Ymat' to facilitate multi-trait analyses. It has also made all types of relationship matrix and appended these to the *mdf* object. For the sexlinked genetic relationship matrix we have assumed that dosage compensation operates by inactivation of the paternal X chromosome in females ("hopi"

model). We could equally as well have assumed random inactivation of one of the female X chromosomes ("hori" model). The two models ("hopi" and "hori") lead to different sexlinked genetic relationship matrices but the same variance component estimates. It is unclear from the literature which model applies to sheep, but it is certain that sheep, along with all mammals, use some form of X chromosome inactivation in females, so for present purpose the choice between "hori" or "hopi" does not matter.

9.2 Fixed effects

In the Australian pastoral environment year of birth has a major effect on the growth and productivity of sheep, so Yearbi and YearDbi are important fixed effects. Males and females differ in wool production and are usually grazed separately after puberty, so Sex is a fixed effect. In the present dataset, males and females were measured at different ages (males at 12 months, females at 15 months) in some years, and both at 15 months in other years. So Agem is a fixed effect, but it only has 2 levels in some years, and in those years it is confounded with Sex. The female flock was divided into two grazing mobs, so Mob may have an effect but it could vary from year to year. Twin lambs suffer a penalty in wool production. There is an argument that adjusting for litter size (single or twin at either birth or weaning) as a fixed effect could remove some genetic variation as well as any environmental effects of twinning. So the effects Birls and Weanls are problematic and will require some consideration.

It is possible to use 'age of dam' as a factor instead of 'YearDbi' and this leads to an equivalent model. One can not fit both 'age of dam' and 'YearDbi' in the presence of 'Yearbi' as the three are completely confounded. We are going to stay with 'YearDbi'.

The easiest way to arrive at a satisfactory fixed effects model is to just use *aov()(* before trying to use *dmm())*

```
> junk <- aov(Ymat ~ 1 + C(Sex, sum) + C(Yearbi, sum) + C(YearDbi, sum) +
   C(Mob, sum) + C(Agem,sum) + C(Birls, sum) + C(Weanls, sum),merino.mdf$df)
> summary(junk)
 Response Birwt :
                  Df Sum Sq Mean Sq  F value    Pr(>F)
C(Sex, sum)        1   26.19  26.185 105.5230 < 2.2e-16 ***
C(Yearbi, sum)    14  120.54   8.610  34.6962 < 2.2e-16 ***
C(YearDbi, sum)   20   18.82   0.941   3.7925 2.778e-08 ***
C(Mob, sum)        1    0.22   0.222   0.8947    0.3443
C(Agem, sum)       1    0.19   0.194   0.7811    0.3769
C(Birls, sum)      1  174.90 174.898 704.8159 < 2.2e-16 ***
C(Weanls, sum)     1    0.42   0.415   1.6724    0.1961
Residuals       2381  590.84   0.248
---
 .....
 Response Woolwt :
```

```
                   Df Sum Sq Mean Sq    F value    Pr(>F)
C(Sex, sum)         1 546.61  546.61 3340.7998 < 2.2e-16 ***
C(Yearbi, sum)     14 447.67   31.98  195.4356 < 2.2e-16 ***
C(YearDbi, sum)    20   3.44    0.17    1.0523  0.395260
C(Mob, sum)         1   0.19    0.19    1.1596  0.281652
C(Agem, sum)        1 130.99  130.99  800.5997 < 2.2e-16 ***
C(Birls, sum)       1   6.47    6.47   39.5323 3.826e-10 ***
C(Weanls, sum)      1   1.21    1.21    7.3768  0.006655 **
Residuals        2381 389.57    0.16
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

1593 observations deleted due to missingness
>
> anova(junk)
Analysis of Variance Table

                   Df  Pillai approx F num Df den Df    Pr(>F)
(Intercept)         1 0.99951   435614     11   2371 < 2.2e-16 ***
C(Sex, sum)         1 0.81591      955     11   2371 < 2.2e-16 ***
C(Yearbi, sum)     14 2.91382       61    154  26191 < 2.2e-16 ***
C(YearDbi, sum)    20 0.16811        2    220  26191 4.198e-13 ***
C(Mob, sum)         1 0.00592        1     11   2371    0.2278
C(Agem, sum)        1 0.61968      351     11   2371 < 2.2e-16 ***
C(Birls, sum)       1 0.26530       78     11   2371 < 2.2e-16 ***
C(Weanls, sum)      1 0.04392       10     11   2371 < 2.2e-16 ***
Residuals        2381
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
>
```

So there is an immediate problem with multiple traits - some fixed effects are significant for some traits and not for others. For example Agem does not affect Birwt because it is not relevant to that trait. Weanls is significant only for Woolwt, Flcwt and Bodywt. Mob is significant only for Length and then only at $P = 0.05$.

We have to make an overall decision, because *dmm()* fits the same model to all traits. The manova table (see *anova(junk)* above) is probably the best guide. It shows all fixed effects are significant except mob. So we will fit all except mob, but there may be some debate later about Birls and Weanls being partly genetic.

9.3 Elementary partitioning of individual additive genetic and environmental variation

We should first do the usual analysis assuming additivity at the individual level.

```
> merino.fitia <- dmm(merino.mdf,Ymat ~ 1 + C(Sex, sum) + C(Yearbi, sum) +
    C(YearDbi, sum) + C(Agem,sum) + C(Birls, sum) + C(Weanls, sum),
    components = c("VarE(I)","VarG(Ia)"))
Dyadic mixed model fit for datafile: merino.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 2
OLS-b step:
no of fixed effect df (k) =  43
no of traits (l) =  11
Setup antemodel matrices:
no of individuals in pedigree (m) =  4014
no of individuals with data and X codes (n) =  2642
Rank of X: 43   No of Fixed Effects: 43
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
> merino.fitia$dme.corre
           VarE(I)   VarG(Ia)
VarE(I)  1.0000000 0.4169643
VarG(Ia) 0.4169643 1.0000000
```

Just as a matter if interest, this run took 5 mins real time and used 28Gb of memory in the author's PC. The correlation of **0.4169** between VarE(I) and VarG(Ia) is not a serious collinearity. We therefore inspect the results with some confidence.

```
> gsummary(merino.fitia)
Call:
gsummary.dmm(dmmobj = merino.fitia)


Proportion of phenotypic var/covariance partitioned by DME to each component (OLS-b):

         Trait Estimate StdErr CI95lo CI95hi
VarE(I)  Birwt    0.681 0.0109  0.660  0.702
VarG(Ia) Birwt    0.319 0.0109  0.298  0.340
VarP(I)  Birwt    1.000 0.0000  1.000  1.000


          Trait Estimate  StdErr CI95lo CI95hi
VarE(I)   Weanwt    0.805 0.00989  0.786  0.825
VarG(Ia)  Weanwt    0.195 0.00989  0.175  0.214
VarP(I)   Weanwt    1.000 0.00000  1.000  1.000


         Trait Estimate StdErr CI95lo CI95hi
VarE(I)  Crimp    0.544 0.0124  0.520  0.568
```

```
VarG(Ia) Crimp     0.456 0.0124   0.432   0.480
VarP(I)  Crimp     1.000 0.0000   1.000   1.000


         Trait Estimate StdErr CI95lo CI95hi
VarE(I)  Densty    0.434 0.0137   0.407   0.461
VarG(Ia) Densty    0.566 0.0138   0.539   0.593
VarP(I)  Densty    1.000 0.0000   1.000   1.000


         Trait Estimate StdErr CI95lo CI95hi
VarE(I)  Diamtr    0.467 0.0133   0.441   0.493
VarG(Ia) Diamtr    0.533 0.0133   0.507   0.559
VarP(I)  Diamtr    1.000 0.0000   1.000   1.000


         Trait Estimate StdErr CI95lo CI95hi
VarE(I)  Yield     0.634 0.0113   0.612   0.656
VarG(Ia) Yield     0.366 0.0113   0.344   0.388
VarP(I)  Yield     1.000 0.0000   1.000   1.000


         Trait Estimate  StdErr CI95lo CI95hi
VarE(I)  Bodywt    0.824 0.00978  0.804   0.843
VarG(Ia) Bodywt    0.176 0.00978  0.157   0.196
VarP(I)  Bodywt    1.000 0.00000  1.000   1.000


         Trait Estimate StdErr CI95lo CI95hi
VarE(I)  Wrinkl    0.563 0.0121   0.539   0.587
VarG(Ia) Wrinkl    0.437 0.0121   0.413   0.461
VarP(I)  Wrinkl    1.000 0.0000   1.000   1.000


         Trait Estimate StdErr CI95lo CI95hi
VarE(I)  Length    0.685 0.0108   0.663   0.706
VarG(Ia) Length    0.315 0.0108   0.294   0.337
VarP(I)  Length    1.000 0.0000   1.000   1.000


         Trait Estimate  StdErr CI95lo CI95hi
VarE(I)  Flcwt     0.875 0.00954  0.856   0.893
VarG(Ia) Flcwt     0.125 0.00955  0.107   0.144
VarP(I)  Flcwt     1.000 0.00000  1.000   1.000


         Trait Estimate  StdErr CI95lo CI95hi
VarE(I)  Woolwt    0.819 0.00981  0.800   0.839
VarG(Ia) Woolwt    0.181 0.00981  0.161   0.200
VarP(I)  Woolwt    1.000 0.00000  1.000   1.000


Correlation corresponding to each var/covariance component partitioned by DME (OLS-b):
 .....
         Traitpair Estimate StdErr CI95lo CI95hi
```

```
VarE(I)   Woolwt:Birwt       0.174 0.0272   0.121   0.228
VarG(Ia)  Woolwt:Birwt       0.295 0.0371   0.223   0.368
VarP(I)   Woolwt:Birwt       0.201 0.0185   0.165   0.237


               Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Woolwt:Weanwt      0.222 0.0248   0.173   0.270
VarG(Ia)  Woolwt:Weanwt      0.334 0.0468   0.242   0.425
VarP(I)   Woolwt:Weanwt      0.243 0.0183   0.207   0.278


               Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Woolwt:Crimp      -0.181 0.0301 -0.240 -0.122
VarG(Ia)  Woolwt:Crimp      -0.433 0.0306 -0.493 -0.373
VarP(I)   Woolwt:Crimp      -0.245 0.0181 -0.281 -0.210


               Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Woolwt:Densty      0.125 0.0344 0.0581   0.193
VarG(Ia)  Woolwt:Densty      0.193 0.0283 0.1377   0.248
VarP(I)   Woolwt:Densty      0.137 0.0186 0.1000   0.173


               Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Woolwt:Diamtr      0.239 0.0325   0.175   0.302
VarG(Ia)  Woolwt:Diamtr      0.158 0.0288   0.102   0.215
VarP(I)   Woolwt:Diamtr      0.197 0.0184   0.161   0.233


               Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Woolwt:Yield       0.356 0.0253   0.306   0.406
VarG(Ia)  Woolwt:Yield       0.716 0.0302   0.657   0.776
VarP(I)   Woolwt:Yield       0.441 0.0162   0.409   0.473


               Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Woolwt:Bodywt      0.4556 0.0225   0.412   0.500
VarG(Ia)  Woolwt:Bodywt      0.0226 0.0493  -0.074   0.119
VarP(I)   Woolwt:Bodywt      0.3783 0.0170   0.345   0.412


               Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)   Woolwt:Wrinkl      0.1955 0.0302   0.1363 0.2548
VarG(Ia)  Woolwt:Wrinkl     -0.0212 0.0331  -0.0861 0.0438
VarP(I)   Woolwt:Wrinkl      0.1269 0.0187   0.0901 0.1636


               Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Woolwt:Length      0.567 0.0223   0.523   0.611
VarG(Ia)  Woolwt:Length      0.294 0.0336   0.228   0.360
VarP(I)   Woolwt:Length      0.495 0.0155   0.464   0.525


               Traitpair Estimate  StdErr CI95lo CI95hi
VarE(I)   Woolwt:Flcwt       0.912 0.00863   0.895   0.929
```

```
VarG(Ia) Woolwt:Flcwt     0.783 0.02397  0.736  0.830
VarP(I)  Woolwt:Flcwt     0.890 0.00675  0.877  0.903


             Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)  Woolwt:Woolwt        1      0      1      1
VarG(Ia) Woolwt:Woolwt        1      0      1      1
VarP(I)  Woolwt:Woolwt        1      0      1      1
 .....
```

Just looking at the proportion of variance for "VarG(Ia)" (ie additive heritability) we find that most wool traits have a heritability of around 0.4 in agreement with published results (eg Brown and Turner(1968) [4]). The exceptions are Woolwt which is only 0.18 (and Flcwt which is almost the same thing except it includes grease and dirt) and Bodywt. This is unexpected so it bears further investigation.

Also listed are the genetic correlations of Woolwt with other traits. These are important in relation to selection for Woolwt. All these genetic correlations look reasonable and relate well to other published estimates. In particular the important genetic correlation of Woolwt with Diamtr is 0.16 agrees closely with published results.

If one were to list all the parameter estimates there would be a printout of around 20 pages. The reason for the *traitset* and *componentset* arguments to *gsummary()* should now be obvious. It is important to be able to browse subsets of the results.

The standard errors of parameter estimates all appear acceptable, and this is another good indicator of a reasonable model for the data and absence of collinearities.

The results for fitting this simple model would seem to be a sound base, and useful for comparison with other more complex models.

9.4 Investigation of maternal additive genetic and maternal environmental variation

We now extend the model to include maternal additive genetic and maternal environmental variances, and their genetic and environmental covariances with individual variation. The aim is to see if maternal effects on wool traits are large enough to warrant consideration in selection studies.

```
> merino.fitiama <- dmm(merino.mdf,Ymat ~ 1 + C(Sex, sum) + C(Yearbi, sum) +
    C(YearDbi, sum) + C(Agem,sum) + C(Birls, sum) + C(Weanls, sum),
    components = c("VarE(I)","VarG(Ia)", "VarE(M)", "VarG(Ma)", "CovG(Ia,Ma)",
    "CovG(Ma,Ia)","CovE(I,M)", "CovE(M,I)"))
Dyadic mixed model fit for datafile: merino.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 8
```

```
OLS-b step:
no of fixed effect df (k) =  43
no of traits (l) =  11
Setup antemodel matrices:
no of individuals in pedigree (m) =  4014
no of individuals with data and X codes (n) =  2642
Rank of X: 43   No of Fixed Effects: 43
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
>merino.fitiama$dme.corre
                VarE(I)  VarG(Ia)     VarE(M)  VarG(Ma)     CovE(I,M)
VarE(I)      1.000000e+00 0.4169643  4.860238e-01 0.3318052  6.577546e-06
VarG(Ia)     4.169643e-01 1.0000000  3.727328e-01 0.4781302  1.811512e-01
VarE(M)      4.860238e-01 0.3727328  1.000000e+00 0.6816931 -4.326229e-05
VarG(Ma)     3.318052e-01 0.4781302  6.816931e-01 1.0000000  1.443848e-01
CovE(I,M)    6.577546e-06 0.1811512 -4.326229e-05 0.1443848  1.000000e+00
CovE(M,I)    6.577546e-06 0.1811512 -4.326229e-05 0.1443848 -8.916420e-06
CovG(Ia,Ma)  2.109631e-01 0.5930327  4.337594e-01 0.6950698  3.702673e-01
CovG(Ma,Ia)  2.109631e-01 0.5930327  4.337594e-01 0.6950698  9.175759e-02
               CovE(M,I) CovG(Ia,Ma) CovG(Ma,Ia)
VarE(I)       6.577546e-06  0.21096311  0.21096311
VarG(Ia)      1.811512e-01  0.59303269  0.59303269
VarE(M)      -4.326229e-05  0.43375940  0.43375940
VarG(Ma)      1.443848e-01  0.69506983  0.69506983
CovE(I,M)    -8.916420e-06  0.37026729  0.09175759
CovE(M,I)     1.000000e+00  0.09175759  0.37026729
CovG(Ia,Ma)   9.175759e-02  1.00000000  0.47942974
CovG(Ma,Ia)   3.702673e-01  0.47942974  1.00000000
>
```

The above run took 9 mins real time and used 29Gb of memory, so adding
6 more components has only doubled the processing time.

The correlations among columns of the dyadic model equation matrix ($\boldsymbol{W}$)
are surprisingly good, the largest being 0.69 between VarG(Ma) and CovG(Ia,Ma).
We can proceed with some confidence to look at parameter estimates. Just 3
traits for a start (the gsummary output for all traits is now so long it exceeds
my scrollback length).

```
> gsummary(merino.fitiama, traitset=c("Birwt","Diamtr","Woolwt"))
Call:
gsummary.dmm(dmmobj = merino.fitiama, traitset = c("Birwt", "Diamtr",
    "Woolwt"))


Proportion of phenotypic var/covariance partitioned by DME to each component (OLS-b):
```

```
           Trait Estimate StdErr  CI95lo   CI95hi
VarE(I)     Birwt    0.4137 0.0174  0.3795   0.44788
VarG(Ia)    Birwt    0.1961 0.0129  0.1708   0.22146
VarE(M)     Birwt    0.1404 0.0117  0.1175   0.16329
VarG(Ma)    Birwt    0.2243 0.0141  0.1966   0.25197
CovE(I,M)   Birwt    0.0399 0.0187  0.0033   0.07645
CovE(M,I)   Birwt    0.0399 0.0187  0.0033   0.07645
CovG(Ia,Ma) Birwt   -0.0271 0.0119 -0.0505  -0.00377
CovG(Ma,Ia) Birwt   -0.0271 0.0119 -0.0505  -0.00377
VarP(I)     Birwt    1.0000 0.0000  1.0000   1.00000

           Trait Estimate StdErr   CI95lo   CI95hi
VarE(I)     Diamtr   0.4670 0.0217  0.42451  0.50956
VarG(Ia)    Diamtr   0.6156 0.0280  0.56076  0.67042
VarE(M)     Diamtr   0.0501 0.0138  0.02302  0.07721
VarG(Ma)    Diamtr   0.0185 0.0140 -0.00895  0.04598
CovE(I,M)   Diamtr  -0.0388 0.0249 -0.08765  0.00998
CovE(M,I)   Diamtr  -0.0388 0.0249 -0.08765  0.00998
CovG(Ia,Ma) Diamtr  -0.0368 0.0145 -0.06523 -0.00835
CovG(Ma,Ia) Diamtr  -0.0368 0.0145 -0.06523 -0.00835
VarP(I)     Diamtr   1.0000 0.0000  1.00000  1.00000

           Trait Estimate StdErr  CI95lo  CI95hi
VarE(I)     Woolwt   0.6994 0.0235  0.6533  0.7455
VarG(Ia)    Woolwt   0.1899 0.0142  0.1620  0.2178
VarE(M)     Woolwt   0.0357 0.0126  0.0109  0.0604
VarG(Ma)    Woolwt   0.0797 0.0133  0.0536  0.1057
CovE(I,M)   Woolwt   0.0505 0.0205  0.0103  0.0907
CovE(M,I)   Woolwt   0.0505 0.0205  0.0103  0.0907
CovG(Ia,Ma) Woolwt  -0.0528 0.0133 -0.0789 -0.0266
CovG(Ma,Ia) Woolwt  -0.0528 0.0133 -0.0789 -0.0266
VarP(I)     Woolwt   1.0000 0.0000  1.0000  1.0000


Correlation corresponding to each var/covariance component partitioned by DME (OLS-b):

           Traitpair  Estimate StdErr   CI95lo   CI95hi
VarE(I)     Birwt:Birwt   1.000 0.0000  1.00000  1.0000
VarG(Ia)    Birwt:Birwt   1.000 0.0000  1.00000  1.0000
VarE(M)     Birwt:Birwt   1.000 0.0000  1.00000  1.0000
VarG(Ma)    Birwt:Birwt   1.000 0.0000  1.00000  1.0000
CovE(I,M)   Birwt:Birwt   0.165 0.0804  0.00791  0.3230
CovE(M,I)   Birwt:Birwt   0.165 0.0804  0.00791  0.3230
CovG(Ia,Ma) Birwt:Birwt  -0.129 0.0527 -0.23261 -0.0261
CovG(Ma,Ia) Birwt:Birwt  -0.129 0.0527 -0.23261 -0.0261
```

```
VarP(I)       Birwt:Birwt    1.000 0.0000  1.00000  1.0000


             Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)      Birwt:Diamtr   0.0118 0.0488 -0.0838  0.107
VarG(Ia)     Birwt:Diamtr  -0.2417 0.0336 -0.3075 -0.176
VarE(M)      Birwt:Diamtr   0.3388 0.1653  0.0147  0.663
VarG(Ma)     Birwt:Diamtr   0.8375 0.3687  0.1149  1.560
CovE(I,M)    Birwt:Diamtr   0.5157 0.1676  0.1872  0.844
CovE(M,I)    Birwt:Diamtr   0.0790 0.0842 -0.0861  0.244
CovG(Ia,Ma)  Birwt:Diamtr  -0.2671 0.2537 -0.7644  0.230
CovG(Ma,Ia)  Birwt:Diamtr  -0.1086 0.0355 -0.1781 -0.039
VarP(I)      Birwt:Diamtr   0.0417 0.0341 -0.0251  0.108


             Traitpair Estimate StdErr   CI95lo  CI95hi
VarE(I)      Birwt:Woolwt    0.153 0.0366  0.08112  0.2245
VarG(Ia)     Birwt:Woolwt    0.313 0.0553  0.20441  0.4212
VarE(M)      Birwt:Woolwt    0.329 0.1665  0.00307  0.6557
VarG(Ma)     Birwt:Woolwt    0.546 0.0901  0.36901  0.7223
CovE(I,M)    Birwt:Woolwt    0.470 0.1889  0.10012  0.8408
CovE(M,I)    Birwt:Woolwt    0.078 0.0656 -0.05060  0.2067
CovG(Ia,Ma)  Birwt:Woolwt   -0.248 0.0993 -0.44216 -0.0530
CovG(Ma,Ia)  Birwt:Woolwt   -0.184 0.0599 -0.30120 -0.0664
VarP(I)      Birwt:Woolwt    0.252 0.0313  0.19021  0.3128


             Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)      Diamtr:Birwt   0.0118 0.0488 -0.0838  0.107
VarG(Ia)     Diamtr:Birwt  -0.2417 0.0336 -0.3075 -0.176
VarE(M)      Diamtr:Birwt   0.3388 0.1653  0.0147  0.663
VarG(Ma)     Diamtr:Birwt   0.8375 0.3687  0.1149  1.560
CovE(I,M)    Diamtr:Birwt   0.0790 0.0842 -0.0861  0.244
CovE(M,I)    Diamtr:Birwt   0.5157 0.1676  0.1872  0.844
CovG(Ia,Ma)  Diamtr:Birwt  -0.1086 0.0355 -0.1781 -0.039
CovG(Ma,Ia)  Diamtr:Birwt  -0.2671 0.2537 -0.7644  0.230
VarP(I)      Diamtr:Birwt   0.0417 0.0341 -0.0251  0.108


             Traitpair Estimate StdErr CI95lo  CI95hi
VarE(I)      Diamtr:Diamtr   1.000  0.000  1.000  1.0000
VarG(Ia)     Diamtr:Diamtr   1.000  0.000  1.000  1.0000
VarE(M)      Diamtr:Diamtr   1.000  0.000  1.000  1.0000
VarG(Ma)     Diamtr:Diamtr   1.000  0.000  1.000  1.0000
CovE(I,M)    Diamtr:Diamtr  -0.254  0.165 -0.577  0.0694
CovE(M,I)    Diamtr:Diamtr  -0.254  0.165 -0.577  0.0694
CovG(Ia,Ma)  Diamtr:Diamtr  -0.345  0.113 -0.566 -0.1233
CovG(Ma,Ia)  Diamtr:Diamtr  -0.345  0.113 -0.566 -0.1233
VarP(I)      Diamtr:Diamtr   1.000  0.000  1.000  1.0000
```

```
                 Traitpair Estimate StdErr  CI95lo   CI95hi
VarE(I)      Diamtr:Woolwt    0.225 0.0377   0.151   0.2986
VarG(Ia)     Diamtr:Woolwt    0.187 0.0347   0.119   0.2555
VarE(M)      Diamtr:Woolwt    0.450 0.2992  -0.137   1.0363
VarG(Ma)     Diamtr:Woolwt    0.913 0.4136   0.102   1.7239
CovE(I,M)    Diamtr:Woolwt   -0.258 0.1854  -0.621   0.1055
CovE(M,I)    Diamtr:Woolwt    0.139 0.1229  -0.102   0.3795
CovG(Ia,Ma)  Diamtr:Woolwt   -0.039 0.0612  -0.159   0.0809
CovG(Ma,Ia)  Diamtr:Woolwt   -0.600 0.2966  -1.182  -0.0192
VarP(I)      Diamtr:Woolwt    0.195 0.0346   0.127   0.2629

                 Traitpair Estimate StdErr   CI95lo   CI95hi
VarE(I)      Woolwt:Birwt     0.153 0.0366  0.08112   0.2245
VarG(Ia)     Woolwt:Birwt     0.313 0.0553  0.20441   0.4212
VarE(M)      Woolwt:Birwt     0.329 0.1665  0.00307   0.6557
VarG(Ma)     Woolwt:Birwt     0.546 0.0901  0.36901   0.7223
CovE(I,M)    Woolwt:Birwt     0.078 0.0656 -0.05060   0.2067
CovE(M,I)    Woolwt:Birwt     0.470 0.1889  0.10012   0.8408
CovG(Ia,Ma)  Woolwt:Birwt    -0.184 0.0599 -0.30120  -0.0664
CovG(Ma,Ia)  Woolwt:Birwt    -0.248 0.0993 -0.44216  -0.0530
VarP(I)      Woolwt:Birwt     0.252 0.0313  0.19021   0.3128

                 Traitpair Estimate StdErr CI95lo   CI95hi
VarE(I)      Woolwt:Diamtr    0.225 0.0377   0.151   0.2986
VarG(Ia)     Woolwt:Diamtr    0.187 0.0347   0.119   0.2555
VarE(M)      Woolwt:Diamtr    0.450 0.2992  -0.137   1.0363
VarG(Ma)     Woolwt:Diamtr    0.913 0.4136   0.102   1.7239
CovE(I,M)    Woolwt:Diamtr    0.139 0.1229  -0.102   0.3795
CovE(M,I)    Woolwt:Diamtr   -0.258 0.1854  -0.621   0.1055
CovG(Ia,Ma)  Woolwt:Diamtr   -0.600 0.2966  -1.182  -0.0192
CovG(Ma,Ia)  Woolwt:Diamtr   -0.039 0.0612  -0.159   0.0809
VarP(I)      Woolwt:Diamtr    0.195 0.0346   0.127   0.2629

                 Traitpair Estimate StdErr CI95lo  CI95hi
VarE(I)      Woolwt:Woolwt    1.000 0.0000  1.0000  1.000
VarG(Ia)     Woolwt:Woolwt    1.000 0.0000  1.0000  1.000
VarE(M)      Woolwt:Woolwt    1.000 0.0000  1.0000  1.000
VarG(Ma)     Woolwt:Woolwt    1.000 0.0000  1.0000  1.000
CovE(I,M)    Woolwt:Woolwt    0.320 0.1422  0.0408  0.598
CovE(M,I)    Woolwt:Woolwt    0.320 0.1422  0.0408  0.598
CovG(Ia,Ma)  Woolwt:Woolwt   -0.429 0.0866 -0.5987 -0.259
CovG(Ma,Ia)  Woolwt:Woolwt   -0.429 0.0866 -0.5987 -0.259
VarP(I)      Woolwt:Woolwt    1.000 0.0000  1.0000  1.000
```

Phenotypic var/covariance from components partitioned by DME (OLS-b):

```
      Traitpair Estimate  StdErr  CI95lo CI95hi
1   Birwt:Birwt    0.3089 0.00950  0.2903 0.3275
2 Birwt:Diamtr    0.0469 0.03812 -0.0279 0.1216
3  Birwt:Woolwt    0.0603 0.00777  0.0451 0.0755
4  Diamtr:Birwt    0.0469 0.03812 -0.0279 0.1216
5 Diamtr:Diamtr    4.0907 0.15276  3.7913 4.3902
6 Diamtr:Woolwt    0.1701 0.03116  0.1091 0.2312
7  Woolwt:Birwt    0.0603 0.00777  0.0451 0.0755
8 Woolwt:Diamtr    0.1701 0.03116  0.1091 0.2312

9 Woolwt:Woolwt    0.1861 0.00635  0.1736 0.1985

>
```

There is very little maternal additive genetic variance, except for traits 'Birwt' and 'Weanwt'. There is a slight amount for 'Bodywt', 'Flcwt', and 'Woolwt'. For wool traits other than weight there is near-zero "VarG(Ma)". The genetic and environmental covariances of individual and maternal effcts are negligable for all traits.

The possibility that fitting 'Birls' and 'Weanls' fixed effects had obscured any maternal efects was investigated with a run omitting the above two fixed effects. The result was almost the same as above, and is not reported.

9.5 Investigation of individual additive genetic sexlinked variation

We now try extending the model in a different direction. We wish to see if there is evidence for existence of genetic sexlinked variation, so we fit "VarGs(Ia)" in addition to the basic model of "VarE(I)" + "VarG(Ia)". In this context, with "VarGs(Ia)" fitted, "VarG(Ia)" should be interpreted as autosomal additive genetic variance.

```
> merino.fitiasl <- dmm(merino.mdf,Ymat ~ 1 + C(Sex, sum) + C(Yearbi, sum) +
    C(YearDbi, sum) + C(Agem, sum) + C(Birls,sum) + C(Weanls, sum),
    components = c("VarE(I)", "VarG(Ia)","VarGs(Ia)"))
Dyadic mixed model fit for datafile: merino.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 3
OLS-b step:
no of fixed effect df (k) =  43
no of traits (l) =  11
Setup antemodel matrices:
no of individuals in pedigree (m) =  4014
no of individuals with data and X codes (n) =  2642
```

```
Rank of X: 43   No of Fixed Effects: 43
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
>
> gsummary(merino.fitiasl)
Call:
gsummary.dmm(dmmobj = merino.fitiasl)

Proportion of phenotypic var/covariance partitioned by DME to each component (OLS-b):

          Trait Estimate  StdErr CI95lo CI95hi
VarE(I)   Birwt     0.208 0.02229  0.165  0.252
VarG(Ia)  Birwt     0.194 0.00897  0.176  0.211
VarGs(Ia) Birwt     0.598 0.02240  0.554  0.642
VarP(I)   Birwt     1.000 0.00000  1.000  1.000


           Trait Estimate StdErr CI95lo CI95hi
VarE(I)   Weanwt     0.442 0.0231 0.3972  0.488
VarG(Ia)  Weanwt     0.118 0.0098 0.0991  0.138
VarGs(Ia) Weanwt     0.439 0.0242 0.3917  0.487
VarP(I)   Weanwt     1.000 0.0000 1.0000  1.000


          Trait Estimate  StdErr CI95lo CI95hi
VarE(I)   Crimp     0.409 0.01996  0.370  0.448
VarG(Ia)  Crimp     0.367 0.00985  0.348  0.386
VarGs(Ia) Crimp     0.224 0.01984  0.185  0.263
VarP(I)   Crimp     1.000 0.00000  1.000  1.000


           Trait Estimate StdErr  CI95lo CI95hi
VarE(I)   Densty 0.50436 0.0191  0.4669 0.5418
VarG(Ia)  Densty 0.49331 0.0109  0.4719 0.5147
VarGs(Ia) Densty 0.00233 0.0193 -0.0355 0.0402
VarP(I)   Densty 1.00000 0.0000  1.0000 1.0000


           Trait Estimate  StdErr CI95lo CI95hi
VarE(I)   Diamtr   0.5116 0.01746 0.4774 0.5459
VarG(Ia)  Diamtr   0.4314 0.00944 0.4129 0.4499
VarGs(Ia) Diamtr   0.0569 0.01768 0.0223 0.0916
VarP(I)   Diamtr   1.0000 0.00000 1.0000 1.0000


          Trait Estimate StdErr CI95lo CI95hi
VarE(I)   Yield     0.534 0.0234 0.4879  0.579
VarG(Ia)  Yield     0.344 0.0118 0.3209  0.367
VarGs(Ia) Yield     0.122 0.0239 0.0755  0.169
```

```
VarP(I)    Yield     1.000 0.0000 1.0000  1.000

           Trait Estimate StdErr CI95lo CI95hi
VarE(I)    Bodywt     0.631 0.0226  0.586  0.675
VarG(Ia)   Bodywt     0.137 0.0102  0.117  0.157
VarGs(Ia)  Bodywt     0.233 0.0241  0.186  0.280
VarP(I)    Bodywt     1.000 0.0000  1.000  1.000

           Trait Estimate StdErr CI95lo CI95hi
VarE(I)    Wrinkl    0.5160 0.0226 0.4717  0.560
VarG(Ia)   Wrinkl    0.4149 0.0121 0.3912  0.439
VarGs(Ia)  Wrinkl    0.0691 0.0229 0.0242  0.114
VarP(I)    Wrinkl    1.0000 0.0000 1.0000  1.000

           Trait Estimate  StdErr CI95lo CI95hi
VarE(I)    Length     0.547 0.01862  0.510  0.583
VarG(Ia)   Length     0.251 0.00876  0.234  0.268
VarGs(Ia)  Length     0.202 0.01928  0.164  0.240
VarP(I)    Length     1.000 0.00000  1.000  1.000

           Trait Estimate  StdErr CI95lo CI95hi
VarE(I)    Flcwt     0.6305 0.02083 0.5897 0.6713
VarG(Ia)   Flcwt     0.0808 0.00922 0.0627 0.0988
VarGs(Ia)  Flcwt     0.2888 0.02236 0.2449 0.3326
VarP(I)    Flcwt     1.0000 0.00000 1.0000 1.0000

           Trait Estimate  StdErr CI95lo CI95hi
VarE(I)    Woolwt     0.657 0.02116  0.615  0.698
VarG(Ia)   Woolwt     0.147 0.00964  0.128  0.165
VarGs(Ia)  Woolwt     0.196 0.02254  0.152  0.241
VarP(I)    Woolwt     1.000 0.00000  1.000  1.000

Correlation corresponding to each var/covariance component:
 partitioned by DME (OLS-b):

 ....
           Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Woolwt:Birwt  -0.0433 0.0717 -0.184 0.0973
VarG(Ia)   Woolwt:Birwt   0.2115 0.0512  0.111 0.3118
VarGs(Ia)  Woolwt:Birwt   0.6284 0.0640  0.503 0.7538
VarP(I)    Woolwt:Birwt   0.2350 0.0162  0.203 0.2668

           Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Woolwt:Weanwt    0.198 0.0484  0.103  0.292
VarG(Ia)   Woolwt:Weanwt    0.347 0.0681  0.214  0.481
VarGs(Ia)  Woolwt:Weanwt    0.467 0.0730  0.323  0.610
```

```
VarP(I)    Woolwt:Weanwt      0.289 0.0170  0.256   0.323


          Traitpair Estimate StdErr CI95lo  CI95hi
VarE(I)    Woolwt:Crimp   -0.0175 0.0467 -0.109   0.0741
VarG(Ia)   Woolwt:Crimp   -0.3813 0.0360 -0.452  -0.3107
VarGs(Ia)  Woolwt:Crimp   -0.6502 0.0948 -0.836  -0.4644
VarP(I)    Woolwt:Crimp   -0.2339 0.0158 -0.265  -0.2030


          Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)    Woolwt:Densty  -0.0171 0.0442 -0.1036 0.0695
VarG(Ia)   Woolwt:Densty   0.1421 0.0319  0.0795 0.2047
VarGs(Ia)  Woolwt:Densty   0.8900 1.2007 -1.4633 3.2434
VarP(I)    Woolwt:Densty   0.0475 0.0163  0.0155 0.0794


          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Woolwt:Diamtr    0.208 0.0399 0.1297  0.286
VarG(Ia)   Woolwt:Diamtr    0.160 0.0323 0.0972  0.224
VarGs(Ia)  Woolwt:Diamtr    0.769 0.1981 0.3809  1.158
VarP(I)    Woolwt:Diamtr    0.242 0.0152 0.2125  0.272


          Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)    Woolwt:Yield     0.391 0.0404  0.3121  0.471
VarG(Ia)   Woolwt:Yield     0.773 0.0366  0.7009  0.845
VarGs(Ia)  Woolwt:Yield     0.231 0.1301 -0.0242  0.486
VarP(I)    Woolwt:Yield     0.441 0.0156  0.4105  0.472


          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Woolwt:Bodywt    0.4281 0.0379  0.354 0.5023
VarG(Ia)   Woolwt:Bodywt   -0.0895 0.0722 -0.231 0.0519
VarGs(Ia)  Woolwt:Bodywt    0.4269 0.0940  0.243 0.6111
VarP(I)    Woolwt:Bodywt    0.3542 0.0163  0.322 0.3860


          Traitpair Estimate StdErr  CI95lo CI95hi
VarE(I)    Woolwt:Wrinkl    0.2142 0.0459  0.1242 0.3042
VarG(Ia)   Woolwt:Wrinkl   -0.0230 0.0390 -0.0995 0.0535
VarGs(Ia)  Woolwt:Wrinkl   -0.0365 0.2342 -0.4956 0.4226
VarP(I)    Woolwt:Wrinkl    0.1148 0.0177  0.0802 0.1494


          Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Woolwt:Length    0.551 0.0331  0.486  0.616
VarG(Ia)   Woolwt:Length    0.277 0.0393  0.200  0.355
VarGs(Ia)  Woolwt:Length    0.690 0.0803  0.533  0.848
VarP(I)    Woolwt:Length    0.521 0.0131  0.495  0.547


          Traitpair Estimate  StdErr CI95lo CI95hi
VarE(I)    Woolwt:Flcwt     0.927 0.01419  0.899  0.955
```

```
VarG(Ia)   Woolwt:Flcwt     0.783 0.03260  0.719  0.847
VarGs(Ia)  Woolwt:Flcwt     0.895 0.03237  0.832  0.958
VarP(I)    Woolwt:Flcwt     0.895 0.00612  0.883  0.907

             Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Woolwt:Woolwt       1      0      1      1
VarG(Ia)   Woolwt:Woolwt       1      0      1      1
VarGs(Ia)  Woolwt:Woolwt       1      0      1      1
VarP(I)    Woolwt:Woolwt       1      0      1      1

 .....

>
```

First look at the proportions. There is a significant "VarGs(Ia)" proportion for Birwt, Weanwt, Crimp, Yield, Bodywt, Length, Flcwt, and Woolwt. In most cases this extra component of variance has come out of VarE(I), which is thereby reduced.

For Woolwt and Flcwt we now have 2 genetic components of variance (VarG(Ia) and VarGs(Ia)) which add up to about 0.35 of the phenotypic variance. This may explain why published results for heritability of Woolwt of around 0.4 disagree with the current analysis. Older published estimates come from either paternal half-sisters or dam-daughter analyses, which would be expected to include sexlinked genetic variance (Fairbairn and Roff(2006) [6]). Matthew Wolak (personal communication) has conducted an extensive study of the effect of ignoring sexlinked additive genetic variance components when estimating additive genetic variance components (presumed autosomal), and has concluded that even in analyses using full pedigree information there is likely to be a bias in estimates of additive genetic variance. In the present analysis based on full pedigree information and including male obsevations the sexlinked variation of Woolwt exists, so the the "VarG(Ia)" component estimate may be biased when sexlinkage is ignored, but should be unbiased when "VarGs(Ia)" is also fitted. For other traits, the sexlinked component is small, and they end up with heritabilities which agree with published figures.

If there is a real sexlinked additive genetic component for Flcwt and Woolwt, it is interesting to ask which physical components of wool weight show it. It can be seen that there is no "VarGs(Ia)" for Densty or Diamtr or Wrinkl, but there is for Length and Bodywt. So the components which are set early in life do not show it.

The standard errors of these proportions are sometimes a little larger than those when only VarE(I) and VarG(Ia) are fitted, but there is nothing to indicate an overfitting or too much collinearity.

The genetic correlations require more caution in interpretation. The standard errors are larger than when just VarE(I) and VarG(Ia) are fitted. One should also be wary of attaching too much importance to genetic correlations for traits for which the corresponding proportion(s) are close to zero. For example,

the important Diamtr x Woolwt correlation - for VarG(Ia) the 0.16 is believable and significant and agrees with published results (Brown and Turner(1968) [4]), but for VarGs(Ia) the 0.76 should be ignored. Correlations among delicately balanced near zero quantities are not of interest. On the other hand, the 0.69 correlation for VarGs(Ia) for Length x Woolwt is large and significant and means something.

The consequences of a VarGs(Ia) component for selection are interesting. Genetic change in a population can be divided into four pathways

- male-to-male

- male-to-female

- female-to-male

- female-to-female

If there is a lot of sexlinked additive genetic variance, the male-to-male pathway is ineffective, because a male passes its X chromasome only to daughters. This may explain why breeding programs involving selection for Woolwt are sometimes less effective than anticipated . Selection tends to be concentrated on males because the intensity can be higher and the male-to-male generation interval is small.

A more comprehensive study of the consequences of selection for Woolwt under this sexlinked additive model must await further development of the *gresponse()* function, which currently does not have the facility to deal with sexlinked components of variation.

There is also a need to check the above conclusions with other datasets. A recognition of the presence of sexlinked additive genetic variation for wool traits in Australian Merino sheep would have consequences for the implementation of breeding plans. One needs to be sure of its correctness and general applicability before proceeding with field recommendations. One simple independent check would be to look separately at the genetic superiority of male and female offspring of selected males.

There is one additional check which we can do. We can rerun the analysis using the *dmeopt="pcr"* argument. This will tell us how serious the collinearities are and offers an option to remove them by constraining the estimates (see section 7.4). Unfortunately the model with 11 traits exceeds the array size limit in R if "pcr" is used. What we can do is revert to just one trait.

```
> merinokeep.mdf <- mdf(merino.df,pedcols=c(1:3),factorcols=c(4:9,12:13),
    ycols=c(10:11,14:22),sexcode=c("M","F"),
    relmat=c("E","A","S.hopi"),keep=T)
 .....
>
> merino.fitiaslpcr <- dmm(merinokeep.mdf, Woolwt ~ 1 + C(Sex, sum) +
    C(Yearbi, sum) + C(YearDbi, sum) + C(Agem, sum) + C(Birls,sum) +
    C(Weanls, sum), components = c("VarE(I)", "VarG(Ia)","VarGs(Ia)"),
```

```
      dmeopt="pcr")
Dyadic mixed model fit for datafile: merinokeep.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 3
OLS-b step:
no of fixed effect df (k) =   43
no of traits (l) =   1
Setup antemodel matrices:
no of individuals in pedigree (m) =   4014
no of individuals with data and X codes (n) =   2735
Rank of X: 43   No of Fixed Effects: 43
DME substep:
PCR option on dyadic model equations:
 .....
TRAINING: % variance explained
       1 comps   2 comps    3 comps
X      77.81599  96.22261  100.00000
evec    0.03008   0.03997    0.04277
DME substep completed:
OLS-b step completed:
>
> attributes(merino.fitiaslpcr)
$names
 [1] "aov"                    "mdf"                    "fixform"
 [4] "b"                      "seb"                    "vara"
 [7] "totn"                   "degf"                   "dme.mean"
[10] "dme.var"                "dme.correl"             "pcr.loadings"
[13] "dmeopt"                 "siga"                   "sesiga"
[16] "vard"                   "degfd"                  "component"
[19] "correlation"            "correlation.variance"   "correlation.se"
[22] "fraction"               "fraction.variance"      "fraction.se"
[25] "variance.components"    "variance.components.se" "phenotypic.variance"
[28] "phenotypic.variance.se" "observed.variance"      "call"

$class
[1] "dmm"

> merino.fitiaslpcr$pcr.loadings

Loadings:
          Comp 1 Comp 2 Comp 3
'VarE(I)'    0.235 -0.563  0.792
'VarG(Ia)'   0.892  0.450
'VarGs(Ia)'  0.387 -0.693 -0.608
```

```
             Comp 1 Comp 2 Comp 3
SS loadings     1.000  1.000  1.000
Proportion Var  0.333  0.333  0.333
Cumulative Var  0.333  0.667  1.000
>
```

The "% variance explained" shows that we can delete one principal component and still analyse 96.22 percent of the variation. The loadings show that if we proceed with just 2 principal components we are applying the constraint

$$0.792 \times VarE(I) - 0.608 \times VarGs(Ia) = 0$$

which defines a plane in 3D component space to which the estimates are confined. If we proceed with two principal components we get the following

```
> merino.fitiaslpcr2 <- dmm(merinokeep.mdf, Woolwt ~ 1 + C(Sex, sum) +
    C(Yearbi, sum) + C(YearDbi, sum) + C(Agem, sum) + C(Birls,sum) +
    C(Weanls, sum), components = c("VarE(I)", "VarG(Ia)","VarGs(Ia)"),
    dmeopt="pcr",dmekeepfit=T,ncomp=2)
Dyadic mixed model fit for datafile: merinokeep.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 3
OLS-b step:
no of fixed effect df (k) =  43
no of traits (l) =  1
Setup antemodel matrices:
no of individuals in pedigree (m) =  4014
no of individuals with data and X codes (n) =  2735
Rank of X: 43   No of Fixed Effects: 43
DME substep:
PCR option on dyadic model equations:
 .....
DME substep completed:
OLS-b step completed:
>
> gsummary(merino.fitiaslpcr2)
Call:
gsummary.dmm(dmmobj = merino.fitiaslpcr2)


Proportion of phenotypic var/covariance partitioned by DME to each component (OLS-b):

          Trait Estimate  StdErr CI95lo CI95hi
VarE(I)    Woolwt    0.365 0.00733  0.351  0.379
VarG(Ia)   Woolwt    0.147 0.01429  0.119  0.175
VarGs(Ia)  Woolwt    0.488 0.00730  0.474  0.503
VarP(I)    Woolwt    1.000 0.00000  1.000  1.000
```

```
Correlation corresponding to each var/covariance component partitioned by DME (OLS-b):

             Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)   Woolwt:Woolwt        1      0      1      1
VarG(Ia)  Woolwt:Woolwt        1      0      1      1
VarGs(Ia) Woolwt:Woolwt        1      0      1      1
VarP(I)   Woolwt:Woolwt        1      0      1      1


Phenotypic var/covariance from components partitioned by DME (OLS-b):

      Traitpair Estimate  StdErr CI95lo CI95hi
1 Woolwt:Woolwt    0.146 0.00472  0.136  0.155

>
```

These estimates compare favourably with those obtained with the default *dmeopt="qr"* argument. The standard errors are smaller, the estimates are similar, and the assumption of independence is not violated because the two principal components used are uncorrelated.

We conclude that one can have some confidence in the default *dmeopt="qr"* results and that the presence of an individual additive genetic sexlinked component of variance is real and of a significant magnitude.

9.6 Investigation of maternal additive genetic sexlinked variation

It is possible for maternal additive effects to be sexlinked. To investigate this we need to add the component "VarGs(Ma)". It would not make sense to fit this component without also fitting a maternal additive autosomal componment. So we need to resurrect "VarG(Ma)" even though we have already concluded that it is insignificant for most traits. Also to fit maternal genetic effects without a corresponding maternal environmental effect would be inappropriate, so we also resurrect "VarE(M)". We will, however, avoid the covariances (individual x maternal) previously fitted. We fit a 6 component model as follows.

```
> merino.fitiamasl <- dmm(merino.mdf,Ymat ~ 1 + C(Sex, sum) + C(Yearbi, sum) +
    C(YearDbi, sum) + C(Agem,sum) + C(Birls, sum) + C(Weanls, sum),
    components = c("VarE(I)","VarG(Ia)","VarGs(Ia)","VarE(M)","VarG(Ma)",
    "VarGs(Ma)"))
Dyadic mixed model fit for datafile: merino.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 6
```

```
OLS-b step:
no of fixed effect df (k) =  43
no of traits (l) =  11
Setup antemodel matrices:
no of individuals in pedigree (m) =  4014
no of individuals with data and X codes (n) =  2642
Rank of X: 43   No of Fixed Effects: 43
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:
> merino.fitiamasl$dme.corre
           VarE(I)   VarG(Ia) VarGs(Ia)   VarE(M)  VarG(Ma) VarGs(Ma)
VarE(I)   1.0000000 0.4169643 0.7540543 0.4860238 0.3318052 0.3923530
VarG(Ia)  0.4169643 1.0000000 0.5575606 0.3727328 0.4781302 0.4516340
VarGs(Ia) 0.7540543 0.5575606 1.0000000 0.6578486 0.6740459 0.7968618
VarE(M)   0.4860238 0.3727328 0.6578486 1.0000000 0.6816931 0.8072148
VarG(Ma)  0.3318052 0.4781302 0.6740459 0.6816931 1.0000000 0.8462307
VarGs(Ma) 0.3923530 0.4516340 0.7968618 0.8072148 0.8462307 1.0000000
>

> gsummary(merino.fitiamasl,traitset=c("Birwt","Diamtr","Woolwt"))
Call:
gsummary.dmm(dmmobj = merino.fitiamasl, traitset = c("Birwt",
    "Diamtr", "Woolwt"))


Proportion of phenotypic var/covariance partitioned by DME to each component (OLS-b):

          Trait Estimate  StdErr CI95lo CI95hi
VarE(I)   Birwt   0.3802 0.02192 0.3373  0.423
VarG(Ia)  Birwt   0.1365 0.00659 0.1236  0.149
VarGs(Ia) Birwt   0.0987 0.02682 0.0461  0.151
VarE(M)   Birwt   0.0720 0.01122 0.0500  0.094
VarG(Ma)  Birwt   0.0986 0.00804 0.0829  0.114
VarGs(Ma) Birwt   0.2139 0.01557 0.1834  0.244
VarP(I)   Birwt   1.0000 0.00000 1.0000  1.000


          Trait Estimate  StdErr  CI95lo CI95hi
VarE(I)   Diamtr  0.40157 0.02475  0.3531 0.4501
VarG(Ia)  Diamtr  0.42181 0.00912  0.4039 0.4397
VarGs(Ia) Diamtr  0.08907 0.03038  0.0295 0.1486
VarE(M)   Diamtr  0.06726 0.01271  0.0424 0.0922
VarG(Ma)  Diamtr  0.00812 0.00904 -0.0096 0.0258
VarGs(Ma) Diamtr  0.01218 0.01738 -0.0219 0.0462
VarP(I)   Diamtr  1.00000 0.00000  1.0000 1.0000
```

```
            Trait Estimate  StdErr  CI95lo CI95hi
VarE(I)    Woolwt  0.58630 0.02550  0.5363 0.6363
VarG(Ia)   Woolwt  0.12212 0.00787  0.1067 0.1375
VarGs(Ia)  Woolwt  0.15043 0.03223  0.0873 0.2136
VarE(M)    Woolwt  0.03789 0.01344  0.0116 0.0642
VarG(Ma)   Woolwt  0.00493 0.00957 -0.0138 0.0237
VarGs(Ma)  Woolwt  0.09832 0.01847  0.0621 0.1345
VarP(I)    Woolwt  1.00000 0.00000  1.0000 1.0000
```

Correlation corresponding to each var/covariance component partitioned by DME (OLS-b):

```
           Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Birwt:Birwt      1      0      1      1
VarG(Ia)   Birwt:Birwt      1      0      1      1
VarGs(Ia)  Birwt:Birwt      1      0      1      1
VarE(M)    Birwt:Birwt      1      0      1      1
VarG(Ma)   Birwt:Birwt      1      0      1      1
VarGs(Ma)  Birwt:Birwt      1      0      1      1
VarP(I)    Birwt:Birwt      1      0      1      1
```

```
           Traitpair  Estimate StdErr  CI95lo  CI95hi
VarE(I)    Birwt:Diamtr -0.0955 0.0644 -0.2218  0.0308
VarG(Ia)   Birwt:Diamtr -0.3268 0.0290 -0.3836 -0.2700
VarGs(Ia)  Birwt:Diamtr  0.6062 0.3522 -0.0841  1.2966
VarE(M)    Birwt:Diamtr  0.4145 0.1851  0.0516  0.7773
VarG(Ma)   Birwt:Diamtr  0.7088 0.4994 -0.2701  1.6876
VarGs(Ma)  Birwt:Diamtr  0.8744 0.6524 -0.4042  2.1530
VarP(I)    Birwt:Diamtr  0.0346 0.0126  0.0099  0.0593
```

```
           Traitpair  Estimate StdErr  CI95lo  CI95hi
VarE(I)    Birwt:Woolwt  0.113  0.0539  0.00694  0.218
VarG(Ia)   Birwt:Woolwt  0.207  0.0528  0.10343  0.310
VarGs(Ia)  Birwt:Woolwt  0.124  0.2187 -0.30437  0.553
VarE(M)    Birwt:Woolwt  0.510  0.2282  0.06284  0.957
VarG(Ma)   Birwt:Woolwt  0.324  0.3921 -0.44458  1.092
VarGs(Ma)  Birwt:Woolwt  0.298  0.1115  0.07924  0.516
VarP(I)    Birwt:Woolwt  0.172  0.0126  0.14728  0.197
```

```
           Traitpair  Estimate StdErr  CI95lo  CI95hi
VarE(I)    Diamtr:Birwt -0.0955 0.0644 -0.2218  0.0308
VarG(Ia)   Diamtr:Birwt -0.3268 0.0290 -0.3836 -0.2700
VarGs(Ia)  Diamtr:Birwt  0.6062 0.3522 -0.0841  1.2966
VarE(M)    Diamtr:Birwt  0.4145 0.1851  0.0516  0.7773
VarG(Ma)   Diamtr:Birwt  0.7088 0.4994 -0.2701  1.6876
VarGs(Ma)  Diamtr:Birwt  0.8744 0.6524 -0.4042  2.1530
```

```
VarP(I)    Diamtr:Birwt   0.0346 0.0126   0.0099   0.0593


               Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Diamtr:Diamtr        1      0   1.00   1.00
VarG(Ia)   Diamtr:Diamtr        1      0   1.00   1.00
VarGs(Ia)  Diamtr:Diamtr        1      0   1.00   1.00
VarE(M)    Diamtr:Diamtr        1      0   1.00   1.00
VarG(Ma)   Diamtr:Diamtr        1      1  -0.96   2.96
VarGs(Ma)  Diamtr:Diamtr        1      0   1.00   1.00
VarP(I)    Diamtr:Diamtr        1      0   1.00   1.00


               Traitpair Estimate StdErr   CI95lo CI95hi
VarE(I)    Diamtr:Woolwt     0.243 0.0552   0.1350   0.351
VarG(Ia)   Diamtr:Woolwt     0.160 0.0321   0.0973   0.223
VarGs(Ia)  Diamtr:Woolwt     0.792 0.2738   0.2553   1.329
VarE(M)    Diamtr:Woolwt     0.389 0.2456  -0.0924   0.870
VarG(Ma)   Diamtr:Woolwt    -0.317 8.3706 -16.7234 16.090
VarGs(Ma)  Diamtr:Woolwt     0.616 0.5470  -0.4565   1.688
VarP(I)    Diamtr:Woolwt     0.285 0.0133   0.2589   0.311


               Traitpair Estimate StdErr   CI95lo CI95hi
VarE(I)    Woolwt:Birwt      0.113 0.0539  0.00694   0.218
VarG(Ia)   Woolwt:Birwt      0.207 0.0528  0.10343   0.310
VarGs(Ia)  Woolwt:Birwt      0.124 0.2187 -0.30437   0.553
VarE(M)    Woolwt:Birwt      0.510 0.2282  0.06284   0.957
VarG(Ma)   Woolwt:Birwt      0.324 0.3921 -0.44458   1.092
VarGs(Ma)  Woolwt:Birwt      0.298 0.1115  0.07924   0.516
VarP(I)    Woolwt:Birwt      0.172 0.0126  0.14728   0.197


               Traitpair Estimate StdErr   CI95lo CI95hi
VarE(I)    Woolwt:Diamtr     0.243 0.0552   0.1350   0.351
VarG(Ia)   Woolwt:Diamtr     0.160 0.0321   0.0973   0.223
VarGs(Ia)  Woolwt:Diamtr     0.792 0.2738   0.2553   1.329
VarE(M)    Woolwt:Diamtr     0.389 0.2456  -0.0924   0.870
VarG(Ma)   Woolwt:Diamtr    -0.317 8.3706 -16.7234 16.090
VarGs(Ma)  Woolwt:Diamtr     0.616 0.5470  -0.4565   1.688
VarP(I)    Woolwt:Diamtr     0.285 0.0133   0.2589   0.311


               Traitpair Estimate StdErr CI95lo CI95hi
VarE(I)    Woolwt:Woolwt        1      0   1.00   1.00
VarG(Ia)   Woolwt:Woolwt        1      0   1.00   1.00
VarGs(Ia)  Woolwt:Woolwt        1      0   1.00   1.00
VarE(M)    Woolwt:Woolwt        1      0   1.00   1.00
VarG(Ma)   Woolwt:Woolwt        1      1  -0.96   2.96
VarGs(Ma)  Woolwt:Woolwt        1      0   1.00   1.00
VarP(I)    Woolwt:Woolwt        1      0   1.00   1.00
```

```
Phenotypic var/covariance from components partitioned by DME (OLS-b):

      Traitpair Estimate  StdErr CI95lo CI95hi
1   Birwt:Birwt   0.4049 0.00478 0.3955 0.4142
2  Birwt:Diamtr   0.0528 0.01918 0.0152 0.0904
3  Birwt:Woolwt   0.0519 0.00391 0.0443 0.0596
4  Diamtr:Birwt   0.0528 0.01918 0.0152 0.0904
5 Diamtr:Diamtr   5.7428 0.07685 5.5922 5.8935
6 Diamtr:Woolwt   0.3242 0.01567 0.2935 0.3550
7  Woolwt:Birwt   0.0519 0.00391 0.0443 0.0596
8 Woolwt:Diamtr   0.3242 0.01567 0.2935 0.3550
9 Woolwt:Woolwt   0.2254 0.00319 0.2191 0.2317

>
```

On initial inspection this model seems to be a reasonable fit. The standard errors of estimates of proportions are generally as small as those in previous models. There is evidence of "VarGs(Ma)" being a significant proportion of the variance for Birwt, Weanwt, Flcwt, and Woolwt. For these traits the other maternal components are reduced, indicating that "VarGs(Ma)" has taken variance away from the autosomal genetic and environmental maternal components. In the case of Birwt, the individual sexlinked component is also reduced, but not for other traits.

There are more collinearities in this model than in previous models. The column correlations have two exceeding 0.8 and two more exceeding 0.7. We need to rerun with *dmeopt="pcr"* and see what effect removing the collinearities has on the estimates. Again we have to revert to one trait

```
> merinokeep.mdf <- mdf(merino.df,pedcols=c(1:3),factorcols=c(4:9,12:13),
    ycols=c(10:11,14:22),sexcode=c("M","F"),
    relmat=c("E","A","S.hopi"),keep=T)
  .....


> merino.fitiamaslpcr <- dmm(merinokeep.mdf, Woolwt ~ 1 + C(Sex, sum) +
    C(Yearbi, sum) + C(YearDbi, sum) + C(Agem, sum) + C(Birls,sum) +
    C(Weanls, sum), components = c("VarE(I)", "VarG(Ia)","VarGs(Ia)",
    "VarE(M)","VarG(Ma)","VarGs(Ma)"), dmeopt="pcr")
Dyadic mixed model fit for datafile: merinokeep.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 6
OLS-b step:
no of fixed effect df (k) =  43
no of traits (l) =  1
```

```
Setup antemodel matrices:
no of individuals in pedigree (m) =  4014
no of individuals with data and X codes (n) =  2735
Rank of X: 43   No of Fixed Effects: 43
DME substep:
PCR option on dyadic model equations:
Data:  X dimension: 7480225 6
Y dimension: 7480225 1
Fit method: svdpc
Number of components considered: 6

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV          0.1611   0.1611   0.1611   0.1611   0.1611   0.1611   0.1611
adjCV       0.1611   0.1611   0.1611   0.1611   0.1611   0.1611   0.1611

TRAINING: % variance explained
       1 comps    2 comps    3 comps    4 comps    5 comps     6 comps
X     71.64997   85.71700   93.01582   96.65201   99.60010   100.00000
evec   0.01946    0.02729    0.03319    0.04001    0.04196     0.04297
DME substep completed:
OLS-b step completed:
>
> merino.fitiamaslpcr$pcr.loadings

Loadings:
           Comp 1 Comp 2 Comp 3 Comp 4 Comp 5 Comp 6
'VarE(I)'    0.112 -0.115  0.344 -0.528 -0.457  0.606
'VarG(Ia)'   0.324 -0.918         0.214
'VarGs(Ia)'  0.250         0.247 -0.604        -0.711
'VarE(M)'    0.388  0.217  0.604  0.548 -0.338 -0.151
'VarG(Ma)'   0.611  0.191 -0.660        -0.392
'VarGs(Ma)'  0.545  0.235  0.139 -0.100  0.719  0.320


               Comp 1 Comp 2 Comp 3 Comp 4 Comp 5 Comp 6
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.167  0.167  0.167  0.167  0.167  0.167
Cumulative Var  0.167  0.333  0.500  0.667  0.833  1.000
>
```

    This shows that we can omit principal components 5 and 6 and still analyze
96.65 percent of the variation. The constraints implied by dropping components
5 and 6 are shown by the respective columns of the loadings. If we rerun the
"pcr" analysis with only 4 principal components we get

```
> merino.fitiamaslpcr2 <- dmm(merinokeep.mdf, Woolwt ~ 1 + C(Sex, sum) +
```

```
        C(Yearbi, sum) + C(YearDbi, sum) + C(Agem, sum) + C(Birls,sum) +
        C(Weanls, sum), components = c("VarE(I)", "VarG(Ia)","VarGs(Ia)",
        "VarE(M)","VarG(Ma)","VarGs(Ma)"), dmeopt="pcr",ncomp=4)
Dyadic mixed model fit for datafile: merinokeep.mdf
 .....
OLS-b step completed:
>
> gsummary(merino.fitiamaslpcr2)
Call:
gsummary.dmm(dmmobj = merino.fitiamaslpcr2)

Proportion of phenotypic var/covariance partitioned by DME to each component (OLS-b):

            Trait Estimate  StdErr CI95lo CI95hi
VarE(I)    Woolwt 3.54e-01 0.00960 0.3354  0.373
VarG(Ia)   Woolwt 1.63e-01 0.01198 0.1391  0.186
VarGs(Ia)  Woolwt 3.72e-01 0.01045 0.3512  0.392
VarE(M)    Woolwt 5.80e-09     Inf   -Inf    Inf
VarG(Ma)   Woolwt 5.80e-09     Inf   -Inf    Inf
VarGs(Ma)  Woolwt 1.12e-01 0.00884 0.0942  0.129
VarP(I)    Woolwt 1.00e+00 0.00000 1.0000  1.000


 .....
Phenotypic var/covariance from components partitioned by DME (OLS-b):

      Traitpair Estimate  StdErr CI95lo CI95hi
1 Woolwt:Woolwt    0.172 0.00435  0.164  0.181


>
```

So we have some nasty standard errors and near-zero estimates for "VarE(M)" and "VarG(Ma)" suggesting that they should not be fitted. However the estimates for the other components are similar to those obtained with *dmeopt="qr"* and their standard errors are small. We conclude that the components "VarGs(Ia)" and "VarGs(Ma)" are real and of sufficient magnitude to warrant inclusion in the model. If the constrained estimate obtained with "pcr" do not differ greatly from the unconstrained estimates obtained with "qr" it is an indication that the collinearities are not serious. If we were to drop "VarE(M)" and "VarG(Ma)" some of the strongest column correlations would be removed. We will pursue this course for the final run with *gls=T* in the following section.

9.7   Final analysis and GLS-b estimates

It is convenient to use OLS-b estimates when investigating models, then finish off with an iterative GLS-b run to obtain bias-corrected ML estimates. We shall now do this omitting "VarE(M)" and "VarG(Ma)" as noted above.

```
> merino.fitgls <- dmm(merino.mdf, Ymat ~ 1 + C(Sex, sum) + C(Yearbi, sum) +
    C(YearDbi, sum) + C(Agem, sum) + C(Birls,sum) + C(Weanls, sum),
    components = c("VarE(I)", "VarG(Ia)","VarGs(Ia)","VarGs(Ma)"),gls=T)
Dyadic mixed model fit for datafile: merino.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 4
OLS-b step:
no of fixed effect df (k) =  43
no of traits (l) =  11
Setup antemodel matrices:
no of individuals in pedigree (m) =  4014
no of individuals with data and X codes (n) =  2642
Rank of X: 43   No of Fixed Effects: 43
DME substep:
QR option on dyadic model equations:
DME substep completed:
OLS-b step completed:

GLS-b step:
Warning: Multivariate GLS is not same as multiple univariate GLS's
Error in matrix(0, am$n * am$n * am$l * am$l, am$v) :
  too many elements specified
>
```

So we cannot complete the GLS-b step with 11 traits - One of the arrays used
exceeds the R limit of $2^{31} - 1$ elements. So lets do it with one trait

```
> merino.fitgls <- dmm(merinokeep.mdf, Woolwt ~ 1 + C(Sex, sum) + C(Yearbi, sum) +
    C(YearDbi, sum) + C(Agem, sum) + C(Birls,sum) + C(Weanls, sum),
    components = c("VarE(I)", "VarG(Ia)","VarGs(Ia)","VarGs(Ma)"),gls=T)
Dyadic mixed model fit for datafile: merinokeep.mdf
Data file is a list containing a dataframe and a list of relationship matrices:
Random effect partitioned into components: Residual:
No of individual variance components partitioned(v): 4
OLS-b step:
 .....
OLS-b step completed:

GLS-b step:
Round =  1  Stopcrit =  0.01250322
Round =  2  Stopcrit =  0.002486147
Iteration completed - count =  2
Convergence achieved
GLS-b step completed successfully:
```

```
>
> gsummary(merino.fitgls,gls=T)
Call:
gsummary.dmm(dmmobj = merino.fitgls, gls = T)

Proportion of phenotypic var/covariance partitioned by DME to each component (OLS-b):

           Trait Estimate StdErr    CI95lo CI95hi
VarE(I)    Woolwt   0.7485 0.0283  0.692933 0.8040
VarG(Ia)   Woolwt   0.1561 0.0102  0.136055 0.1761
VarGs(Ia)  Woolwt   0.0675 0.0399 -0.010722 0.1458
VarGs(Ma)  Woolwt   0.0279 0.0145 -0.000569 0.0564
VarP(I)    Woolwt   1.0000 0.0000  1.000000 1.0000


 .....


Proportion of phenotypic var/covariance partitioned by DME to each component (GLS-b):

           Trait Estimate  StdErr    CI95lo CI95hi
VarE(I)    Woolwt   0.7429 0.02831  0.687363 0.7983
VarG(Ia)   Woolwt   0.1546 0.00999  0.135021 0.1742
VarGs(Ia)  Woolwt   0.0778 0.03987 -0.000332 0.1560
VarGs(Ma)  Woolwt   0.0247 0.01450 -0.003692 0.0532
VarP(I)    Woolwt   1.0000 0.00000  1.000000 1.0000


 .....


Phenotypic var/covariance from components partitioned by DME (GLS-b):

       Traitpair Estimate  StdErr CI95lo CI95hi
1 Woolwt:Woolwt    0.166 0.00311   0.16  0.172


>
```

The OLS-b and GLS-b estimates are almost identical for this trait. GLS-b
offers a slight improvement in the standard errors. In my experience this is what
happens whenever the model is reasonable for the dataset and the dataset is
sufficiently large. What causes OLS-b and GLS-b to differ is correlations among
the residuals, and that only happens when the model is inappropriate.

We conclude that the above 4 component model is appropriate and proceed
with some further analysis of genetic (co)variation among wool traits.

9.8 Investigation of genetic (co)variation among Merino wool traits

Wool is a more complicated product than most production traits to which quantitative genetics is applied. It takes a number of measured traits to describe the quantity and quality of the wool obtained from a sheep, and a further number of traits to understand its relationship to the sheep as a whole.

Now that we have an appropriate model for estimating genetic (co)variance components for wool traits, we can take a closer look at the genetic corelations and what they might imply about the structure of the wool genome.

We start with a final run of the 'best' model, and them embark on a study of the structure of each genetic (co)variance matrix (VarG(Ia), VarGs(Ia), and VarGs(Ma)).

```
> merino.fitlast <- dmm(merino.mdf, Ymat ~ 1 + C(Sex, sum) + C(Yearbi, sum) +
    C(YearDbi, sum) + C(Agem, sum) + C(Birls,sum) + C(Weanls, sum),
    components = c("VarE(I)", "VarG(Ia)","VarGs(Ia)","VarGs(Ma)"))
 .....
>
```

One of the difficulties of principal components analysis is that its results are seriously biased if all traits are not measured in the same units. The traditional way of dealing with this is to do everything in standard deviation units, that is to use a correlation matrix uinstead of a covariance matrix. In our case this would amount to scaling to genetic standard deviation units. However we are not going to do that, we are going to scale to phenotypic standard deviation units which is the traditional approach of quantitative genetics to scaling. To do this we need to do a principal component analysis on the matrix shown below for the two trait case

$$H_{G(Ia)} = \begin{bmatrix} h_1^2 & h_1 h_2 r_{G(Ia)} \\ h_1 h_2 r_{G(Ia)} & h_2^2 \end{bmatrix}$$

and similar matrices for $Gs(Ia)$ and $Gs(Ma)$.

One reason for this approach is to avoid giving excess weight to traits for which the proportion of variance is small. By putting heritabilities (or proportion of variance) on the diagonal we are weighting each trait by its heritability. That is what is required if we wish to compare genetic variation of various traits.

The H matrices are not the same as the matrix GP^{-1} from the multivariate breeders equation 14. That matrix is in trait units and is not symmetric and is for prediction of genetic change. Here we are attempting to study genetic variation itself, not prediction.

The required H matrices are readily calculated from variance component estimates as follows. First extract the required covariance matrices

```
> covgia <- matrix(merino.fitlast$siga["VarG(Ia)",],11,11,
  dimnames=list(dimnames(merino.fitlast$b)[[2]],dimnames(merino.fitlast$b)[[2]]))
```

```
> covgsia <- matrix(merino.fitlast$siga["VarGs(Ia)",],11,11,
   dimnames=list(dimnames(merino.fitlast$b)[[2]],dimnames(merino.fitlast$b)[[2]]))
> covgsma <- matrix(merino.fitlast$siga["VarGs(Ma)",],11,11,
   dimnames=list(dimnames(merino.fitlast$b)[[2]],dimnames(merino.fitlast$b)[[2]]))
> covpia <- merino.fitlast$phenotypic.variance
```

Then do the $H_{G(Ia)}$ matrix and its principal component analysis

```
> pdiag <- diag(1/sqrt(diag(covpia)))
> covhia <- pdiag %*% covgia %*% pdiag
> dimnames(covhia) <- dimnames(covgia)
> covhia
              Birwt       Weanwt        Crimp       Densty      Diamtr
Birwt    0.185250434  0.03929988 -0.03353903  0.099824593 -0.09390297
Weanwt   0.039299878  0.10941694 -0.04923602 -0.088406290  0.07556103
Crimp   -0.033539029 -0.04923602  0.37070449  0.068197321 -0.08545129
Densty   0.099824593 -0.08840629  0.06819732  0.492360864 -0.36754001
Diamtr  -0.093902972  0.07556103 -0.08545129 -0.367540007  0.43504635
Yield    0.076601452  0.07578843 -0.19534958  0.008542196  0.01562528
Bodywt   0.033581775  0.09258608 -0.03186814 -0.101092848  0.07110460
Wrinkl  -0.059993649 -0.08225562  0.11941112  0.022897920  0.12636478
Length   0.003896161  0.04859919 -0.12248895 -0.145037715  0.02423141
Flcwt    0.012666618  0.01374432 -0.00896495  0.046896192  0.02653705
Woolwt   0.031968963  0.04022985 -0.08175608  0.035292205  0.03720690
               Yield        Bodywt       Wrinkl       Length        Flcwt
Birwt    0.0766014519  0.0335817755 -0.059993649  0.003896161  0.01266662
Weanwt   0.0757884344  0.0925860805 -0.082255622  0.048599189  0.01374432
Crimp   -0.1953495806 -0.0318681375  0.119411117 -0.122488953 -0.00896495
Densty   0.0085421965 -0.1010928476  0.022897920 -0.145037715  0.04689619
Diamtr   0.0156252782  0.0711045965  0.126364776  0.024231412  0.02653705
Yield    0.3373245271  0.0007260227 -0.157966278  0.166419417  0.03442103
Bodywt   0.0007260227  0.1368491936 -0.044559996  0.007385097 -0.01056552
Wrinkl  -0.1579662783 -0.0445599956  0.399766067 -0.223431622  0.05312845
Length   0.1664194166  0.0073850966 -0.223431622  0.229604913 -0.01642479
Flcwt    0.0344210271 -0.0105655207  0.053128447 -0.016424791  0.07006928
Woolwt   0.1580510425 -0.0118929591 -0.005243948  0.046841331  0.07292928
             Woolwt
Birwt    0.031968963
Weanwt   0.040229846
Crimp   -0.081756079
Densty   0.035292205
Diamtr   0.037206896
Yield    0.158051042
Bodywt  -0.011892959
Wrinkl  -0.005243948
Length   0.046841331
```

```
Flcwt    0.072929279
Woolwt   0.123820403
>
> covhia.list <- list(cov=covhia,center=rep(0,11),n.obs=2599)
> prinhia <- princomp(covmat=covhia.list)
> summary(prinhia)
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
Standard deviation     0.9957620 0.9192753 0.6441519 0.48108792 0.44317179
Proportion of Variance 0.3430688 0.2923892 0.1435643 0.08007906 0.06795389
Cumulative Proportion  0.3430688 0.6354580 0.7790223 0.85910138 0.92705527
                           Comp.6    Comp.7     Comp.8      Comp.9      Comp.10
Standard deviation     0.30851264 0.2368932 0.20775519 0.116454014 0.0529312089
Proportion of Variance 0.03293184 0.0194167 0.01493392 0.004692227 0.0009693792
Cumulative Proportion  0.95998711 0.9794038 0.99433773 0.999029954 0.9999993333
                          Comp.11
Standard deviation     1.388134e-03
Proportion of Variance 6.667039e-07
Cumulative Proportion  1.000000e+00
>
> loadings(prinhia)

Loadings:
       Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
Birwt         -0.243        -0.572         0.721 -0.233  0.144
Weanwt  0.203                -0.460        -0.326                0.654 -0.452
Crimp  -0.372  0.256  0.335 -0.226  0.778
Densty -0.555 -0.405 -0.290               -0.290         0.267 -0.291 -0.433
Diamtr  0.450  0.470 -0.265                              0.577 -0.310 -0.227
Yield   0.308 -0.412 -0.359         0.359         0.599
Bodywt  0.152         0.111 -0.575 -0.201 -0.314  0.230 -0.385 -0.530
Wrinkl -0.249  0.486 -0.538                0.261  0.199 -0.468        -0.284
Length  0.341 -0.241  0.229  0.261  0.256  0.194 -0.239 -0.378 -0.274 -0.559
Flcwt                -0.289         0.196 -0.218 -0.602 -0.178 -0.109  0.159
Woolwt  0.113 -0.149 -0.397         0.316 -0.164 -0.238 -0.151         0.350
       Comp.11
Birwt
Weanwt
Crimp
Densty -0.108
Diamtr
Yield   0.313
Bodywt
Wrinkl
Length -0.117
Flcwt   0.627
```

```
Woolwt -0.686

              Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
SS loadings    1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var 0.091  0.091  0.091  0.091  0.091  0.091  0.091  0.091  0.091
Cumulative Var 0.091  0.182  0.273  0.364  0.455  0.545  0.636  0.727  0.818
              Comp.10 Comp.11
SS loadings     1.000   1.000
Proportion Var  0.091   0.091
Cumulative Var  0.909   1.000
>
```

Then construct the $\boldsymbol{H_{Gs(Ia)}}$ matrix and do its principal component analysis

```
> covhsia <- pdiag %*% covgsia %*% pdiag
> dimnames(covhsia) <- dimnames(covgsia)
> covhsia
               Birwt         Weanwt        Crimp         Densty        Diamtr
Birwt    0.060128044 -2.570243e-02 -0.060687741  0.008348901  0.05209222
Weanwt  -0.025702433  4.315879e-02  0.061761044 -0.003353315 -0.02661258
Crimp   -0.060687741  6.176104e-02  0.159211280 -0.014082210 -0.09986772
Densty   0.008348901 -3.353315e-03 -0.014082210  0.001774278  0.01177068
Diamtr   0.052092218 -2.661258e-02 -0.099867718  0.011770679  0.09047465
Yield    0.047431553 -2.191617e-02 -0.056677638  0.007172361  0.03608505
Bodywt   0.016112986  5.055761e-02  0.035293668  0.003982618  0.01385322
Wrinkl  -0.004737217 -2.861619e-02  0.008036627 -0.004155290 -0.01149093
Length   0.002021006  1.725220e-04 -0.095304872  0.010574906  0.10112966
Flcwt    0.022473403  9.346233e-05 -0.087593073  0.011411457  0.11699630
Woolwt   0.048915663 -1.786171e-02 -0.138921426  0.016499676  0.11833350
               Yield         Bodywt        Wrinkl        Length        Flcwt
Birwt    0.047431553  0.016112986 -0.0047372169  0.0020210061  2.247340e-02
Weanwt  -0.021916172  0.050557606 -0.0286161921  0.0001725220  9.346233e-05
Crimp   -0.056677638  0.035293668  0.0080366270 -0.0953048716 -8.759307e-02
Densty   0.007172361  0.003982618 -0.0041552896  0.0105749065  1.141146e-02
Diamtr   0.036085050  0.013853220 -0.0114909298  0.1011296643  1.169963e-01
Yield    0.047598384  0.013098530 -0.0214068355 -0.0134504843 -1.201379e-02
Bodywt   0.013098530  0.111362916 -0.0615829646  0.0224709056  3.270802e-02
Wrinkl  -0.021406835 -0.061582965  0.1005839556  0.0001208749  3.500162e-03
Length  -0.013450484  0.022470906  0.0001208749  0.2469677111  2.356635e-01
Flcwt   -0.012013793  0.032708021  0.0035001621  0.2356634618  2.692642e-01
Woolwt   0.039745698  0.040076273 -0.0315794032  0.1957919828  1.822150e-01
            Woolwt
Birwt    0.04891566
Weanwt  -0.01786171
Crimp   -0.13892143
Densty   0.01649968
```

141

```
Diamtr  0.11833350
Yield   0.03974570
Bodywt  0.04007627
Wrinkl -0.03157940
Length  0.19579198
Flcwt   0.18221497
Woolwt  0.21701532
>
> covhsia.list <- list(cov=covhsia, center=rep(0,11), n.obs=2599)
> prinhsia <- princomp(covmat=covhsia.list)
> summary(prinhsia)
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
Standard deviation     0.8942314 0.4953254 0.4440996 0.25763953 0.18687404
Proportion of Variance 0.5934147 0.1820705 0.1463589 0.04925876 0.02591531
Cumulative Proportion  0.5934147 0.7754853 0.9218442 0.97110296 0.99701827
                           Comp.6       Comp.7       Comp.8       Comp.9
Standard deviation     0.063386995 2.394215e-04 1.554766e-04 7.119635e-05
Proportion of Variance 0.002981665 4.253877e-08 1.793860e-08 3.761612e-09
Cumulative Proportion  0.999999934 1.000000e+00 1.000000e+00 1.000000e+00
                          Comp.10      Comp.11
Standard deviation     4.762722e-05 1.586003e-05
Proportion of Variance 1.683329e-09 1.866666e-10
Cumulative Proportion  1.000000e+00 1.000000e+00
>
> loadings(prinhsia)

Loadings:
       Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
Birwt   0.105  0.319 -0.221 -0.501        -0.129  0.751
Weanwt        -0.372 -0.192                       0.106         0.659 -0.610
Crimp  -0.320 -0.544        -0.260                      -0.460 -0.512 -0.205
Densty
Diamtr  0.302  0.192        -0.228  0.251  0.506 -0.289 -0.613  0.149
Yield          0.334 -0.285 -0.127 -0.109        -0.249  0.362 -0.436 -0.616
Bodywt        -0.334 -0.598 -0.339 -0.257  0.205 -0.201  0.257         0.437
Wrinkl                0.592 -0.542 -0.516        -0.184         0.191
Length  0.512 -0.296  0.211  0.243 -0.243  0.512  0.396  0.133 -0.213
Flcwt   0.525 -0.319  0.191 -0.325  0.516 -0.358 -0.161  0.233
Woolwt  0.495        -0.184  0.196 -0.506 -0.524 -0.124 -0.361
       Comp.11
Birwt
Weanwt
Crimp
Densty  0.996
Diamtr
```

```
Yield
Bodywt
Wrinkl
Length
Flcwt
Woolwt


              Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
SS loadings    1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var 0.091  0.091  0.091  0.091  0.091  0.091  0.091  0.091  0.091
Cumulative Var 0.091  0.182  0.273  0.364  0.455  0.545  0.636  0.727  0.818
              Comp.10 Comp.11
SS loadings     1.000   1.000
Proportion Var  0.091   0.091
Cumulative Var  0.909   1.000
>
```

Then construct the $H_{Gs(Ma)}$ matrix and do its principal component analysis

```
> covhsma <- pdiag %*% covgsma %*% pdiag
> dimnames(covhsma) <- dimnames(covgsma)
> covhsma
               Birwt        Weanwt         Crimp         Densty         Diamtr
Birwt    0.267363026  0.137241387  0.0340141735  0.0224145197 -0.0027497253
Weanwt   0.137241387  0.206176335 -0.0025871132  0.0269415586 -0.0070518859
Crimp    0.034014174 -0.002587113  0.0366677689  0.0007726814 -0.0052437348
Densty   0.022414520  0.026941559  0.0007726814  0.0046208666  0.0007816775
Diamtr  -0.002749725 -0.007051886 -0.0052437348  0.0007816775  0.0071639766
Yield   -0.050803949 -0.026813088 -0.0219610494 -0.0094145295 -0.0043997441
Bodywt   0.073497448  0.099223370  0.0104932768  0.0117539224 -0.0102842487
Wrinkl   0.049807133  0.039388864  0.0143645776  0.0057306131 -0.0013607385
Length  -0.011232973 -0.012591844 -0.0191543654 -0.0045433551  0.0030477222
Flcwt    0.031466941  0.060447774  0.0166763649  0.0026909844 -0.0087321184
Woolwt  -0.006234029  0.029886280  0.0171577640 -0.0027997247 -0.0108419599
               Yield        Bodywt        Wrinkl        Length          Flcwt
Birwt   -0.050803949  0.07349745  0.049807133 -0.011232973  0.031466941
Weanwt  -0.026813088  0.09922337  0.039388864 -0.012591844  0.060447774
Crimp   -0.021961049  0.01049328  0.014364578 -0.019154365  0.016676365
Densty  -0.009414529  0.01175392  0.005730613 -0.004543355  0.002690984
Diamtr  -0.004399744 -0.01028425 -0.001360738  0.003047722 -0.008732118
Yield    0.043100910 -0.01277511 -0.014046311  0.026590788  0.010112202
Bodywt  -0.012775111  0.05591314  0.022014273 -0.011791399  0.038009444
Wrinkl  -0.014046311  0.02201427  0.015504705 -0.006105715  0.020147123
Length   0.026590788 -0.01179140 -0.006105715  0.026340875  0.002615934
Flcwt    0.010112202  0.03800944  0.020147123  0.002615934  0.075838894
Woolwt   0.020990923  0.02575638  0.013663099  0.006666227  0.071446316
```

```
              Woolwt
Birwt   -0.006234029
Weanwt   0.029886280
Crimp    0.017157764
Densty  -0.002799725
Diamtr  -0.010841960
Yield    0.020990923
Bodywt   0.025756376
Wrinkl   0.013663099
Length   0.006666227
Flcwt    0.071446316
Woolwt   0.076916059
>
>
> covhsma.list <- list(cov=covhsma, center=rep(0,11), n.obs=2599)
> prinhsma <- princomp(covmat=covhsma.list)
> summary(prinhsma)
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
Standard deviation     0.6794515 0.4198404 0.3065772 0.25493113 0.12124967
Proportion of Variance 0.5660258 0.2161165 0.1152388 0.07968288 0.01802521
Cumulative Proportion  0.5660258 0.7821423 0.8973811 0.97706403 0.99508924
                           Comp.6       Comp.7       Comp.8       Comp.9
Standard deviation     0.063285913 3.249387e-04 1.407503e-04 9.169898e-05
Proportion of Variance 0.004910587 1.294560e-07 2.428947e-08 1.030975e-08
Cumulative Proportion  0.999999831 1.000000e+00 1.000000e+00 1.000000e+00
                          Comp.10      Comp.11
Standard deviation     6.037734e-05 2.042471e-05
Proportion of Variance 4.469585e-09 5.114829e-10
Cumulative Proportion  1.000000e+00 1.000000e+00
>
> loadings(prinhsma)

Loadings:
       Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
Birwt  -0.667 -0.470  0.371 -0.390         0.139  0.148
Weanwt -0.588  0.240 -0.612  0.105  0.114 -0.121                0.191  0.354
Crimp                 0.477  0.412 -0.138 -0.389 -0.301 -0.308  0.439  0.208
Densty                              0.122
Diamtr                              0.602         0.105                0.597 -0.508
Yield   0.133  0.270        -0.544 -0.424        -0.113  0.292  0.540  0.114
Bodywt -0.310  0.167 -0.135  0.147 -0.490 -0.105 -0.129               -0.745
Wrinkl -0.163         0.119         0.196 -0.436 -0.241  0.797 -0.167
Length                      -0.577  0.221 -0.540 -0.277 -0.409 -0.260
Flcwt  -0.204  0.513  0.290         0.281  0.528 -0.490        -0.101
Woolwt         0.585  0.360               -0.206  0.685
```

144

```
        Comp.11
Birwt
Weanwt -0.116
Crimp
Densty  0.976
Diamtr
Yield   0.169
Bodywt
Wrinkl
Length
Flcwt
Woolwt


                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
SS loadings      1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var   0.091  0.091  0.091  0.091  0.091  0.091  0.091  0.091  0.091
Cumulative Var   0.091  0.182  0.273  0.364  0.455  0.545  0.636  0.727  0.818
                Comp.10 Comp.11
SS loadings       1.000   1.000
Proportion Var    0.091   0.091
Cumulative Var    0.909   1.000
>
```

Now we are ready to interpret these analyses. First some general observations. The first principal component of an H matrix defines the direction in 11-dimensional space in which it is easiest to achieve genetic change by selection. The last (11th in this case) principal component defines the direction in which it is most difficult to achieve genetic change by selection. All of the significant components, that is those which explain 5 percent or more of the variance, define directions in which it is possible to achieve genetic change by selection. The remaining non-significant components define directions in which it is very difficult or impossible to achieve genetic change by selection. Thus the insignificant components are also of interest. In the present case the structure of the genetic correlation matrices constrains the potential for genetic change quite dramatically. In all three cases (G(Ia), Gs(Ia), and Gs(Ma)) there are only 4 to 6 significant components - we have 11 traits, but genetic variation is confined to a 4 to 6 dimensional subspace.

A word of caution. This does not mean that we can not change individual traits. There has been ample demonstration of that in single-trait selection experiments (Turner, Brooker, and Dolling(1970) [27]). Nor does it mean that the 4 to 6 principal components are the only possible directions of change, there can be change in directions indicated by any mixture of the significant components. What it means is that we can not change certain combinations of traits. Multi-trait selection is known to be more difficult than single-trait selection. Adverse genetic correlations may be due to the sheer physical impossibility of some trait

combinations or they may be due to more subtle biological limitations, or they may be simply a genetic phenomenon such as pleiotropy or linkage.

Another word of caution. The rrelationship between traits may be nonlinear. For exampls

$$Woolwt = Flcwt \times Yield$$

For this reason a precise interpretation of the numerical values of the loading coefficients for any principal component is not advisable. Each component is a linear function of the traits, but it may be approximating a set of relationships which are not linear.

Now let us look specifically at the individual autosomal additive genetic variation , that is the principal components of matrix *covhia*. There are 6 significant principal components but the first 4 are most notable. We attempt to interpret their loadings as follows

Comp.1 Large sheep with coarse, long fibres and low wrinkle. And the opposite.

Comp.2 Coarse fibred sheep with short fibres and high wrinkle. And the opposite

Comp.3 Fine fibred sheep with low yield (lots of grease), low Woolwt, low wrinkle and high crimp frequency. And the opposite.

Comp.4 Small sheep with poor growth. And the opposite.

Among the insignificant components we might note that Comp.11 defines the impossibility of changing Flcwt and Woolwt in opposite directions, and also that Comp.9 and Comp.10 define the virtial impossibility of changing Densty and Diamtr in opposite directions. The latter case has been extensively studied and it is understood that Densty and Diamtr have a common cause because they are the outcome of a development system in which the number of follicle papilla cells determines both the number and size of follicles and fibres (Moore and Jackson (1984) [20], Moore, Jackson, and Lax (1989) [21]).

None of the above principal components put much emphasis on Woolwt - for the simple reason that it has a lower individual additive heritability than most other traits.

We now look at individual sexlinked additive genetic variation. There are only 3 significant principal components of matrix *covhsia*. We interpret their loadings as follows

Comp.1 High Woolwt and long coarse fibres. And the opposite.

Comp.2 Large at birth but do not grow, low crimp and short coarse fibres. And the opposite.

Comp.3 Small sheep with low growth and not wrinkled. And the opposite.

Comp.3 is similar to Comp.4 of the individual autosomal additive genetic variance.

We also look at maternal sexlinked additive genetic variation. There are 4 significant principal components of matrix *covhsma*. We interpret their loadings as follows

Comp.1 Small sheep with low growth and not wrinkled. And the opposite.

Comp.2 Small at birth but grow large and high Woolwt. And the opposite.

Comp.3 Small at birth but grow. High crimp and Woolwt. And the opposite.

Comp.4 Low yield, short fibres, high crimp. And the opposite

Again Comp.1 is similar to Comp.3 of *covhsia* and to Comp.4 of *covhia*.

The most striking thing across all three analyses is the close tie between body size and growth, on one hand, and wool length growth rate, Crimp and Woolwt, on the other. The only suggestion of independence is around Densty and Diamtr and only for individual additive variation.

It is tempting to interpret the independence of these principal components, as suggesting that there are independent sets of genes affecting variation in each orthogonal direction. I have never seen a proof that such a conclusion can be drawn. Certainly the G(Ia), Gs(Ia), and Gs(Ma) effects are independent - they are defined that way. But the principal components within each of these may or may not indicate separate gene effects or separate parts of the genome.

There remains the problem of response to selection. If we select for Woolwt, which of the above components change, and what does that mean? If we select for the commercially desirable combination of high Woolwt and low Diamtr, which components change? If we favour large sheep, what are the consequences for wool traits? We have to leave these questions until *gresponse()* function has been extended to deal with sexlinked variation.

9.9 Conclusions

We have shown that genetic variation in Woolwt comes from three sources, individual autosomal additive gene effects, individual sexlinked additive gene effects, an maternal sexlinked additive gene effects. The heritabilities of Woolwt from these three sources sum to the heritability of 0.40 found in published work. This is explained by published estimates being largely based on dam-daughter analyses which would include sexlinked genetic variation.

The genetic correlations among the 11 traits studied severely constrain the directions in which genetic improvement could be achieved, to 4 to 6 dimensions in the 11 dimensional trait space. One dimension, which was identified as a sort of small,low growth to large, high growth axis, was affected by all 3 types of gene effects, individual autosomal additive, individual sexlinked additive, and maternal sexlinked additive. Another dimension, associated with

changing Densty and Diamtr in opposite directions, was only affected by individual autosomal additive gene effects. The ways in which correlations constrain genetic improvement are given by the nonsignificant dimensions of variation.

There is some concern that we should have looked at whether the genetic parameters estimated here are sex-specific. That is, are the genetic parameters different for rams and ewes and are the cross-sex genetic corelations different from unity? There is work ongoing to extend *dmm()* to allow sex-specific analyses.

There have been some problems, but we have done enough to show that *dmm()* can be useful for quantitative genetic analysis of a sizeable multi-trait research dataset with a multi-generation pedigree and some significant fixed effects.

10   Wish list

Further development of *dmm()* and associated functions is likely. Some areas requiring attention are

- Using an error term other than 'Residual' from the monadic model to form the dyadic model equations and consequent variance component estimates. This would allow repeated measures models

- Extending the *gresponse()* function to deal with individual sexlinked and maternal sexlinked genetic variation. An initial (somewhat experimental) attempt at this has been made in release (dmm_1.6-2)

- Exploration of other regression techniques for solving the dyadic model equations. On my list are bootstrap methods and total least squares. The latter would allow for errors in the pedigree information as well as in the traits observed

- Allowing variance components to be split among levels of a fixed effect. For example with Sex as a fixed effect it is possible to form male-male, female-female, female-male, and male-female dyads and do their dyadic model equations separately thus leading to sex-specific variance components and genetic parameters. An initial attampt at this has been made in release (dmm_2.1-1).

- Do something about the very restrictive array size requirements with options *gls=T* and *dmeopt="pcr"*

- There has been no reference to the excellent R package *pedigreemm* which implements an entirely different approach to analysis of mixed model pedigree data. Some comparison of methods and results would be desirable.

- Look at sparse matrix techniques, possibly with the *Matrix* package

Suggestions and criticisms are welcome. The work started out of a need to analyse certain types of sheep breeding research data, and grew into an exploration of the feasibility of directly solving the dyadic model equations. With this narrow base, it could benefit from some input form other areas.

References

[1] Anderson, R.D. (1984) Use of mixed models for prediction and for estimation of (co)variance components. BLUP School Handbook, Animal Genetics and Breeding Unit, University of New England, NSW, Aust, Feb 5-7 1984.

[2] Anon (2014) UCLA: Statistical Computing Group. From http://www.ats.ucla.edu/stat/faq/deltamethod.htm (accessed 8 Dec 2014)

[3] Bijma, P. (2006) Estimating maternal genetic effects in livestock. J. Anim. Sci. 84:800-806 (http://www.journalofanimalscience.org/content/84/4/800)

[4] Brown, G.D. and Turner, H.N. (1968). Response to selection in Australian Merino sheep. II. Estimates of phenotypic and genetic parameters for some production traits in Merino ewes and an analysis of the possible effects of selection on them. Aust. J. Agric. Res. 19: 303-22.

[5] Dickerson, G.E. (1947) Composition of hog carcasses as influenced by heritable differences in rate and economy of gain. Iowa Agricultural Research Station Bulletin No. 354. pages 489-524.

[6] Fairbairn, D.J. and Roff, D.A. (2006) The quantitative genetics of sexual dimorphism: assessing the importance of sex-linkage. Heredity 97:319-328

[7] Falconer, D.S. (1961) Introduction to Quantitative Genetics. Oliver and Boyd, Edinburgh and London, 1961.

[8] Fraser, A.S. and Short, B.F. (1960). The Biology of the Fleece. CSIRO Animal Research Labs. Technical Paper No. 3.

[9] Gilmour, A., Thompson, R. and Cullis. B.R.(1995) AI an efficient algorithm for REML estimation in mixed models. Biometrics 51:1440-1450.

[10] Griffing, B. (1966) Influence of Sex on Selection. III Joint contribution of sex-linked and autosomal genes. Aust. J. Biol. Sci. 19: 775-93

[11] Harvey, W.R. (1960) Least Squares Analysid of Data with Unequal Subclass Numbers. ARS-20-8 July 1960. Agricultural Research Service, USDA.

[12] Jackson, N. (2015) Solving the Dyadic Model Equations. From https://github.com/nevillejackson/dmm

[13] Jackson, N. (2017) Estimating class-specific genetic parameters with $dmm()$ From https://github.com/nevillejackson/dmm

[14] Jouini, W. Le Guennec , D. and Moy, C (2011) Log-Normal approximation of Chi-square distributions for signal processing. URSIGASS.2011.6050531 Conference: General Assembly and Scientific Symposium, 2011 XXXth URSI . From http://www.researchgate.net/publication/224263352_Log-normal_ approximation_of_chi-square_distributions_for_signal_processing

[15] Kempthorne, O. (1957) An Introduction to Genetic Statistics. John Wiley and Sons, New York, 1963.

[16] Koch, G.G. (1967) A general approach to the estimation of variance compnents. Technometrics 9:93-118

[17] Lynch, M. and Walsh, B. (1998) Genetics and Analysis of Quantitative Traits. Sinauer, Sunderland, MA.

[18] Mevik, Bjorne-Helge and Wehrens, Ron (2007) The pls Package: Principal Component and Partial Least Squares Regression in R. Journal of Statistical Software 18(2):1-24.

[19] Meyer, K. (1998) DFREML Version 3.0 β User Notes. Document distributed with the DFREML program.

[20] Moore, G.P.M. and Jackson, N. (1984). An hypothesis implicating a founder cell population in the regulation of wool follicle formation and distribution in sheep skin. J. Embryol. Exp. Morph. 82 (Suppl.), 259.

[21] Moore, G.P.M., Jackson, N. and Lax, J. (1989). Evidence of a unique developmental mechanism specifying follicle density and fibre size in sheep selected for single skin and fleece characters. Genetical Research. 53, 57-62.

[22] Morley, F.H.W. (1955). Selection for economic characters in Australian Merino sheep. V. Further estimates of phenotypic and genetic parameters. Aust. J. Agric. Res. 6, 77-90.

[23] Mueller, J.P. and James, J.W. (1985) Phenotypic response to selection for traits with direct and maternal components when generations overlap. Theor. Appl. Genet. 70:123-127

[24] Pukelsheim, F. (1976) Estimating variance components in linear models. J. Multivariate Analysis 6:626-629

[25] Searle, S.R. Notes on variance component estimation. A detailed account of maximum likelihood and kindred methodology. Biometrics Unit, Cornell University, Ithaca, New York.

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

[27] Turner, H.N., Brooker, M.G. and Dolling, C.H.S. (1970). Response to selection in Australian Merino sheep. III. Single character selection for high and low values of wool weight and its components. Aust. J. Agric. Res. 21, 955-984.

[28] Watson, N., Jackson, N. and Whiteley, K.J. (1977). Inheritance of the resistance to compression property of Australian Merino wool and its genetic correlation with follicle curvature and various wool and body characters. Aust. J. Agric. Res. 28, 1083-94.

[29] Whiteley, K.J. and Jackson, N. (1981). Breeding for apparel wool. Proc. World Congress of Sheep and Beef Cattle Breeding, Palmerston North, New Zealand, October-November, 1980.

[30] Willham, R.L. (1963) The covariance between relatives for characters composed of components contributed by related individuals. Biometrics 19:18-27

[31] Wolak, M.E. (2012) 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

[32] Wright, S. (1922) Coefficients of inbreeding and relationship. Am. Nat. 56:330-338