WOMBAT

A program for Mixed Model Analyses by Restricted Maximum Likelihood

USER NOTES

Karin Meyer

Animal Genetics and Breeding Unit,
University of New England
Armidale, NSW 2351,
AUSTRALIA
kmeyer.agbu@gmail.com

MEAT & LIVESTOCK AUSTRALIA

ANIMAL GENETICS AND BREEDING UNIT

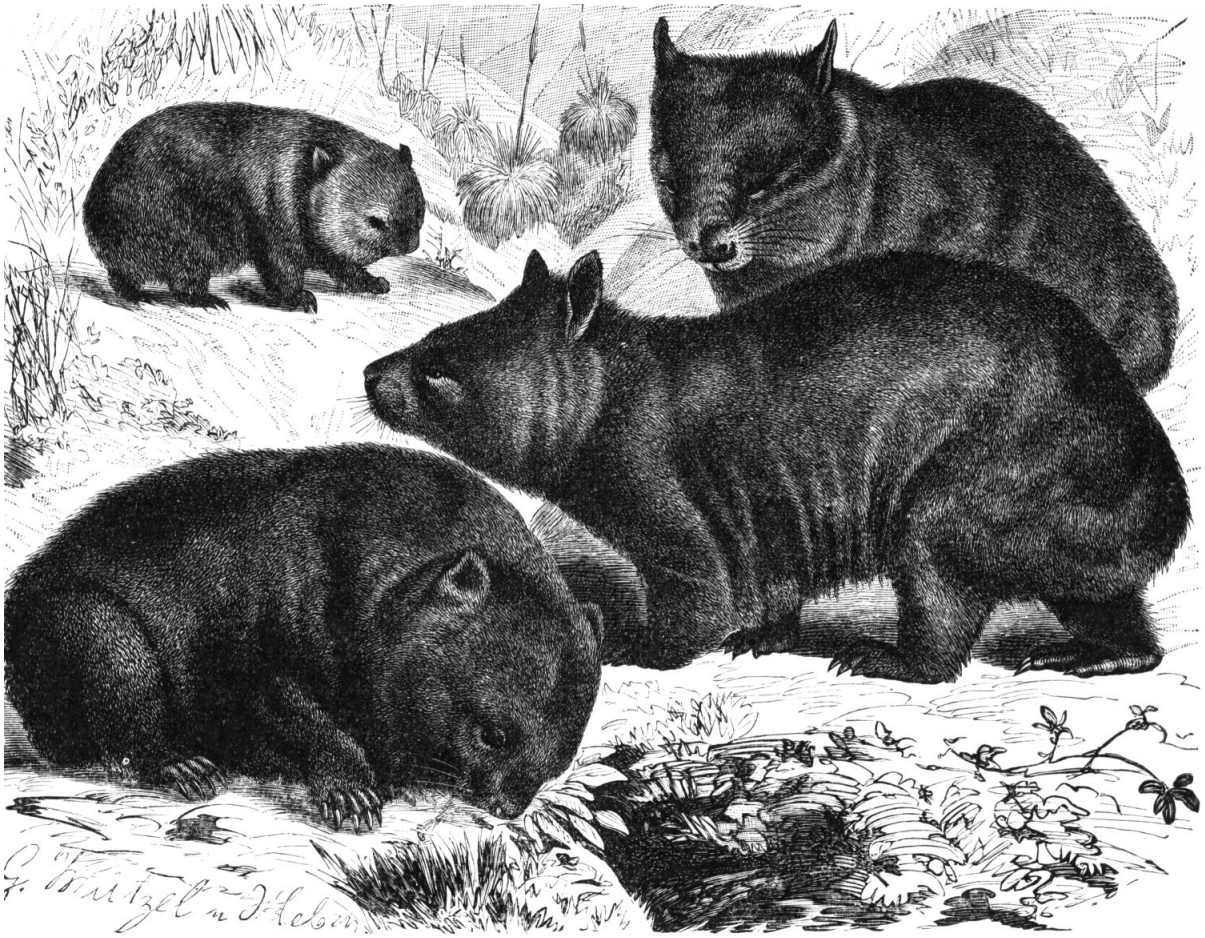
This document has been typeset using LaTeX2e with the **hyperref** package.

This gives a document which is fully navigable - all references to other sections and citations are ‘clickable’ within document links, and all links to external URLs can be accessed from within the PDF viewer (if this is configured to do so).

This copy of the manual has been produced on 8 March 2017.

Acknowledgements

Development of WOMBAT has been supported by Meat and Livestock Australia Ltd. (www.mla.com.au), and the International Livestock Resources and Information Centre (www.ilric.com).



Original caption: "Tasmanischer Wombat, *P*hascolomys ursinus G. Cuv. (links), und Breitstirnwombat, *P*h. latifrons Owen (rechts), 1/8 natürlicher Größe."

Translation (partly): "Coarse-haired wombat, *V*ombatus ursinus G. Cuv. (left), and Southern hairy-nosed wombat, *L*asiorhinus latifrons Owen (right), 1/8 natural size."

Originator: Gustav Mützel; Source: Brehms Tierleben, Small Edition 1927

1 Introduction

Purpose

WOMBAT is a program to facilitate analyses fitting a linear, mixed model via restricted maximum likelihood (REML). It is assumed that traits analysed are continuous and have a multivariate normal distribution.

WOMBAT is set up with quantitative genetic analyses in mind, but is readily applicable in other areas. Its main purpose is the estimation of (co)variance components and the resulting genetic parameters. It is particularly suited to analyses of moderately large to large data sets from livestock improvement programmes, fitting relatively simple models. It can, however, also be used as simple generalised least-squares program, to obtain estimates of fixed and predictions (BLUP) of random effects. In addition, it provides the facilities to simulate data for a given data and pedigree structure, invert a matrix or combine estimates from different analyses.

WOMBAT replaces DfReml [21, 22] which has been withdrawn from distribution at the end of 2005.

Features

WOMBAT consists of a *single* program. All information on the model of analysis, input files and their layout, and (starting) values for (co)-variance components is specified in a parameter file. A large number of run options are available to choose between (partial) analyses steps, REML algorithms to locate the maximum of the likelihood function, strategies to re-order the mixed model equations, and parameterisations of the model.

Types of analyses

- WOMBAT accommodates standard uni- and multivariate analyses, as well as random regression (RR) analyses, allowing a wide range of common models to be fitted and offering a choice between full and reduced rank estimation of covariance matrices.

REML algorithms

- WOMBAT incorporates the so-called ‘average information’ (AI) algorithm, and the standard (EM) as well as the ‘parameter expanded’ (PX) variant of the expectation-maximisation algorithm. In addition, derivative-free maximisation via Powell’s method of conjugate directions or the Simplex procedure is available. By default, WOMBAT carries out a small number of PX-EM iterates

to begin with, then switches to an AI REML algorithm.

Ordering strategies

- Computational efficiency and memory requirements during estimation depend strongly on the amount of 'fill-in' created during the factorisation of the coefficient matrix. This can be reduced by judicious ordering of the equations in the mixed model, so that rows and columns with few elements are processed first. Several ordering procedures aimed at minimising fill-in are available in WOMBAT, including MeTiS [16], a multilevel nested dissection procedure which has been found to perform well for large data sets from livestock improvement programmes.

Analysis steps

- WOMBAT allows for analyses to be broken up into individual steps. In particular, carrying out the 'set-up' steps separately facilitates thorough checking and allows memory requirements in the estimation step to be minimised.

Parameterisation

- Generally, WOMBAT assumes covariance matrices to be estimated to be unstructured. Estimation can be carried out on the 'original' scale, i.e. by estimating the covariance components directly, or by reparameterising to the matrices to be estimated elements of the Cholesky factors of the covariance matrices. The latter guarantees estimates within the parameter space, in particular when combined with a transformation of the diagonal elements to logarithmic scale. Reduced rank estimation, equivalent to estimating the leading principal components only, is readily carried out by estimating the corresponding columns of the respective Cholesky factor(s) only.

WOMBAT offers few features related to data editing, such as selection of subsets of records, transformation of variables or tabulation of data features. There are a number of general statistical packages to choose from which perform these tasks admirably, including free software such as the R package [15].

Remarks

This document is a manual with instructions how to use WOMBAT. It does not endeavour to explain restricted maximum likelihood estimation and related issues, such as approximation of sampling errors, likelihood ratio test, use of information criteria, or how to assess significance of fixed effects.

Throughout the manual, it is assumed that users have a thorough knowledge of mixed linear models and a sound understanding of maximum likelihood inference in general. Numerous textbooks covering

these topics are available. To get the best from WOMBAT, users should also be familiar with some of the technical aspects of REML estimation, in particular properties of various algorithms to maximise the likelihood, ordering strategies and parameterisations – otherwise many of the (advanced) options provided will not make a great deal of sense.

2 Availability

WOMBAT is available only on a help-yourself basis, via downloading from **http://didgeridoo.une.edu.au/km/wombat.php**

Material available comprises the compiled program ('executable'), together with these User Notes and a suite of worked examples.

WOMBAT has been developed under and is meant for a Linux operating system. Highly optimised executables are available for this environment that are suitable for large analyses.

In addition, a Windows versions are provided – best to be run in a CMD window.

Conditions of use

WOMBAT is available to the scientific community free of charge. Users are required, however, to credit its use in any publications.

Suggested (preferred) form of citation:

> Meyer, K. (2007). WOMBAT – A tool for mixed model analyses in quantitative genetics by REML, *J.* Zhejiang Uni. SCIENCE B **8**: 815–821. [doi:10.1631/jzus.2007.B0815]

Alternatives:

> Meyer, K. (2006). WOMBAT – Digging deep for quantitative genetic analyses by restricted maximum likelihood. *P*roc. 8th World Congr. Genet. Appl. Livest. Prod., Communication No. 27–14.
>
> Meyer, K. (2006). WOMBAT – A program for mixed model analyses by restricted maximum likelihood. User notes. Animal Genetics and Breeding Unit, Armidale, $n$pp.

All material is provided on an 'as is' basis. There is no user support service, i.e. this manual is the only help available !

DISCLAIMER

While every effort has been made to ensure that WOMBAT does what it claims to do, there is absolutely no guarantee that the results provided are correct.

Use of WOMBAT is entirely at your own risk !

3 Getting started with WOMBAT

3.1 Installation

WOMBAT is distributed as a pre-compiled executable file. A number of versions are available. The main target operating system is Linux. In addition, executable files for Windows environments are available.

3.1.1 Installation for Linux operating systems

1. Download the version appropriate to your machine from
   http://didgeridoo.une.edu.au/km/wombat.php
2. Uncompress and unpack the file ($nn = 32$ or 64 or win) using
   `tar -zxvf wombat`*nn*`.tar.gz`
   This will create the directory **WOMBAT** which contains the executable **wombat**.
3. Check that your system recognises **wombat** as an executable file, e.g. using
   `ls -l WOMBAT/wombat`
   If necessary, use the **chmod** command to add executable permission or access permission for other users, e.g.
   `chmod a+x WOMBAT/wombat`.
4. Make sure that the operating system can find **wombat**, by doing *one* of the following :
   - ❏ Add the directory **WOMBAT** to your **PATH** statement, or
   - ❏ Move **wombat** to a directory which is included in your **PATH**, e.g.
     - ○ **/usr/local/bin** (this requires super-user privileges), or

- ∼/**bin** in your home directory,

or

- Make a symbolic link from a **bin** directory to **WOMBAT/wombat** (using **ln -s**).

Hint

PATH is set in your login file, e.g. **.login_usr**, or shell start-up file, e.g. **.tcshrc**; use **printenv PATH** to check your current PATH settings.
You make have to log out and log in again before changes in your **PATH** are recognised, and **wombat** is found by the system.

3.1.2 Installation under Windows

Windows is not an operating system that is well suited to scientific computations. Two versions of WOMBAT are available that will run under Windows. However, these are slower than their Linux counterparts and restricted in the size of the analyses that they will accommodate (see Section 3.1.5).

1. Download
   http://didgeridoo.une.edu.au/km/wombat.php

   - **wombatwin2.zip**

   UNZIP **wombatwin2.zip**. This will give you the executable file **wombat.exe**.

2. Check that this is recognised as an executable file by clicking the file properties tab – change if required.

3. Move **wombat.exe** to a directory of your choice and, if necessary, add this directory to the **PATH** that the operating system searches when looking for executable files – see the documentation for your version of Windows for details.

Hint

You may be able to inspect the setting for **PATH** by opening a command window (type cmd.exe into the "Run" box in the start menu & hit return) and typing **ECHO %PATH%**.
Under XP, right-click on **My Computer** (from the start menu), choose **Properties**, then **Advanced** and the **Environment Variables** tab. This will open a dialog box which will list the existing variables and their settings. If **PATH** already exists, choose **Edit** and add the **WOMBAT** directory – remember to use the full path and to separate it from the existing variables by a semi-colon. Otherwise choose **New** and create it.

3.1.3 Running WOMBAT

WOMBAT is expected to be run from a command line interface, i.e. you need to open a 'terminal' window and type in the appropriate command, hitting return to start the program running.

The general form of the command (under Linux and its emulation) is

wombat *o*ptions **parfile**

with **parfile** the 'parameter file' specifying everything WOMBAT needs to know about the input files and models of analysis (see Chapter 4 for full details), and *o*ptions any run time options you may wish to give (see Chapter 5). If you have not put the executable file somewhere the operating system can find it (or have not modified your **PATH** settings), you will need to replace **wombat** above by the full path, e.g. **/home/agbu/WOMBAT/wombat** if the program has been placed in **/home/agbu/WOMBAT**.

3.1.3.1 Running WOMBAT under Windows

- Click "Start", then "Run", type CMD.EXE into the "Open" box and click "O.K.", or
- Find "Command Prompt" under "Accessories" (Under "All Programs") and click it. To use this regularly, add this to your "Start" menu.

If you have installed MinGW/MSYS, you should open such 'terminal' and run WOMBAT in it in the same way as under Linux. Otherwise, you should open a DOS type window and type the command in it. Remember to use backwards rather than forward slash(es) if you need to type a full path to the executable.

You can run **wombat.exe** by simply double-clicking on it. However, this is *not* recommended! WOMBAT creates screen output which lets

you monitor how a run is progressing. More importantly, this screen output will provide some diagnostics when something has gone wrong, e.g. help you remedy errors in the parameter or other input files. Hence, if you really want to start WOMBAT by double-clicking on an icon, you should make sure that the window which is opened does not close automatically after the run finishes or aborts! A simple way to do this is to create a `.bat` file.

wombat.bat

```
rem Simple .bat file to run WOMBAT in a DOS window
rem Assumptions:
rem  1) parameter file is wombat.par
rem  2) other run time options are given in wombat.par
rem  3) wombat.exe resides in  C:\Program_Files\WOMBAT
@echo off
color 70
echo "Ready to run WOMBAT"
pause
C:\Program_Files\WOMBAT\wombat -v
echo "WOMBAT has finished"
pause
exit
```

3.1.4 Examples and testing

Install examples

A number of worked examples, complete with output files and captured screen output, are provided; see chapter Chapter 9 for further details. These can be downloaded and installed in analogous manner to the program. This description is appropriate for Linux and Cygwin; for Windows adjust as appropriate, e.g. by replacing forward with backward slashes, etc.

1. Download **`examples.tar.gz`** from
   http://didgeridoo.une.edu.au/km/wombat.php
2. Uncompress and unpack the file using
   `tar -zxvf examples.tar.gz`
   Do this in the same directory as above! This will create the directory **`Examples`** with subdirectories **`Example1`**, . . ., **`Example`**n.

Test runs

Each subdirectory contains the input and output files for a run of WOMBAT, together with a records of the screen output in the file **`typescript`**. To test your installation,

1. Choose an example and change into its directory.
2. Make a new temporary subdirectory, e.g. **`try`**.

N.B.

This should be a 'parallel' directory to sub-directories **A**,**B**,. . ., i.e. **Example**n**/try** not **Example**n**/A/try**. Otherwise, you need to adjust the paths to the data and pedigree files in the parameter file for WOMBAT to be able to find them !

3. Copy all input files to the temporary directory.
4. Change into the temporary directory and run WOMBAT, using the same command line options (if any) as shown in the beginning of **typescript**.
5. Compare your output files with those supplied – apart from date and time they should be virtually identical; small discrepancies in higher decimals may occur though.
6. Delete temporary subdirectory.

Repeat for at least one more example.

3.1.5 Compilation notes

3.1.5.1 Linux

64-bit versions of WOMBAT have been compiled under Ubuntu 14.04 or CentOS 6, using either the Lahey™ lf95 (version 8.00a) or the Intel™ ifort (version 13.01.1) compiler. For the latter, versions loading multi-threaded BLAS and LAPACK routines from the MKL library are available.

32-bit versions are no longer maintained.

3.1.5.2 Windows

The Windows versions of WOMBAT have been 'cross-compiled' under Linux using the `gfortan` compiler. Both 32- and 64-bit versions are available.

Previous versions compiled under Cygwin or MinGW are no longer maintained.

3.1.6 Updates

Expiry

WOMBAT is set up to 'expire' after a certain date. Usually this is approximately 2–3 years after compilation (not download!) date. This feature aims at reducing the number of outdated copies being used, and any associated problems. WOMBAT will print out a warning message when used in the month preceding the expiry date. In addition, a run time option is available to query the program for this date.

If your copy of WOMBAT has expired – or you simply want to update to a newer version, please repeat the installation steps outlined above (section 3.1).

3.2 Using the manual

Must read

WOMBAT caters for novice users by supplying defaults for most aspects of estimation. Essential sections of the manual to 'get started' are :

- Chapter 4 on how to set up the parameter file,
- Chapter 6, especially sections 6.2 and 6.3 on the format of the data and pedigree file, respectively.
- Chapter 7 which describes the output files generated by WOMBAT, in particular section 7.1.
- Chapter 9 which describes the worked examples available.

The most 'difficult' part of using WOMBAT is to *correctly* set up the parameter file. The detailed rules given in chapter 4 are best understood following some example(s). The suite of examples provided templates for various types of analyses performed by WOMBAT, and a range of different models.

Hint

A suitable strategy might be

i) Choose the type of analysis you are interested in, and decide on the model of analysis. Start with a relatively simple scenario.
ii) Try to find an example which matches the type of analysis and fits a not too dissimilar model.
iii) Inspect the example parameter and input files.
iv) Read the description of individual entries in the parameter file (Chapter 4). Compare each section to the relevant section in the example parameter file.
v) Try to modify the example file for your data (& pedigree) file and model.

3.3 Troubleshooting

WOMBAT has undergone fairly rigorous testing, more so for some models than for others. However, there are bound to be errors and inconsistencies – especially early in its development.

Input errors

Errors in the input files are the most likely source of problems which are *not* a program bug. Some of these are trapped, leading to a programmed error stop (with a screen message of "`exit WOMBAT`"

or "`Programmed (error) stop for WOMBAT encountered`"). Almost certainly, this is due to erroneous input, in particular a mistake in the parameter file ! You should be able to figure out what is wrong from the accompanying brief error message and fix it. Others might simply cause the program to abort.

Program bugs

If – after thoroughly checking your parameter file and other input files – you think you have encountered a *genuine* error in the program, please submit a 'bug report', as specified below[1].

To submit an *informative* 'bug report', please carry out the following steps:

1. Download the *latest* version of WOMBAT which has been compiled with optimisation switched off, and checks switched on from **http://didgeridoo.une.edu.au/km/wombat.php**
   - **`wombatcheck.tar.gz`**

   and extract the executable **`wombat_chk`** as described above. This is a 32-bit version, which should run on both 32- and 64-bit Linux machines.
2. Use **`wombat_chk`** for run(s) to demonstrate the error.
3. Try to recreate the problem using one of the test data sets and pedigree files supplied. Edit these as appropriate to generate the structure causing the problem if necessary, e.g. delete or add records or effects in the model.
   Only if *absolutely* necessary, use a small subset of your data and pedigree files – the smaller the better – but definitely less than 100 animals in the pedigree and less than 500 records !
4. Use a new, 'clean' directory for the run(s).
5. Run **`wombat_chk`** with the **`-v`** or **`-d`** option.
   Remember that **`wombat_chk`** is compiled with checking switched on and thus will require several times longer than **`wombat`** for the same task.
6. Capture all screen output using the **`script`** command.
7. **`tar`** the complete directory and **`gzip`** it (or use any other common compression utility), and send it to me.
   I need all input files, output files and the **`typescript`** file !
8. If you have any theory on what might be the problem, please tell me.

[1] Never send any .doc, .rtf, etc files !

This may sound like a lot of work, but is necessary to for me to even begin to try understanding what is going on !

4 The Parameter File

4.1 Overview

All information on the the model of analysis and the data and pedigree files is specified through the *parameter file*.

File name

The name of the parameter file should have extension '.par'. By default, WOMBAT expects to find a file **`wombat.par`** in the current working directory.
Other filenames can be specified at run time as the last command line option – see Chapter 5 for details.

Setting up the parameter file is straightforward, but care and attention to detail are required to get it 'just right'. The worked examples give 'templates' for various types of analyses which are readily adaptable to other problems. Parsing of the lines in the parameter file is fairly elementary, hence it is important to adhere strictly to the format

described in the following in painstaking detail.

Checking

WOMBAT performs a limited number of consistency checks on the variables and models specified, but these are by no means exhaustive and should not be relied upon !

Error stops

If WOMBAT does find an obvious error, it will stop with a message like "`exit WOMBAT`" or, in verbose mode, "`Programmed(error) stop for WOMBAT encountered`". Inconsistencies or errors not discovered are likely to wreak havoc in subsequent steps !

Hint

Use the **`-v`** option at run time. This will cause WOMBAT to echo each line in the parameter file as it is read (to the screen) – if there is a programmed error stop at this stage, it is easier to find the mistake as you know which is the offending line.

4.2 General rules

Line length

The parameter file is read line by line. Each line is assumed to be 88 characters long, i.e. anything in columns 89 onwards is ignored.

Comments

Any line with a **#** in column 1 is considered to be a comment line and is skipped.

Info codes

WOMBAT relies on specific codes at the beginning of each line (leading blank spaces are ignored) to distinguish between various types of information given. Valid codes are summarised in Table 4.1. Most codes can be abbreviated to 3 letters. The code and the information following it should be separated by space(s). Codes are not case sensitive, i.e. can be given in either upper or lower case letters.

Depending on the initial code, WOMBAT expects further information on the same or following lines. All variable and file names specified are treated as case sensitive. File names can be up to 30 characters long, and variable names can comprise up to 20 characters. All codes, names and variables must be separated by spaces.

The parameter file can have two different types of entries :

1. 'Short' entries (e.g. file names, analysis type, comment) which consist of a single line.
   Each of these lines must start with a code for the type of information given.

Example

```
PEDS pedigrees.dat
```

Here **PEDS** is the code for pedigree information, “pedigrees.dat” is the name of the file from which pedigree information is to be read.

2. ‘Long’ or block entries, spanning several lines (e.g. for the layout of the data file, or the model of analysis).
   Each of these entries starts with a line which gives the code for the entry (see Table 4.1), and possibly some additional information. This is followed by lines with specific information, where of each of these lines again starts with a specific code. Except for blocks of starting values of covariance components, the block is terminated by a line beginning with **END**.

Example

```
MODEL
  FIX  CGroup
  RAN  Animal  NRM
  TRA  Weight
END MODEL
```

This shows a block entry for the model of analysis, where the model fits **CGroup** as a crossclassified, fixed effect and **Animal** as random effect with covariance matrix proportional to the numerator relationship matrix, and **Weight** is the trait to be analysed.

Different entries should be specified in the order in which they are listed in the following sections.

4.3 Run options

While run time options (see Chapter 5) are primarily intended to be read from the command line, WOMBAT also provides the facility to specify such options on the *first* line of the parameter file. To be recognised, this line must start with the code **RUNOP** and, after space(s), all options are expected to be given on the same line (space separated), in the same form as on the command line.

4.4 Comment line

WOMBAT allows for a single, optional comment line (up to 74 characters) to be specified. This is specified by the code **COMMENT** (can be

Table 4.1: Valid codes for entries in the parameter file

Code	Within	Indicator for
RUNOP	–	Line with run options
COMMENT	–	Comment line
PEDS	–	Name of pedigree file
DATA	–	Name of data file
TRNOS	**DAT**	Trait numbers (grouped input)
NAMES	**DAT**	Multiple column names (grouped input)
ANALYSIS	–	Code for analysis type
MODEL	–	Model of analysis
FIX	**MOD**	Name of fixed effect
COV	**MOD**	Name of fixed covariable
RAN	**MOD**	Name of random effect
RRC	**MOD**	Name of control variable
SUBJ	**MOD**	Name of variable identifying “subject”
EXT	**MOD**	Name extra variable
TRAIT	**MOD**	Trait name
ZEROUT		Fixed effects levels to be set to zero
PSEUDOFX		Random effects levels to be treated as random
SE+USR		Additional, user defined functions of covariances
SUM	**SE+**	Weighted sum
VRA	**SE+**	Variance ratio
COR	**SE+**	Correlation
POOL	–	Options for pooling of covariance components
PSEUPED	**POOL**	Pseudo pedigree structure
SMALL	**POOL**	Minimum eigenvalue
SPECIAL	–	Miscellaneous options
COVZER	**SPECIAL**	How to deal with ‘zero’ covariables
WEIGHT	**SPECIAL**	Weighted analysis
REPEAT	**SPECIAL**	Run with low proportion of repeated records
RPTCOV	**SPECIAL**	Specify residual covariance structure
CLONES	**SPECIAL**	Ignore sorting of data file
QTLEFF	**SPECIAL**	Specify which covariable is a QTL/SNP effect
PENALTY	**SPECIAL**, **POOL**	Invoke penalized estimation
SOCIAL	**SPECIAL**	Analysis with ’social’ genetic effect
INCORE	**SPECIAL**	Analysis storing info in core
FORCE-SE	**SPECIAL**	Force calculation of standard errors
GENGROUPS	**SPECIAL**	Fit ‘explicit’ genetic groups
AOM-RES	**SPECIAL**	Calculate outlier statistics for residuals
VARIANCE	–	Name of random effect
RESIDUAL	–	
ERROR	–	
END		

abbreviated to **COM**) at the beginning of the line. Anything following the spaces after **COM**(**MENT**) is treated as comment on the analysis. It is printed in some of the output files generated by WOMBAT to assist the user, and has no other use.

4.5 Analysis type

This is a single line entry beginning with code **ANALYSIS** (can be abbreviated to **ANA**), followed by a two- or three-letter code describing the type of analysis. The following codes are recognised :

UNI : for a univariate analysis,
MUV : for a 'standard' multivariate analysis,
RR : for a single-trait random regression analysis, and
MRR : for a multi-trait random regression analysis.

Principal components

Except for **UNI**, this can be followed (after space(s)) by the code **PC** to select an analysis which fits the leading principal components only for some of the random effects fitted and yields reduced rank estimates of the corresponding covariance matrices. For **MUV** and **MRR** the number of traits in the analysis must be given as well – this should be the last entry of the line.

Example

```
ANALYSIS MUV PC 8
```

specifies a reduced rank, multivariate analysis for 8 traits

4.6 Pedigree file

This is a single line entry beginning with code **PEDS** (can be abbreviated to **PED**), followed by the name of the pedigree file. There is no default name. This entry is 'optional', in such that it is only required if a code of **NRM** if specified for the covariance structure of a random effect (see Section 4.8.1.2). Only one pedigree file can be given. The format of the pedigree file required by WOMBAT is described in detail in Section 6.3.

Sire model

By default, WOMBAT fits an animal model. If a sire model is to be fitted, the code **SIREMODEL** (can be abbreviated to **SIR**) should be given after the filename.

Additional options, given after the filename and, if applicable the sire model option, that are recognised are **+INBR** or **+SEX**. These instruct WOMBAT to read animals' inbreeding coefficients or sex

Inbreeding

codes (1=X/2=XX) from the fourth of fifth column in the pedigree file. If given, **+INBR** causes WOMBAT to skip calculation of inbreeding coefficients and use the values supplied instead. Option **+SEX** is required if an NRM for X-linked genetic effects is to be set up.

4.7 Data file

This is a block entry. The block begins with a line containing the code **DATA** (can be abbreviated to **DAT**) followed by the name of the data file. There is no default name. The general form of the data file required in described in Section 6.2.

The following lines specify the record layout of the data file for all traits. There are two alternative ways of specification.

4.7.1 Simple

For each trait in turn, there should be one parameter file line for each column, up to the last column used in the analysis.
The lines can have up to 3 elements :

(a) The code **TR**n where n is a one- or two-digit trait number. This can be omitted for univariate analyses.
(b) The name of the variable in this column.
(c) The maximum number of levels. This is required if the column represents a fixed or random effect in the model of analysis or a control variable in a random regression analysis.

The block is terminated by a line with the code **END**.

Example

```
DATA  mydata.dat
  TR1  traitno 2
  TR1  animal  1000
  TR1  fixeffect 50
  TR1  weight
  TR2  traitno 2
  TR2  animal 500
  TR2  fixeffect 30
  TR2  feedintake
END DATA
```

This shows the block for an analysis reading records for 2 traits from the file **mydata.dat**.

4.7.2 Compact

If there are several traits for which the record layout is the same, the respective record layout can be given for the whole group of traits. This avoids tedious duplication of lines.

Grouped traits

This alternative is selected by placing the code **GRP** after the name of the data file (same line, separated by a space).
For each group of traits, the following lines need to be given :

1. A 'header' line beginning with the code **TRNOS** (can be abbreviate to **TRN**), followed by the running numbers of the traits in the group on the same line.
2. One line for each column in the data file (up to the last column used) which is the same for all traits, containing
   (a) the variable name
   (b) the maximum number of levels, if the column represents a fixed or random effect in the model of analysis
3. One line for each column which has a different name for different traits (e.g. representing the traits to be analysed), containing
   (a) the code **NAMES** (can be abbreviated to **NAM**)
   (b) the variable names (on the same line, space separated; the same number of variables as traits in the group must be given)

Again, the block is terminated by a line with the code **END**.

Example

```
DATA  mydata.dat  GRP
   TRNOS  1 2
   traitno     2
   animal   1000
   fixeffect  50
   NAMES   weight  feedintake
END DATA
```

This shows the 'grouped' alternative for specifying the data file layout, for two traits with the same column structure in the example above.

4.8 Model of analysis

MODEL

This is another block entry. The block begins with a line containing the code **MODEL** (can be abbreviated to **MOD**), and finishes with a line beginning with **END**. The block then should contain one line for each effect to be fitted and one line for each trait in the analysis.

4.8.1 Effects fitted

Each of the 'effect' lines comprises the following

(a) a three-letter code for the type of effect,
(b) the effect name, where the effect name is a combination of the variable name for a column in the data file and, if appropriate, some additional information.
No abbreviations for variable names are permitted, i.e. there must be an exact match with the names specified in the **DATA** block.
(c) If the effect is to be fitted for a subset of traits only, the running numbers of these traits must be given (space separated).

4.8.1.1 Fixed effects

Fixed effects can be cross-classified or nested fixed effects or covariables. The following codes are recognised :

FIX : This specifies a fixed effect in the model of analysis.

NB The name for a fixed effect should not contain a "(", otherwise it is assumed that this effect is a covariable.

A simple, one-way interaction of two variables can be specified as **vn1*vn2**, with **vn1** and **vn2** valid variables names. [Not yet implemented !]

Hint

'Not implemented' here means merely that WOMBAT will not code the interaction for you – you can, of course, still fit a model with an interaction effect, but you a) have to insert an additional column in the data file with the code for the appropriate subclass, b) fit this as if it were a crossclassified fixed effect, and c) specify any additional dependencies arising explicitly (using **ZEROUT**, see below).

COV : This specifies a fixed covariable. The effect name should have the form "**vn**(n,**BAF**)", where **vn** is a variable name, the integer variable n gives the *number* of regression coefficients fitted and **BAF** stands for a three-letter code describing the basis functions to be used in the regression. NB: By default, intercepts for fixed covariables are not fitted.

Valid codes for basis functions are

POL : for ordinary polynomials.
This is the default and can be omitted, i.e "**vn**(n)" is equiva-

lent to "**vn**(n,**POL**)". For instance, $n = 2$ denotes a quadratic regression. Note that WOMBAT deviates both records and covariables from their respective means prior to analysis.

N.B.
This yields a regression equation of the form

$$y - \bar{y} = b_1 (x - \bar{x}) + b_2 (x - \bar{x})^2 + \cdots$$

rather than an equation of form

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots$$

This should be born in mind when interpreting any solutions for regression coefficients for **POL** covariables from WOMBAT - while there is a straightforward relationship between coefficients β_i and b_i, they are not interchangeable.

LEG : for Legendre polynomials.
For example, $n = 3$ denotes a cubic polynomial, i.e. comprises a linear, quadratic and cubic regression coefficient, but no intercept (This differs from the implementation for random regressions where $n = 4$ denotes a cubic polynomial).

BSP : for B-spline functions
For analyses fitting spline functions, the degree of the spline is selected by specifying "L", "Q" or "C" for linear, quadratic and cubic, respectively, immediately (no space) after the code **BSP**. Note that the default spline function is an equidistant B-spline (i.e. the range of values of the covariable is divided into intervals of equal size), with the number of knots and intervals determined from the number of regression coefficients and the degree of the spline ($k = n - d + 1$ where k is the number of knots and d is the degree, $d = 1$ for "L", $d = 2$ for "Q" and $d = 3$ for "C") [24, section 2.2]. Other spline functions are readily fitted as user-defined basis functions.

Splines

USR : for user defined functions

Fitting of an intercept (in addition to deviation from means) can be enforced by preceding n with a minus sign – this is not recommended unless there are no other fixed effects in the model.

A covariable to be fitted as nested within a fixed effect is specified as "**vn1***vn2(n,**BAF**)", with **vn1** the name of the fixed effect. If **vn1** is not fitted as a fixed effect, it must be specified an an 'extra' effect (see below).

Nested covariable

Zero covariable

WOMBAT is fussy when it encounters a covariable which has a value of zero: Covariables which can take such value are only permitted if a SPECIAL option is given which confirms that these are valid codes and not 'missing' values; see Section 4.10.6.

4.8.1.2 Random effects

Random effects include the 'control variables', i.e. random covariables for random regression analyses. The following codes are recognised:

RAN : This code specifies a random effect. It should be followed (space separated) by the name of the random effect. After the name, a three-letter code describing the covariance structure of the effect can be given.
Valid codes for covariance structures are :

Genetic effect

NRM : which denotes that the random effect is distributed proportionally to the numerator relationship matrix.
If this code is given, a pedigree file must be supplied.

X-linked effect

SEX : which denotes that the random effect is distributed proportionally to the numerator relationship matrix for X-linked genetic effects. This inverse of this matrix is set up directly from the pedigree information, as described by Fernando and Grossman [10].
If this code is given, it is assumed that an autosomal genetic effects (option **NRM** with corresponding pedigree file) is also fitted and has already been specified. In addition, the pedigree file is expected to have an additional column specifying the number of X-chromosomes for each individual.

IDE : which denotes that different levels of the random effect are uncorrelated. This is the default and can be omitted.

GIN : which denotes that the random effect is distributed proportionally to an 'arbitrary' (relationship or correlation) matrix. The user must supply the inverse of this matrix in the form outlined in Chapter 6.

PEQ : which denotes a permanent environmental effect of the animal for data involving 'repeated' records, which is not to be fitted explicitly. Instead, an equivalent model is used, which accounts for the respective covariances as part of the residual covariance matrix. This is useful for the joint analysis of traits with single and repeated records.

N.B.
Do not use this option for other effects - WOMBAT has no mechanism for checking that this option is appropriate.

For 'standard' uni- and multivariate analyses, the random effect name is simply the variable name as given for data file.

For random regression analyses, the variable name is augmented by information about the number of random regression coefficients for this effect and the basis functions used. It becomes "**vn**(n,**BAF**)", analogous to the specification for covariables above. As above, n specifies the number of regression coefficients to be fitted. In contrast to fixed covariables, however, an intercept is always fitted. This implies that n gives the order, not the degree of fit. For instance, $n = 3$ in conjunction with a polynomial basis function specifies a quadratic polynomial with the 3 coefficients corresponding to the intercept, a linear and a quadratic term. WOMBAT allows for different control variables to be used to fit random regressions for different effect. If the model has more than one **RRC** statement (see below), the specification of random effects needs to be extended to tell WOMBAT which control variable to use for which effect, i.e. "**vn**(n,**BAF**,**rrcn**)" with **rrcn** the name of the control variable.

Valid codes for **BAF** are:

POL : for ordinary polynomials.
This is the default and can be omitted, i.e "**vn**(n)" is equivalent to "**vn**(n,**POL**)".

LEG : for Legendre polynomials.

BSP : for B-spline functions

USR : for user defined functions

IDE : for an identity matrix, i.e. the $i-$th basis function has a single coefficient of "1" for the $i-$th coefficient with all other elements zero. This requires the number of RR coefficients to be equal to the number of levels of the control variable. It is useful when fitting a multi-trait model as a RR model, e.g. to allow for heterogeneous residual variances.

ONE : to assign a value of unity ("1") to all coefficients.
This option has been introduced to facilitate a standard multivariate analysis with repeated records by fitting a random regression model to model an arbitrary pattern of temporary environmental covariances correctly.

RRC : This code specifies a 'control variable' in a random regression analysis. It should be followed (space separated) by the name of the variable, as given in the DATA statement.
Optionally, immediately after the name (no spaces), the range

of values of the variable to be considered can be specified as $(m - n)$, with m the lower and n the upper limit.

N.B.
WOMBAT expects the value of the control variable to be non-negative (i.e. 0 or greater) and not to be a fraction (the control variable is read as a real variable but converted to the nearest integer, e.g. values of 3,.0, 3.1 and 3.4 would be treated as 3) – scale your values appropriately if necessary!

N.B.
For multivariate analyses, WOMBAT collects the range of values for the control variable (i.e. minimum and maximum) across all traits. This may be undesirable if different traits have distinct ranges and Legendre polynomials are used as basis function - if so, use the **USR** option and set up your own file which maps the range for each trait exactly as you want it!

4.8.1.3 'Subject' identifier

For animal model analyses, WOMBAT assumes that the code for the first genetic effect fitted also identifies the subject on which measurements are taken. For some analyses (in particular those *not* fitting any additive genetic effects!) and sire models this is not appropriate, and such identifier needs to be supplied as an extra column in the data file.

SUBJ : This code needs to be followed by the name of the variable which identifies the individual.

4.8.1.4 'Extra' effects

For some models, coding is required for effects which are not explicitly fitted in the model of analysis, for instance, when fitting nested effects. These need to be specified as 'extra' effects.

EXT : This code, followed by the respective variable name, denotes an effect which is not fitted in the model but which is required to define other effects.

4.8.2 Traits analysed

TRAIT

One line should be given for each trait. It should contain the following information :

(a) The code **TRAIT** (can be abbreviated to **TR**).
(b) The name of the trait, as specified in the **DATA** block.

(c) The running number of the trait.
- In most cases, this is simply the number from 1 to q, where q is the total number of traits in a multivariate analysis.
- In addition, WOMBAT provides the opportunity to replace the trait number in the data file with a different number. This is useful, for instance, to carry out an analysis involving a subset of traits without the need to edit the data file. The syntax for this is : "k"->m. This specifies that value k in the data file should be replaced with value m for the analysis. If this is encountered, any records with trait numbers not selected in this fashion are ignored, i.e. this provides a mechanism for automatic subset selection.

Hint

All q traits in the analysis should be specified in this manner, even if $k = m$ for some trait(s).

(d) Optional : A numeric value (integer) representing a 'missing' value - any records with the trait equal to this value are ignored in the analysis (default is -123456789).[1]

4.9 Covariance components

Next, the parameter file needs to specify values for all (co)variance components in the model of analysis. For variance component estimation, these are the starting values used. For simple BLUP analyses and simulation runs, these are the values assumed to be the population values.

The input matrices for full rank analyses must be positive definite, i.e. cannot have eigenvalues less than the operational zero. For reduced rank (**PC**) analyses, some 'zero' but no negative eigenvalues are acceptable, provided the rank (i.e. the number of non-zero eigenvalues) is equal to (or greater than) the number of principal components to be fitted.

4.9.1 Residual covariances

4.9.1.1 'Standard' multivariate analyses

The residual covariance matrix is specified by

[1] This may sound contradictory to Section 6.2 - but that comment pertains to multivariate analyses, where WOMBAT expects a separate record for each trait. The 'missing value' here merely represents a simple mechanism which allows selected records in the data file to be skipped.

1. A line beginning with the code **RESIDUAL** (can be abbreviated to **RES**) or **ERROR** (can be abbreviated to **ERR**). This is followed by the dimension of the covariance matrix, q (integer number, space separated). If an analysis type **PC** has been specified, a second number specifying the rank of the estimated covariance matrix, needs to be given (even if this is equal to q).
   Optionally, this can be followed (space separated) by a qualifier:
   DIAG : specifies that the residual covariance matrix is diagonal
   NOSTART : (can be abbreviated to **NOS**) specifies that no starting values are given; this is only valid in conjunction with the run option **--pool** !
2. Without qualifier: The $q(q+1)/2$ elements of the *upper* triangle of the residual covariance matrix, given row-wise (i.e. σ_1^2, σ_{12}, ..., σ_{1q}, σ_2^2, σ_{23}, ..., σ_q^2).
   These can be given several elements per line (space separated, taking care not to exceed the total line length of 78 characters), or one per line, or a mixture – WOMBAT will attempt to read until it has acquired all $q(q+1)/2$ elements required.
   With qualifier: If **DIAG** is specified only the q diagonal elements (σ_1^2, ..., σ_q^2) are read, and if **NOSTART** is given, no values are read.

4.9.1.2 Random regression analyses

Again, the residual covariance matrix is specified by

1. A line beginning with the code **RESIDUAL** (can be abbreviated to **RES**) or **ERROR** (can be abbreviated to **ERR**). This should be followed by the dimension (q) of each residual covariance matrix (integer number), usually equal to the number of traits, and a code indicating what kind of error covariance function is to be fitted. The following codes are recognised :
   HOM : This code specifies homogeneous error covariances for all values of the control variable.
   HET : This code specifies heterogeneous error covariances, with the covariance function a step function of the control variable. It should be followed (space separated) by an integer number giving the number of steps.
2. If the model involves multiple control variables, the name of the variable to be used to determine the temporary environmental covariance structure needs to be given at the end of the line.
3. One or more lines with the residual covariance matrices, consisting of the $q(q+1)/2$ elements of the upper triangle given row-wise, and, if applicable, additional information.

- For **HOM** only a single covariance matrix needs to be given.
- For **HET** the number of covariance matrices specified must be equal to the number of steps (in the step function, i.e. intervals). Each should begin on a new line and be preceded by two integer values (space separated) which give the upper and lower limits (inclusive) of the interval of the control variable, for which this matrix is applicable.

N.B.

'Step intervals' must cover the complete range of values for the control variable encountered in the data.

Example

For a univariate RR analysis with values of the control variable ranging from 1 to 5, say we want to fit separate residual variances for 1 & 2, 3 & 4 and 5:

```
VAR  residual  1  HET  3
    1  2   1.234
    3  4   3.456
    5  5   5.678
```

4.9.2 Covariances for random effects

Similarly, for each covariance matrix due to random effects to be estimated, a 'header' line and the elements of the covariance matrix need to be specified.

1. A line beginning with the code **VARIANCE** (can be abbreviated to **VAR**), followed by the name of the random effect and the dimension of the covariance matrix, q (integer number, space separated). If an analysis type **PC** has been specified, a second number specifying the rank of the estimated covariance matrix, needs to be given (even if this is equal to q).
   The name can simply the random effects name as specified for the model of analysis. Alternatively, it can be of the form "**vn1**+**vn2**" where **vn1** and **vn2** are names of random effects specified in the **MODEL** block. This denotes that the two random effects are assumed to be correlated and that their joint covariance matrix is to be estimated[1].
   Again, a qualifier **DIAG** or **NOSTART**, as described above for the residual covariance matrix (see Section 4.9.1.1) is recognized.

Correlated random effects

2. The $q(q+1)/2$ elements of the *upper* triangle of the covariance

[1] Currently implemented for full-rank, 'standard' analyses only !

matrix, given row-wise (i.e. σ_1^2, σ_{12}, ..., σ_{1q}, σ_2^2, σ_{23}, ..., σ_q^2).
Again, these can be given several elements per line (space separated, taking care not to exceed the total line length of 78 characters), or one per line, or a mixture – WOMBAT will attempt to read until it has acquired all elements required.
As above, if **DIAG** is specified, only the q variances are read and with **NOSTART** no values are acquired.

4.10 Special information

Finally, the parameter file can select some special features.

4.10.1 Dependencies among fixed effects

WOMBAT requires a coefficient matrix in the mixed model equations which is of full rank. Hence, constraints must be placed on the fixed effects part of the model if more than one fixed effect is fitted. By default, the first level of each cross-classified fixed effect other than the first is 'zeroed out' for each trait to account for dependencies. If there are additional dependencies, these should be identified and specified explicitly prior to each analysis.

WOMBAT performs a simple least-squares analysis on the fixed effects part of the model, attempting to find such additional dependencies. However, this procedure should not be relied upon, in particular for large data sets where numerical errors tend to accumulate sufficiently to obscure identification. Dependencies not properly taken into account can lead to problems during estimation !

ZEROUT

Additional effects to be zeroed out are specified in a block entry. The block begins with a line containing the code **ZEROUT** (can be abbreviated to **ZER**), and finishes with a line beginning with **END**. The block then should contain one line for each additional dependency. Each line should contain three entries :

(a) The name of the fixed effect, as specified in the **MODEL** block.
(b) The 'original' code for the level to be zeroed out, as encountered in the data file.
(c) The trait number; this can be omitted for univariate analyses.

4.10.2 Random effects to be treated as fixed

In some instances, it is desirable to treat selected levels of a random, *genetic* effect as if it were 'fixed'. A typical example is the analysis of dairy data under a sire model. If the data includes highly selected

proven bulls which only have records from their second crop of daughters, we might want to treat these bulls as 'fixed' as we would expect estimates of the genetic variance to be biased downwards otherwise. In other cases, our pedigree records may include codes for 'parents' which are not animals but represent genetic groups, to be treated as fixed effects.

Genetic groups

Treating selected random genetic effects levels as fixed is achieved by replacing the diagonal in the numerator relationship matrix by a value of zero when setting up the inverse of the numerator relationship matrix. If there is any pedigree information for the effect, this is ignored.

N.B.
Treating levels of random effect(s) as fixed may lead to additional dependencies in the fixed effects part of the model, especially for multivariate analyses. Great care must be taken to identify these and specify them explicitly, as outlined above (Section 4.10.1). WOMBAT has no provision to account for this possibility !

PSEUDOFX

Random genetic effects levels to treated as fixed are specified in a block entry. The block begins with a line containing the code **PSEUDOFX** (can be abbreviated to **PSE**). This can be followed (space separated) by a real number. If given, this value (which should be a small positive number, e.g. 0.0001 or 0.001) is used instead of the value of 0 when specifying the contributions to the inverse of the numerator relationship matrix. Effectively, this treats the respective effect level as "just a little bit random". This option is provided as a mean to counteract additional dependencies in the model (see warning above). It is particularly useful for prediction, and should be used very carefully with estimation runs. As usual, the block finishes with a line beginning with **END**. The block then should contain one line for each effect. Each line should contain two entries :

(a) The name of the random effect, as specified in the **MODEL** block. This should be a genetic effect, distributed proportionally to the numerator relationship matrix among animals.
(b) The 'original' code for the level to be treated as 'fixed', as given in the pedigree file.

N.B.
This option has only undergone fairly rudimentary testing !
It is not available in conjunction with the PX-EM algorithm.
Selecting this option prohibits re-use of inverse NRM matrices set up in any previous runs.

4.10.3 Covariance components to be treated as fixed

FIXVAR

WOMBAT provides limited facilities to fix certain covariance components at their input values, while maximising the (log) likelihood with respect to the remaining parameters. Again, this information is given in a block entry. The block begins with a line containing the code **FIXVAR** (no abbreviation) and, as usual, ends with a line containing the code **END**. Each line within the block is then expected to have the following entries:

(a) The name of the random effect, as specified in the **MODEL** block.
(b) The code **ALL** to signify that all pertaining covariances are fixed.

In the moment, no options to fix individual elements of covariance matrices are recognised.

4.10.4 Additional functions of covariance components

SE+USR

In addition, users can define other functions of covariance components which should be calculated and for which sampling errors should be approximated. This is done in a block entry, beginning with a line containing the code **SE+USR** (can be abbreviated to **SE+**), and ending with a line beginning with **END**. The block should then contain one line for each function to be calculated. The content of the line depends on the type of function. Three types are recognised

1. Linear combinations (weighted sums) of the covariance components in the model of analysis. For these, space separated entries o should be
   (a) The code **SUM** at the beginning of the line.
   (b) A name to identify the sum to be calculated.
   (c) An entry of the form $n(w)$ for each component of the weighted sum, with n the running number of the covariance component and w the weight it is to be multiplied with. If w is unity, it can be omitted, i.e. the entry n is interpreted as $n(1)$.
2. Ratios of two covariance components. For these, the line should have three entries
   (a) The code **VRA** at the beginning of the line.
   (b) A name for the variance ratio to be calculated.
   (c) The running number of the covariance component in the numerator.
   (d) The running number of the covariance component in the denominator.
3. Correlations, i.e. the ratio of a covariance component and the square root of the product of two other covariances. Line entries for these are

(a) The code **COR** at the beginning of the line.
(b) A name to identify the correlation to be calculated.
(c) The running number of the covariance component in the numerator.
(d) The running number of the first covariance component in the denominator.
(e) The running number of the second covariance component in the denominator.

Example

```
SE+USR
  SUM  siga+pe  1  2
  VRA  repeat   5  4
END
```

Consider a univariate analysis with repeated records per individual. Fitting additive genetic and permanent environmental effects of the animal, variance components in the model are σ^2_A, σ^2_{PE} and σ^2_E (residual), with running numbers 1, 2 and 3, respectively. WOMBAT automatically calculates the phenotypic variance $\sigma^2_P = \sigma^2_A + \sigma^2_{PE} + \sigma^2_E$ and gives it running number 4. To calculate the repeatability and approximate its sampling error, we first need to define the sum of σ^2_A and σ^2_{PE} as a new covariance (which receives running number 5), and then define the repeatability as the ratio of this component and σ^2_P.

Hint

Run WOMBAT with the **--setup** option to begin with. Inspect the file **ListOfCovs** generated in this step – this gives a list of running numbers for individual covariance components.

4.10.5 Pooling estimates of covariance components

POOL

Information related to pooling of covariance components using a (penalized) maximum likelihood approach is specified in a block entry, beginning with a line containing the code **POOL** and ending with a line beginning with **END**. Within the block, the following directives are recognized:

PSEUPED : followed (space separated) by a three letter code specifying the assumed pedigree structure and values on sizes or numbers of families. The following pseudo-pedigree codes are available:

PHS : denotes a simple balanced paternal half-sib design. Optionally, two integer numbers giving the numbers of sires and the number of progeny per sire, respectively, can be given

(space separated, on the same line) after the code. If not given, default values of 10 and 4 are used.
This option is suitable for a simple animal model only. WOMBAT will check that a) there is only one random effect fitted, and b) that this has covariance option NRM. If the **MINPAR** option (see below) is used, WOMBAT cannot perform these checks; hence the **DIRADD** code together with the name of the random effect, as described below, need to be given.

HFS : implies a balanced hierarchical full-sib design comprised of s sires, d dams per sire and n progeny per dam. Assumed values for s, d and n can be given (space-separated) on the same line. If omitted, default values of $s = 10$, $d = 5$ and $n = 4$ are used. This option is suitable for a simple animal model or a model fitting maternal permanent environmental effects in addition. Again, if the **MINPAR** option is used, codes **DIRADD** and **MATPE** need to be given in addition.

BON : selects a design comprising 8 individual per family in two generations, due to Bondari et al. [4]. Optionally, this code can be followed (space separated) by the number of such families (integer); if not given, a default value of 2 is used. For this design, expectations of covariances between relatives due to direct and maternal effects are available. In order for WOMBAT to 'know' which random effect has which structure, additional information is required. This should comprise one additional line per random effect fitted, with each line consisting of a keyword specifying the type of random effect followed (space separated) by the name of the effect as specified in the model of analysis. Keywords recognized are **DIRADD** for direct additive genetic, **MATADD** for maternal genetic and **MATPE** for maternal permanent environmental effects.

Example

```
MODEL
   RAN      animal  NRM
   RAN      gmdam   NRM
   RAN      pedam   IDE
   trait    bwgt    1
   trait    wwgt    2
   trait    ywgt    3
END MOD
VAR animal   3  NOS
VAR gmdam    3  NOS
VAR pedam    3  NOS
VAR residual 3  NOS
POOL
   PSEUPED     BON  10
      DIRADD   animal
      MATADD   gmdam
      MATPE    pedam
   SMALL       0.0010
   DELTAL      0.0010
END
```

USR : Other designs and models are readily accommodated by specifying the expectations of covariances between family members for each random effect fitted (excluding residuals - this is set automatically to be an identity matrix). This is selected by the **USR** option which must be followed (space separated) by an integer specifying the family size. Optionally, a second number giving the number of families is recognized (a default value of 2 is used if not given). Then the *upper* triangle of the matrix of coefficients in the expectation of covariances has to be given for each random effect fitted. For multiple random effects, these have to be given in the *same* order in which they have been specified in the **VAR** statements in parameter file, and the matrix for each effect has to begin on a new line.

Example

```
MODEL
  RAN  animal  NRM
    ...
END MOD
VAR   animal   4  NOSTART
VAR   residual 4  NOSTART
POOL
   PSEUPED  USR  5
   1.0  0.50  0.50  0.50  0.50
   1.0  0.25  0.25  0.25
   1.0  0.25  0.25
   1.0  0.25
   1.0
END
```

This shows the coefficients for direct additive genetic effects for a family comprising a sire (individual 1) with four progeny from unrelated dams.

SMALL : followed (space separated) by a real number which gives the lower limit (≤ 1) for smallest eigenvalue allowed in the combined matrices. If this is not specified, a default value of 0.0001 is used.

DELTAL : followed (space separated) by a real number specifying the convergence criterion: if the increase in log likelihood between iterates falls below this value, convergence is assumed to have been achieved. If not specified a stringent default of 0.00005 is used.

MINPAR : The default assumption for the parameter file used is that it is a 'full' parameter file as for a corresponding, multivariate analysis. However, the information on pedigree and data file and their layout are not actually required. Hence the option **MINPAR** (on a line by itself) is provided which allows use of a 'minimum' parameter file, switching off some of the consistency checks otherwise carried out. If used, **MINPAR** must be the first entry within the **POOL** block.

The minimum information to be given in the parameter file must comprise:

1. The **ANAL**ysis statement
2. A **VAR** line for each covariance matrix, together with the **NOSTART** option telling WOMBAT not to expect a matrix of starting values.
3. The **POOL** block, including statements showing which random effect represents which type of genetic or non-genetic

effect.

Example

```
ANAL MUV 14
VAR  animal    14  NOSTART
VAR  residual  14  NOSTART
POOL
   MINPAR
   SMALL  0.001d0
   PSEUPED  hfs  100  10  4
     DIRADD  animal
END
```

SINGLE : The default form of input for results from part analysis is to read estimates from separate files, in the form of output generated by WOMBAT when carrying out multiple analyses of subsets of traits. Alternatively, all information can be given in a single file. This is selected by the option **SINGLE**, followed (space separated) by the name of the input file (same line). The layout required for this file is described in Section 6.5.4.2.

PENALTY : followed (space separated) by a code word(s) defining the type and strength of penalty to be applied. Codes for the penalty type recognised are:

CANEIG : selects a penalty on the canonical eigenvalues. This has to be followed (space separated) by either **ORG** or **LOG** specifying a penalty on eigenvalues on the original scale or transformed to logarithmic scale (no default).

COVARM : specifies shrinkage of a covariance matrix towards a given target.

CORREL : chooses shrinkage of a correlation matrix towards a given target correlation matrix.

Either **COVARM** or **CORREL** can be followed (space separated) by the keyword **MAKETAR**. If this is given, WOMBAT determines the shrinkage target as the phenotypic covariance (or correlation) matrix obtained by summing estimates of covariances for all sources of variation from the preceding, unpenalized analysis. If this is not given, the upper triangle of the target matrix is expected to be read from a file with the standard name **PenTargetMatrix**; see Section 6.5.7.1.

The last entry on the line relates to the tuning factor(s) to be used: If a single penalized analysis is to be carried out, the corresponding tuning factor should be given (real value). To specify multiple penalized analyses, specify the number of separate tuning factors

multiplied by -1 as an integer value (e.g. -3 means 3 analyses), and list the corresponding tuning factors space separated on the next line.

Example

1. Shrink all correlation matrices towards the phenotypic correlation matrix using a single tuning factor of 0.1, and calculate the shrinkage target from unpenalized results.

```
PENALTY CORREL  MAKETAR  0.1
```

2. Shrink canonical eigenvalues on the logarithm scale towards their mean, using 5 different tuning factors

```
PENALTY CANEIG LOG  -5
0.01  0.1  0.5  1.0  2.0
```

4.10.6 Miscellaneous

A place to collect miscellaneous options which don't fit into the categories above is provided through a block entry beginning with a line **SPECIAL**, and ending with a line **END**. Options are available to address the following problems:

4.10.6.1 Weighted analysis

A weighted analysis is specified by a line with the following space separated entries:

(a) The code **WEIGHT** at the beginning of the line.
(b) The name of the column in the data containing the weight.
(c) Optionally, for standard uni- and multivariate analyses, this can be followed by a code to select the type of weighting to apply:

none : is the default: each record is multiplied by the weight given prior to the analysis.

DESIGN : causes entries of "1" in the design matrices to be replaced with the weight specified (observations are not scaled).

RESID : multiplies the diagonal elements of the residual covariance matrix pertaining to the observation with the weight given; in addition, any non-zero elements for a pair of records (on the same individual) are scaled with the geometric mean of the corresponding pair of weights (i.e. $\sqrt{w_i w_j}$).

Example

Consider an animal with two records and corresponding weights $w_1 = 0.8$ and $w_2 = 2$. Let the unweighted residual covariance matrix be

$$\begin{pmatrix} 10 & -5 \\ -5 & 20 \end{pmatrix} \quad \text{with inverse} \quad \begin{pmatrix} 0.1143 & 0.0286 \\ 0.0286 & 0.0571 \end{pmatrix}$$

The weighted matrix is then

$$\begin{pmatrix} 8 & -6.325 \\ -6.325 & 40 \end{pmatrix} \quad \text{with inverse} \quad \begin{pmatrix} 0.14286 & 0.02259 \\ 0.02259 & 0.02857 \end{pmatrix}$$

4.10.6.2 Covariables that are zero

WOMBAT now checks for covariables which have a value of zero. If you fit a covariable which has such values, it expects you to confirm that this is intentional. This is done through a single line with space separated entries:

(a) The code **COVZER** at the beginning of the line.
(b) The full name of the covariable as given in the model statement (i.e. including brackets, number of coefficients and, if applicable, type).
(c) A three letter option: **FIT** to confirm that a value of zero for this covariable is a valid code, or **IGN** to skip such records (any other option will lead to a programmed error stop).

4.10.6.3 Animals that are clones

WOMBAT will stop if it encounters non-consecutive records for the same subject in different parts of the data file, assuming that the data has not been sorted as required. However, there are situations where this is a 'planned' scenario, for example to fit data on clones, where we want the same genetic effect but different residual effects for members of a clone. Hence, this check can be switched off by including a line with the code **CLONES** in the **SPECIAL** block.

4.10.6.4 Standard errors for fixed and random effect solutions

WOMBAT reports solutions for fixed and random effects fitted – these are calculated as a by-product by all algorithms to obtain REML estimates of variance components. However, standard deviations/errors are only reported for those algorithms which invert the coefficient matrix in the mixed model equations. An option is provided to enforce

this – this is invoked by a line (in a **SPECIAL** block) with the code **FORCE-SE**. If standard errors have been obtained automatically, this has no effect. Otherwise, it forces calculation of the inverse of the coefficient matrix at convergence.

N.B.

For large analyses, this can take quite some extra time.

4.10.6.5 Repeated records

WOMBAT attempts to check the data for records representing repeated measurements for a trait and individual, and stops when this is fairly low. This can be overridden to some extent: Analyses with repeated records for a trait for less than 20% and more than 10% of individuals can be enabled by a single line with the code **REPEAT** at the beginning of the line.

Standard multivariate analyses (**MUV**) with repeated records can represent a somewhat difficult scenario, as proper modelling of the residual covariance structure relies on differentiating between records for different traits are taken simultaneously and thus should be assumed to be subject to a common,*temporary* environmental (residual) covariance, and which should not. There are four strategies which can be chosen by specifying a single line with the following space separated entries:

(a) The code **RPTCOV** at the beginning of the line.
(b) A seven-letter qualifier to select he strategy to be used. Valid terms are:
   INDESCR : to fit a residual covariance $\sigma_{E\,ij}$ between all observations for traits i and j for an individual (this was the previous default).
   ALIGNED : to fit a residual covariance $\sigma_{E\,ij}$ only between obervations with the same running running number, e.g. the first record for trait i and the first record for trait j (this is the *current* default).
   ZEROALL : for the case where all temporary environmental covariances $\sigma_{E\,ij}$ are assumed to be zero. This should be accompanied by corresponding starting values for the residual covariance components.
   TSELECT : when none of the other options are appropriate *and* where the data file has a column with a 'time of recording' indicator (integer). The name of this column must be given on the same line, space-separated after **TSELECT**. This allows for

proper identification of records for different traits taken at the same time – and thus assumed to have a non-zero, temporary environmental covariance – and those which are not when there is an arbitrary pattern of missing records.

Example

```
SPECIAL
   RPTCOV TSELECT  mtime
END
```

Specification of this option is required for multivariate analyses combining traits with single and repeated records.

4.10.6.6 Fitting SNP effects

When using the run option **--snap**, the covariable in the model representing SNP effects needs to be identified by a line in a **SPECIAL** block containing

(a) The code **QTLEFF** at the beginning of the line.
(b) Space separated, the full name of the covariable, as specified in the **MODEL** block (i.e. including brackets and the order of it).
(c) Optional, for analyses estimating more than one regression coefficient: Space separated, the code **COVOUT** to specify that the *u*pper triangle of the covariance matrix of regression coefficients is written out.

In addition, a **COVZER** statement may be required to ensure WOMBAT treats count of zero as valid covariable values.

For 'house-keeping' reasons, WOMBAT requires that these covariables are specified (in the **MODEL** block) *before* any other, 'regular' covariables to be fitted.

Example

```
SPECIAL
   QTLEFF  snp(1)
   COVZER  snp(1)  FIT
END
```

4.10.6.7 Sampling to approximate standard errors

The default in drawing samples from distribution of estimates of covariance matrices is to sample the parameters estimated, i.e. the elements of the Cholesky factor of the matrix of interest, and to construct the matrix samples from these. This yields samples within the parameter

space. However, it can also yield substantial differences in the mean of samples and their variances from the 'input' values. Hence an option is provided to select sampling of the covariance matrices instead. This will yield means close to the REML estimates and estimates of sampling variances closer to those derived from the inverse of the average information matrix, but is likely to yield samples out of the parameter space.

To select a single covariance matrix to be sampled when approximating sampling variances (see Section 5.2.9) requires a single line in a changed **SPECIAL** block with at least two entries:

(a) The code **SAMPLEAI** at the beginning of the line.
(b) The name of the random effect as specified in the **MODEL** block (space separated).
(c) Optionally, this can be followed (space separated) by the code **COVAR** to select sampling of covariance matrices.
(d) Optionally again, **COVAR** can be followed (space separated) by the code **TRUNC** which invokes modification of samples with eigenvalues close to zero (truncating at 10^{-4}) to ensure the sample has the rank specified.

Example

```
SPECIAL                                                    1
   SAMPLEAI  animal                                        2
END                                                        3
```

4.10.6.8 Fitting "social" genetic effects

To fit a model with an additional random effect representing individuals' competitive genetic effects, a random regression model analysis has to be selected – fitting random effects at an order of 1 with a basis function of "1" (option **ONE**) yields an equivalent model to a standard, univariate analysis whilst providing the flexibility to model residual variances as heterogeneous. WOMBAT then requires a "social" group code to be supplied and collects the identities of the individuals in each group as part of its set-up steps.

Additional information required is to be supplied by a single line in the SPECIAL block. This should contain the following, space separated entries:

(a) The code SOCIAL at the start of the line.
(b) The name of the random effect representing the "social" genetic effect. This can be abbreviated by omitting the part in brackets

specifying the order of fit and basis function (e.g. `(1,ONE)`).

(c) The name of the fixed or extra effect representing the "social" group.

(d) An Integer variable giving the maximum group size.

(e) Optional: A Real variable giving the dilution factor d to be used. If omitted, a default value of $d = 0$ (i.e. no dilution) is used.

(f) Optional: If a non-zero dilution factor has been given, an additional, four-letter code is recognised which specifies the "target group size" for which competitive genetic variance components are to be estimated. Codes available are `NTWO` which targets a group size of 2 (by using an entry of $[1/(n-1)]^d$ in the corresponding design matrix) and `NBAR` which targets the average group size (using coefficients $[(\bar{n}-1)/(n-1)]^d$ in the design matrix; see Bijma [3]). If omitted, the default used is `NTWO`.

For each run where a dilution factor has been specified, WOMBAT appends the value of the dilution factor used together with the corresponding maximum log likelihood to a file in the working directory named `LogL4Quapprox.dat` (see Section 7.3.8). These represent points on the profile likelihood curve for the dilution factor, and a quadratic approximation can be used to determine its maximum. The run time option **`--quapp`** is provided to assist in this task.

4.10.6.9 Fitting 'explicit' genetic groups

see example 18

4.10.6.10 In core storage

The default for WOMBAT is to repeatedly read data and relationship information from disk so as to minimize memory requirements. In-core storage of selected information can be specified with a line in a `SPECIAL` block. This should begin with the code **`INCORE`** followed (space separated) by one or more qualifier(s) choosing which information is to be stored in core. Valid qualifiers are **`NRM`** (inverse of numerator relationship matrix), **`GIN`** (`*.gin` matrix), **`DATA`** (recoded data), **`RAW`** (raw data) and **`ALL`** (all four).

Example

```
SPECIAL
      ...
   INCORE GIN DATA
      ...
END
```

4.10.6.11 Calculation of alternative outlier model statistics

Calculation of alternative outlier model (AOM) statistics for residuals or random effects fitted is invoked by specifying the option **AOM-RES** or **AOM-RND** by itself on a line in a **SPECIAL** block in the parameter file.

Example

```
SPECIAL                                                    1
        ...                                                2
   AOM-RES                                                 3
        ...                                                4
END                                                        5
```

The AOM options are implemented for uni- and full rank standard multivariate analyses only; they are ignored for reduced rank (PC) analyses and random regression models. Calculations are carried out for one observation or random effects level at a time. For large models or cases with dense coefficient matrices **C**, this can take quite a long time.

AOM statistics for residual effects, $\mathbf{R}^{-1}\hat{\mathbf{e}}/\sqrt{\mathrm{Dia}(\mathbf{R}^{-1}-\mathbf{R}^{-1}\mathbf{W}\mathbf{C}^{-1}\mathbf{W}'\mathbf{R}^{-1})}$, are reported together with residuals in **Residuals.dat**. This file has one row per record with up to seven columns. Columns 1 to 3 are written out for all analyses and contain the residual, the predicted observation and the actual observation (for reference). Specifying **AOM-RES** adds the diagonal element of the projection matrix (p_{ii}; column 4), the respective element of $\mathbf{R}^{-1}\hat{\mathbf{e}}$ (the nominator of the AOM statistic; column 5) and the AOM statistic (column 6). For univariate analyses, the corresponding diagonal element of the "Hat" matrix ($\sigma^2[1-\sigma^2 p_{ii}]$ with σ^2 the residual variance) is given in addition (column 7).

AOM statistics for random effects, $\mathbf{G}^{-1}\hat{\mathbf{u}}/\sqrt{\mathrm{Dia}(\mathbf{G}^{-1}-\mathbf{G}^{-1}\mathbf{C}^{ZZ}\mathbf{G}^{-1})}$, are reported alongside the corresponding solutions in the **RnSoln_*.dat** files. This file has up to 10 columns. If **AOM-RND** has been specified, the last three columns give the diagonal element of the covariance matrix (`T_ii`), the element of $\mathbf{G}^{-1}\hat{\mathbf{u}}$ (`(G{-1}u)_i`) and the AOM statistic, respectively. Where the denominator is essentially zero (e.g. for additive genetic effects for parent without records), the AOM statistics is set to zero.

4.10.6.12 Analyses calculating and using a Gaussian Kernel matrix

Kernel based prediction using genomic information is attracting increasing interest. This can be parameterised as

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-d_{ij}^2/\theta^2)$$

To facilitate estimation of the bandwidth parameter, WOMBAT has been adapted to calculate the inverse of the kernel matrix for a chosen value of θ, with d_{ij} the 'distance' for individuals i and j.

To fit a Gaussian kernel matrix as 'relationship matrix', the random effect which this matrix pertains to needs to be specified with the covariance option **GIN**. The corresponding ***.gin** file should follow the same format as for other effects with such covariance structure, *except* that the elements given should be the squared distances, d_{ij}^2 and that the diagonals of 0 can be omitted. Only rudimentary checks of the input values are performed, e.g. WOMBAT stops if a zero distance between a pair of individuals is found, as this will result in a non-positive definite matrix.

The bandwidth parameter to be used needs to be specified in a **SPECIAL** block. This should contain a line starting with the code **KERNEL** followed (space-separated) by the name of the random effect as given in the **MODEL** block, and the value of θ. E.g. for random effect **animal** and $\theta = 1.5$

Example

```
SPECIAL
       ...
   KERNEL animal   1.5
       ...
END
```

WOMBAT will then construct the kernel matrix for the given value of θ and attempt to invert it, as well as calculate its log determinant. **K** is assumed to be dense and is automatically stored 'in-core'. Inversion is carried out using LAPACK [2] routine **DPFTRF** [14] which requires a positive definite matrix.

On completion of the estimation step for a given value of θ, it is written to the file **LogL4Quapprox.dat** together with the corresponding maximum log likelihood. To determine the next value on the profile likelihood for θ to be evaluated, WOMBAT can be run with the command line option **--quapp** (see Section 5.2.11) which performs a quadratic approximation of the profile log likelihood for θ.

4.10.6.13 Writing out correlations for random regression analyses

By default, WOMBAT calculates and writes out a small number of correlations on the 'original scale' for the estimated covariance functions. These are primarily given to assist with checking your own calculations to obtain the correlations you are interested in - covariances and correlations are easily calculated using the file of evaluated basis functions provided.

The complete correlation matrix is not provided by default as this might result in big output files and many calculations needed. However, if you really want correlations for all values of the control value, you can specify this by a single line in a SPECIAL block containing the code RRCORR-ALL. If the analysis fits a single control variable, this writes out not only **RanRegCorrels.dat**, but also writes an additional file **RanRegCorrAll.dat** (see Section 7.2.4) with the same information in a format more convenient for further processing, e.g. plotting.

4.10.7 Penalized estimation

WOMBAT can carry out estimation imposing a penalty on the log likelihood function to improve the sampling properties of estimates. Three types of penalties are accommodated:

'Bending'

1. A penalty regressing the so-called canonical eigenvalues towards their mean, as described by Meyer and Kirkpatrick [29], also referred to as 'bending'. This can be done for the eigenvalues directly or after transformation to logarithmic scale. In addition, there is a choice of parameterisation.
   As the canonical transformation involves only two matrices, this type of penalty is restricted to multivariate analyses fitting a simple animal model or random regression analyses fitting animals' genetic and permanent environmental effects only.

Matrix shrinkage

2. A penalty proportional to the matrix divergence between a covariance matrix to be estimated and a specified target matrix. This will shrink the estimate towards the target. Options are available to shrink a covariance matrix or a corresponding correlation matrix. A particular choice of target is the phenotypic covariance or correlation matrix estimated in a standard (unpenalized) analysis [see 30] – hence WOMBAT provides the facility to write out such target matrices. While this could, in principle, be applied to multiple matrices, the current implementation is restricted to a single matrix.

NEW

3. A penalty on a correlation matrix, shrinking the corresponding partial auto-correlations towards zero or their phenotypic counter-

parts. The penalty is derived assuming a (shifted) Beta distribution and its stringency is regulated through choice of the so-called prior effective sample size.

Penalized estimation is invoked by one or more lines in a **SPECIAL** block. It generates several additional output files, described in Section 7.2.11.

4.10.7.1 Selecting a penalty

Penalized estimation is specified by a line with space separated entries:

(a) The code **PENALTY** at the beginning of the line.
(b) A code specifying what type of penalty is to be applied. Values recognised are:
CANEIG : selects a penalty on the canonical eigenvalues.
COVARM : specifies shrinkage of a covariance matrix.
CORREL : chooses shrinkage of a correlation matrix towards a given target correlation matrix.
KANEIG : selects a penalty on the canonical eigenvalues, obtained assuming a Beta distribution.
KORREL : selects a penalty on individual correlations, assumed to have a Beta distribution.
PACORR : chooses shrinkage of a the matrix of partial auto-correlations for a random effect towards the corresponding phenotypic or an identity matrix.

The following variables depend on the form of penalty chosen.

For **CANEIG**:

(c) A character variable to specify the 'scale' to be used. Options available are:
ORG : selects a penalty on the canonical eigenvalues on the original scale.
LOG : specifies a penalty on the canonical eigenvalues after transformation to logarithmic scale.
LOGB : is similar to **LOG** but a penalty is applied to both $\log(\lambda_i)$ and $\log(1-\lambda_i)$, where λ_i denotes the $i-$th canonical eigenvalue.
(d) A character variable to specify the parametrisation of λ_i to be used. Options available are:
ORG : estimate λ_i directly.
LOG : estimate $\log(\lambda_i)$, i.e. λ_i transformed to logarithmic scale.
TIC : estimate $\log(\lambda_i/(1-\lambda_i))$, i.e. λ_i transformed to logistic scale.
This variable is optional; if omitted, it is set to the same value as that given for the scale of the penalty (**LOG** for **LOGB**).

(e) A real or integer variable specifying either the tuning factor ψ to be used, or, if estimates for multiple tuning factors are to be obtained, the number of ψ's to be read subsequently or the number of different ranges of ψ's to be stepped through:

- If a real value is given, $\mathbb{WOMBAT}$ interprets this as the (single) tuning factor to be used, and expects no further input.
  N.B. This value must contain a decimal point to be recognised as such!

  Example

  ```
  SPECIAL                                          1
     PENALTY  CANEIG  LOG 5.0                     2
  END                                              3
  ```

  specifies a penalty on the canonical eigenvalues transformed to logarithmic scale for a tuning factor of $\psi = 5.0$.

- If a positive integer is found, this is interpreted as the number of different values for ψ to be used. These are to be read in a space-separated list, beginning on the *next* line.

  Example

  ```
  SPECIAL                                          1
     PENALTY  CANEIG  ORG  6                       2
     0.5  1.0  2.5  5.2  8.7  11.0                 3
  END                                              4
  ```

- If a negative integer, $-n$ is specified, this is interpreted as the number of ranges of ψ to be considered. These are expected to be found on the following n lines, with one triplet of space-separated real values per line, consisting of starting value, end value and step size.

  Example

  ```
  SPECIAL                                          1
     PENALTY  CANEIG  LOG TIC -2                   2
     0.0  3.0  0.5                                 3
     5.0  8.0  1.0                                 4
  END                                              5
  ```

- Finally, a value of 0 (integer) specifies that the values of ψ are to be read form a separate file. In this case, the name of this file needs to be given (space-separated from 0) on the same line. This file should give the tuning factors to be used, with each value given on a separate line.

All values of ψ given should be in ascending order.

For **COVARM** and **CORREL**:

(c) An optional character variable:

PHENV : This option specifies that the estimate of the phenotypic covariance or correlation matrix is to be written out to the file **PenTargetMatrix**.
It is best used in a preliminary step with a single $\psi = 0$ to generate the matrix to be used in subsequent penalized runs where we wish to shrink towards this matrix.
The default is for this file to exist and to be read.

Example

```
SPECIAL
   PENALTY  COVARM PHENV  animal  0.0
END
```

KLDIV : The default penalty on a matrix of size $q \times q$ is

$$\mathcal{P} \propto C \log |\mathbf{\Sigma}| + \mathrm{tr}\left(\mathbf{\Sigma}^{-1}\mathbf{T}\right)$$

with $\mathbf{T}$ the shrinkage target and $C = (\psi + q + 1)/\psi$. The latter value originates from the assumption of an Inverse Wishart distribution for $\mathbf{\Sigma}$. This option sets $C = 1$, i.e. applies a penalty proportional to the Kullback-Leibler divergence between $\mathbf{\Sigma}$ and $\mathbf{T}$.

Example

```
SPECIAL
   PENALTY  COVARM  KLDIV  animal 7.50
END
```

(d) The name of the random effect covariance matrix to be penalized. This must match the name given in a **VARIANCE** statement (see Section 4.9) earlier in the parameter file.

Example

```
SPECIAL
   PENALTY  CORREL  animal -1
   0.0 5.0 1.0
END
```

(e) Information on tuning parameter(s) to be used, as for **CANEIG** (see above).

NEW For **KANEIG**: see example 19 For **KORREL**: see example 19 For **PACORR**:

(c) An optional character variable:

PHENV : This specifies that the shrinkage target is the phenotypic matrix of partial auto-correlations

IDENT : This specifies that the shrinkage target is the identity

matrix

If omitted, the default shrinkage target is the identity matrix.

(d) The name of the random effect covariance matrix to be penalized. This must match the name given in a **VARIANCE** statement (see Section 4.9) earlier in the parameter file.

(e) A real value specifying the 'effective sample size' of the Beta prior (no multiple values allowed for this form of penalty).

Example

```
SPECIAL
   PENALTY PACORR  animal  4.0
END
```

4.10.7.2 Stopping after a given change in likelihood

If multiple tuning factors are given, it is sometimes desirable to stop once the deviation in the unpenalized log likelihood from the maximum (at $\psi = 0$) has exceeded a certain limit. This can be specified by an additional line with three space-separated entries:

(a) The code **PENALTY** at the beginning of the line.

(b) The code **LIMIT**.

(c) A real variable with the value for the limit.

Example

```
SPECIAL
   PENALTY  CORREL  KLDIV  animal -2
   0.0  5.0   0.5
   6.0  15.0  1.0
   PENALTY  LIMIT  3.315
END
```

For this option to work, the first tuning factor given has to be $\psi = 0$!

Hint

Use run time option **---valid** (Section 5.3.2) for validation steps when using cross-validation to estimate the value for ψ to be used.

5 Run options

5.1 Overview

Running WOMBAT can be as simple as specifying its name at command level, i.e.

Example

```
wombat
```

1

A number of options are available, however, to modify the run time behaviour of WOMBAT. These range from simple options to set the verbosity level of screen output or to select a continuation run, to highly specialised options to modify the search strategies for the maximum

of the likelihood function. Multiple options can be combined, but some care is required so that options given last do not unintentionally cancel out some of the effects of options given first. Options available are summarised in Table 5.1, Table 5.2 and Table 5.3.

Run options can be given in three ways :

1. On the command line, e.g. **wombat -v -c myparfile.par**
2. In a file with the default name **RunOptions** in the current working directory.
   This file must contain a single option per line. If such file exists, it is read before any command line options are parsed. There are no checks for options specified in duplicate. This implies that the last setting for a particular option is the one to be used, and that the options in **RunOptions** can be overridden by command line options.
3. On the first line of the parameter file, after a code of **RUNOP** (see Section 4.3). Any such options are read before options specified on the command line, i.e. the latter may overide them.

Form

Following Linux standard, run options have the form "**-a**" where **a** stands for a single, lower-case letter, or the form "**---abcdef**" where **abcdef** stands for a multi-letter code.

5.2 Basic run options

5.2.1 Continuation run

-c

In some instances, REML estimation **c**ontinuing from the 'best' estimates so far is necessary. This is invoked with the **-c** option.
If specified, WOMBAT will attempt to read its 'starting' values from the file **BestPoint** in the current working directory. *N.B.* If this is not available or if a read error occurs, the run proceeds as a 'new' run, i.e. using the starting values given in the parameter file instead. The **-c** option selects estimation using the AI algorithm, unless another procedure is selected explicitly.

5.2.2 Level of screen output

The amount of screen output can be regulated by the following options

-t : selects **t**erse output
-v : selects **v**erbose output, useful to check the parameter file
-d : selects **d**etailed output, useful for debugging

Table 5.1: 'Basic' run options for WOMBAT

Option	Purpose
-c	Specify a **c**ontinuation run
-v	Specify **v**erbose screen output
-d	Specify very **d**etailed screen output
-t	Specify **t**erse screen output
Default REML algorithms	
--good	Tell that you have **good** starting values
--bad	Tell that you have **bad** starting values
Non-estimation runs	
--setup	Select a run performing the **set-up** steps only
--best	Select a run printing out estimates for the currently **best** point only
--blup	Select a prediction (**BLUP**) run; direct solution
--solvit	Select a prediction (BLUP) run; **solv**e **it**eratively
--mmeout	Like **--solvit**, but write **MME out** to file.
--s1step	Like **--solvit**, but for **s**ingle **step** model.
--snap	Carry out a GWAS type analysis for multiple SNPs.
--simul	Select a run **simul**ating data only
--sample	Select a run **sampl**ing estimates of covariance matrices
--subset	Write out parameter files for analyses of **subset**s of traits
--itsum	**It**erative **sum**mation of partial covariance matrices
--pool	**Pool**ing covariance components by penalized maximum likelihood
--invert	**Invert** a dense symmetric matrix; pivot on largest diagonal
--invrev	**Inv**ert a dense symmetric matrix; **rev**erse pivoting on no. of off-diagonal elements
--inveig	**Inv**ert a dense symmetric matrix; use **eig**en decomposition
--invlap	**Inv**ert a dense symmetric p.d. matrix; use **LAP**ACK routines
--invspa	**Inv**ert a **spa**rse symmetric matrix
--quapp	Perform **qu**adratic **app**roximation of likelihood
Matrix storage	
--dense	Specify **dense** mixed model equations.
Miscellaneous	
--expiry	Print out the expiry date for the program (to screen)
--limits	Print out the hard-coded program limits (to screen)
--times	Print out various intermediate times for a run
--wide	Write 'wide' output files

5.2.3 Set-up steps

--setup It is good practice, to start each analysis with a run which carries out the set-up steps only, checking that the summary information provided on the model of analysis and data structure in file **SumModel.out**

(see Section 7.1.2) is as expected, i.e. that WOMBAT is fitting the correct model and has read the data file correctly. This is specified using **--setup**. This option also invokes verbose screen output.

5.2.4 Quality of starting values

For analyses comprising 18 or less covariance components to be estimated, WOMBAT defaults to the AI algorithm for maximisation. For other analyses, the default is for WOMBAT to begin with up to 3 iterates of the PX-EM algorithm, before switching to an AI algorithm. For reduced rank estimation, every tenth AI iterate is subsequently replaced by a PX-EM step. Two options are provided to modify this according to our perception of the quality of starting values for covariance components available, without the need to consider the 'advanced' options below.

--good

If we are confident, that we have good starting values, this might cause an unnecessary computational overhead. Specifying **--good** reduces the maximum number of (PX-)EM iterates to 1. Similarly, for potentially bad starting values, estimation might converge more reliably if a few more initial (PX-)EM iterates were carried out. Specifying **--bad** sets this number to 8. If the increase in log likelihood during the initial (PX-)EM iterates is less than 2.0, WOMBAT will switch to the AI algorithms immediately.

--bad

5.2.5 Numerical settings

--zero

WOMBAT uses a small positive value as operational zero to reduce numerical errors. The default value is 10^{-8}. This can be changed specifying the run option **--zero**n where n is a single-digit integer. This option redefines the operational zero to be 10^{-n}.

--pivot

For variance component estimation parameterising to the elements of the Cholesky factors of covariance matrices, estimates are constrained to the parameter space by restricting the pivots in the decomposition to a small value. The default is 0.001. This can be changed through the run option **--pivot**x where is the real value specifying the new limit.

5.2.6 Intermediate results

--best

WOMBAT writes out the currently best values of estimates of covariance components to the file **BestPoint** whenever the likelihood is increased. The option **--best** causes WOMBAT to read this file and write out the matrices of covariances and corresponding correlations

in more readily legible format to the file **BestSoFar.out**. WOMBAT with this option can be used in a directory in which an estimation run is currently active, without interfering with any of the files used.

5.2.7 Prediction only

WOMBAT can be used for a simple BLUP run, using the 'starting' values given in the parameter files as assumed values for the true covariances. In this mode, no pruning of pedigrees is carried out. If **-c** is specified in addition, WOMBAT will try to read the values from the file **BestPoint** instead – this is useful to obtain 'backsolutions' after convergence of an estimation run. Solutions can be obtained either directly or iteratively:

--blup

- Run option **--blup** selects the direct solution scheme. This involves sparse matrix inversion of the coefficient matrix, i.e. computational requirements are similar to those of a single REML iterate using the EM algorithm. While this can be computationally demanding for larger problems, it allows standard errors and expected accuracies (correlations between estimated and true effects) for random effects to be calculated from the diagonal elements of the direct inverse.

--solvit

- Run option **--solvit** specifies that the mixed model equations are to be solved iteratively. The default is a preconditioned conjugate gradient (PCG) algorithm, but Gauss-Seidel iterations can be invoked by specifying **--solvitgs** instead. Where applicable (multivariate or random regression models), either scheme utilises the inverse of the diagonal block comprising all equations pertaining to a particular random effects level. Up to $50\,000$ iterates are performed. The default convergence criterion requires the square root of the sum of squared deviations in solutions between iterates divided by the sum of squared solutions to be less than 10^{-8}. This can be overridden by specifying an integer number - to be the new exponent - immediately after the option; e.g. **--solvit6** sets the convergence criterion to a less stringent value of 10^{-6}.
  This option increases the limit on the maximum number of effects (equations) which can be fitted to a multiple of the maximum imposed for REML analyses. No attempt is made to approximate standard errors, and no checks for redundant animals in the pedigree are performed in this mode.

  Hint

  For large problems, **--solvit** is best combined with **--choozhz** (see Section 5.3.1).

--mmeout

- Run option **--mmeout** acts like **--solvit**, but writes out the complete mixed model equations in a form suitable for further analyses. This includes a file containing all non-zero elements in the lower triangle of the coefficient matrix (default name **MMECoeffMatrix.dat**), and the file **MMEEqNos+Solns.dat** which specifies the mapping of equation numbers to effects fitted, as well as giving the vector of right hand sides and solutions; see Section 7.2.9 for details.
- Run option **--s1step** provides an iterative solver as in **--solvit**, but targeted towards a so-called single-step analysis and providing a PCG algorithm only. The main difference is that equations for genotyped individuals are grouped together and are then processed as a dense block. This requires the inverse of the combined relationship matrix ($\mathbf{H}^{-1}$) to be supplied as a `*.gin` matrix, and a code for each individual – specifying whether is has been genotyped or not – to be given in the third, additional column of the pertaining `*.codes` file.

NEW Change: The default for the PCG algorithm has been changed to the so-called SSOR preconditioner. A simple diagonal preconditioner can be chosen by specifying **--s1stepA** and a block-diagonal scheme (the previous default) is selected via **--s1stepC**.

--snap

- Run option **--snap** selects a simple GWAS analysis – for fixed variance components – fitting a mixed model with a single SNP effect (fitted as a linear covariable) but processing multiple SNPs with reference allele counts read in from a separate file (default name **QTLAllels.dat** or **QTLAllels.bin**; see Section 6.5.3). This assumes no missing counts and employs an efficient computing strategy which exploits that only the equation for the SNP changes for different SNPs [32]. Estimated SNP effects together with their standard errors are written out to a file with the default name **QTLSolutions.dat** (see Section 7.2.10). This run option must be used in conjunction with a **SPECIAL** statement which identifies which of the covariables in the model of analysis is to be treated as SNP effects (see Section 4.10.6.6).

5.2.8 Simulation only

--simul

Specifying **--simul** causes WOMBAT to sample values for all random effects fitted for the data and pedigree structure given by the respective files, from a multi-variate normal distribution with a mean of zero and covariance matrix as specified in the parameter file. Again **-c** can be used to acquire population values from the file **BestPoint** instead. No fixed effects are simulated, but the overall (raw) mean

for each trait found in the data set is added to the respective records. Optionally, **--simul** can be followed directly (i.e. no spaces) by an integer number n in the range of 1 to 999. If given, WOMBAT will generate n simulated data sets (default $n = 1$).

Simulation uses two integer values as seeds to initialise the pseudo-random number generator. These can be specified explicitly in a file **RandomSeeds** (see Section 6.5.5.3). Output file(s) have the standard name(s) **SimData001.dat**, **SimData002.dat**, ..., **SimData**n**.dat**. These have the same layout as the original data file, with the trait values replaced by the simulated records; see Section 7.2.5.

This option is not available for models involving covariance option **GIN** (see Section 4.8.1.2).

5.2.9 Sampling to approximate standard errors

Large sample theory indicates that maximum likelihood estimates are normally distributed with covariance matrix equal to the inverse of the information matrix. Hence, sampling from this distribution has been advocated as a simple strategy to approximate standard errors of 'complicated' functions of variance components without the need for a linear approximation or to evaluate derivatives [27].

--sample

WOMBAT provides the run option **--sample**n to obtain such samples in a post-analysis step, with n denoting the number of samples to be drawn. If n is omitted, the number of samples is set to a value of 10000. It requires output files from an estimation run (at convergence), namely **BestPoint**, **AvInfoCovs** and **AvInfoParms** to specify the multivariate Normal distribution to sample from. By default, samples for all covariance components in the model are generated. In addition, the covariance matrix for a single random effect only can be selected by specifying its name in a **SPECIAL** statement (see Section 4.10.6.7). Samples generated are written to a file with the standard name **CovSamples_ALL.dat** or **CovSamples_**$name$.dat with $name$ the name of the random effect selected (see Section 7.2.12). These files are 'formatted' (text) files that are suitable for further analysis using standard statistical software packages. In addition, a file named **SumSampleAI.out** is written which summarizes details of the run together with means and variances across replicates.

This option is not implemented for random regression analyses or random effects with diagonal covariance matrices. It tends to work best for estimation runs with the PC option, even when matrices are estimated at full rank.

5.2.10 Matrix inversion only

WOMBAT can be used as a 'stand-alone' program to invert a real, symmetric matrix. Both a 'full' inverse and a 'sparse' inverse are accommodated. 'Full' inversion of a dense matrix is limited to relatively small matrices - use run time option **--limit** to find the maximum size allowed in WOMBAT.

The matrix is expected to be supplied in a file, with one record per non-zero element in the *u*pper triangle. Each record has to contain three space separated items: row number, column number and the matrix entry. There are several different 'modes':

--invert

1. Using **--invert** `filename` selects 'standard' generalised matrix inversion with pivoting on the largest diagonal; if the matrix if not of full rank, rows and columns corresponding to pivots with absolute value less than the operational zero are set to 0.

--invrev

2. Option **--invrev** `filename` is similar, but inversion is carried out processing rows and columns in ascending order of the number of off-diagonal elements in the original matrix. The rationale for this is that the computational effort required is proportional to the square of the number of off-diagonal elements. Hence, this option tends to provide faster inversion, but is more succeptible to numerical instabilities than **--invert**.

--inveig

3. Option **--inveig** `filename` first performs an eigen-value and -vector decomposition of the matrix. If eigenvalues less than a specified value (default: 0) are found, these are set to the minimum, and a modified matrix is written out to a file `filename.new` before obtaining the generalised inverse from the inverse of the diagonal matrix of eigenvalues (ignoring any 0 entries) and the matrix of eigenvectors. For a matrix not of full rank and the default minimum of 0, this yield a generalised inverse of the same rank as option **--invert**, but the generalised inverse is different. A value different from 0 can be selected by appending it to the option (no spaces).

Example

```
wombat -v inveig0.0001 matrix.dat
```

[1]

specifies inversion of the matrix stored in `matrix.dat`, obtaining a generalised inverse by setting any eigenvalues less than 0.0001 to this value, prior to inversion.

This option should only be used for relatively small matrices.

Hint

Use this feature (with a minimum eigenvalue > 0) to modify an 'invalid' matrix of starting values for multivariate analyses.

--invlap

4. Option **--invlap** `filename` inverts a symmetric, positive definite matrix (only!) by calling the appropriate LAPACK routines.

--invspa

5. Option **--invspa** `filename` chooses sparse matrix inversion. This is suitable for the inversion of large, sparse matrices where selected elements of the inverse are sufficient. Calculated and written out are all elements of the inverse corresponding to the non-zero elements in the Cholesky factor of the original matrix; any other elements of the inverse are ignored. The methodology is the same as that used in the Expectation-Maximisation steps in REML estimation. The relevant options to select the ordering strategy etc. (see Section 5.3.1) are recognised (except for **--metisall**), but must be given on the command line before **--invspa**.

The inverse is written out to `filename.inv` with one row per non-zero element, containing row number, column number and the element of the matrix (space separated).

Example

```
wombat -v ---amd ---invspa matrix.dat
```

specifies sparse matrix inversion of the matrix stored in `matrix.dat`, using approximate minimum degree ordering and requesting 'verbose' output. The output file containing the inverse is `matrix.dat.inv`.

5.2.11 Quadratic approximation

For some types of analyses a parameter (e.g. the dilution factor for "social" genetic effects) is fixed at a given value and variance components are estimated for this value. To estimate the parameter then requires multiple runs for different values, each generating a point of the profile likelihood for this parameter. Provided an initial triplet of points can be established, comprised of a 'middle' parameter value with the highest likelihood bracketed by points with lower likelihoods, a quadratic approximation of the curve can be used to locate its maximum. WOMBAT collects the pairs of parameter values and corresponding likelihoods in a file with the standard name `LogL4Quapprox.dat`. Specifying the

--quapp

run option **--quapp** invokes an attempt at a quadratic approximation, utilizing the information from this file. If successful (i.e. if a suitable

triplet of points can be found), the estimate of the parameter value at the maximum of the parabola is given, together with its approximate standard error.

5.2.12 Analyses of subsets of traits

For multivariate problems involving more than a few traits, it is often desirable to carry out analyses considering subsets of traits, for example, to obtain good starting values or to trace problems in higher-variate analyses back to particular constellations of traits.

--subset

WOMBAT provides the run time option **--subset**n to make such preliminary analyses less tedious. It will cause WOMBAT to read the parameter file for a multivariate analysis involving q traits and write out the parameter files for all q uni- ($n = 1$) or $q(q-1)/2$ bi-variate ($n = 2$) analyses possible. For $n > 2$, the program will prompt for the running numbers of the n traits to be considered and write out a single parameter file. These subset analyses are based on the data (and pedigree) file for the 'full' multivariate model, using the facility for automatic renumbering of traits and subset selection, i.e. no edits of these parameter files are necessary.

N.B.

This option does not carry through any information from a **SPECIAL** block, and is *not* implemented for random regression models or analyses involving correlated random effects or permanent environmental effects fitted as part of the residuals.

On analysis, encountering the syntax for trait renumbering (see Section 4.8.2) causes WOMBAT to write out an additional file with the estimates for the partial analysis (Standard name **EstimSubset**$n + \ldots + m$**.dat** with n to m the trait numbers in the partial analysis, e.g. **EstimSubset2+7.dat**; see Section 7.2.6). In addition, the name of this output file is added to a file called **SubSetsList**.

Hint

WOMBAT will add a line to **SubSetsList** on each run – this may cause redundant entries. Inspect & edit if necessary before proceeding to combining estimates !

5.2.13 Pooling estimates of covariance components from part analyses

Combining estimates of covariances from analyses involving different subsets of traits is a regular task. This may be a preliminary step to a higher-dimensional multivariate analysis to obtain 'good' starting

values. Alternatively, analyses for different subsets of traits may involve different data sets – selected to maximise the amount of information available to estimate specific covariances – and we may simply want to obtain pooled, possibly weighted estimates of the covariance matrices for all traits which utilise results from all partial analyses and are within the parameter space.

WOMBAT provides two procedures to combine estimates of covariance components from different analyses or modify existing matrices. These options are *not* available for analyses involving random regression models, correlated random effects or permanent environmental effects fitted as part of the residuals.

5.2.13.1 Iterative summing of expanded part matrices

--itsum

Option **--itsum** selects a run to combine estimates from partial analyses, using the 'iterative summing of expanded part matrices' approach of Mäntysaari [19] (see also Koivula et al. [17]), modified to allow for differential weighing of individual analyses. For this run, a file **SubSetsList** is assumed to exist and list the names of files containing results from analyses of subsets, and, optionally, the weightings to be applied (see Section 7.3.9). Pooled covariance matrices are written to a file name **PDMatrix.dat** as well as a file named **PDBestPoint** (see Section 7.2.7).

Hint

Use of **--itsum** is not limited to combining bi-variate analyses or the use of files with standard names (**EstimSubset**$n + \ldots + m$**.dat**), but all input files must have the form as generated by WOMBAT.
To use **--itsum** to combine estimates from analyses involving different data sets, be sure to a) number the traits in individual analyses appropriately (i.e. $1, \ldots, q$ with q the total number of traits, not the number of traits in a partial analysis), and b) to use the syntax described in Section 4.8.2 to renumber traits – this will 'switch on' the output of subset results files.
Copy **PDBestPoint** to **BestPoint** and run WOMBAT with option **--best** to obtain a listing with values of correlations and variance ratios for the pooled results.

5.2.13.2 Pooling using a (penalized) maximum likelihood approach

--pool
NEW

Option **--pool** is similar to **--itsum** but a) employs a maximum likelihood approach, as described by Meyer [26] , b) facilitates pooling of covariance matrices for all sources of variation simultaneously, and

c) allows for penalties to be imposed aimed at 'improving' estimates by reducing sampling variation.

Input is as for **--itsum** and pooled estimates are written to files named **PoolEstimates.out** (summary file) and **PoolBestPoint**.

5.2.14 Miscellaneous

--expiry Option **--expiry** will print the expiry date for your copy of WOMBAT to the screen.

--limits Option **--limits** can be used to find at the upper limits imposed on analyses feasible in WOMBAT, as 'hard-coded' in the program. *N.B.* Sometimes these are larger than your computing environment (memory available) allows.

--times Option **--times** causes WOMBAT to print out values for the CPU time used in intermediate steps.

--wide Option **--wide** will generate formatted output files which are wider than 80 columns.

--help Run option **--help** causes a list of available run options to be printed to screen.

5.3 Advanced run options

There are default values for most of these codes, which are adequate for most simpler analyses, and analyses of small to moderately sized data sets. Hence these options are predominantly of interest to users which want to fit complicated models or analyse large data sets, and thus need to tweak some of the options for ordering of equations, parameterisations, maximisation strategies and convergence criteria to get the best possible performance from WOMBAT.

In contrast to the basic options above, several of these options can be followed by one or more, optional numerical arguments. If such arguments are given, they must follow immediately (no spaces), and multiple options need to be separated by comma(s). Arguments must be given sequentially, i.e. if we want to set argument k, all preceding arguments $(1, \ldots, k-1)$ are required as well. If no arguments are given, the corresponding variables are set to their default values (see Table A.1).

5.3.1 Ordering strategies

The order in which the equations in the mixed model are processed can have a dramatic impact on the computational requirements of REML analyses. Hence, WOMBAT attempts to reorder the equations, selecting a default ordering strategy based on the number of equations; see Section A.1 for details. This can often be improved upon by selecting the strategy used explicitly.

--mmd Option **--mmd** specifies ordering using the multiple minimum degree procedure.

--amd Option **--amd** selects an approximate minimum degree ordering [1].

--metis Option **--metis** selects an ordering using a multilevel nested dissection procedure. Up to four optional arguments modifying the behaviour of this subroutine can be specified.

1. The number of graph separators to be considered, which can be between 0 and 20. As a rule, the higher this number, the better the quality of ordering tends to be. However, the time required for ordering increases substantially with this number, and in some cases intermediate values (between 3 and 9) have been found to work best. The manual for MeTiS recommends values up to 5. However, Meyer [23] found values as high as 12 or 14 to be advantageous for large analyses. By default, WOMBAT uses a value of 5 for analyses with more than 50 000 and up to 200 000 equations, and a value of 10 for larger models.
2. A factor which tells MeTiS which rows in the matrix are to be treated as dense. For instance, a value of 150 means all rows with more than 15% (factor divided by 10) elements than average. The default used by WOMBAT is 0.
3. An option to select the edge matching strategy employed by MeTiS. Valid values are 1 for random, 2 for heavy and 3 for standard edge matching. WOMBAT uses a default of 1.
4. An option to select whether MeTiS uses one- or two-sided node refinement. WOMBAT uses a default of 1 which, paradoxically (to agree with the MeTiS convention), invokes two-sided refinement. This is deemed to produce somewhat better orderings. An option of 2 here selects one-sided refinement, which can be faster.

In addition, the option **--metisall** is available. This causes WOMBAT to cycle through the number of graph separators from 5 to 16, considering density factors of 0 and 200, as well as random and standard edge matching (48 combinations). *Warning :* This can be quite time-consuming !

--chooz

To obtain the ordering, WOMBAT assigns a large matrix to store information on the non-zero elements in the mixed model matrix. The default size chosen is based on the number of equations and traits analysed, and is meant to be generous enough to be sufficient for a wide range of cases. In some instances, however, this can result in WOMBAT trying to allocate arrays which exceed the amount of RAM available and thus failing to run while, in reality, much less space is required for the analysis concerned. In other cases, the default guess is simply not sufficient. To solve these problems, the run time option **--choozhz** is supplied, which allows the user to set this number. If used as shown, it causes WOMBAT to pause, write out the default value, and read in (input from the terminal!) the new value. Such interactive input may be inconvenient in some scenarios.

NEW

Hence, alternatively, the new value may be specified as part of the run time option, immediately after (no spaces!) **--choozhz**. For example, **--choozhz11000000** sets the value no 11 million. If **--choozhz0** is given, WOMBAT will replace the 0 with the current, hard-coded maximum value for the array size (use run option **--limit** to find out what it is); this is convenient if something larger than the default is required and plenty of RAM is available.

5.3.2 REML algorithms

WOMBAT can be told to use a particular algorithm to locate the maximum of the likelihood function. In the following, let n (must be an Integer number) denote the maximum number of iterates to be carried out by an algorithm, and let C denote the convergence criterion to be used. Unless stated otherwise, C represents the threshold for changes in log likelihood between subsequent iterates, i.e. convergence is assumed to be reached if this is less than C.

--aireml

Option **--aireml** specifies a 'straight' AI algorithm. It can be followed by values for n and C. Valid forms are **--aireml**, **--aireml**n and **--aireml**n, C but not **--aireml**C.

Example

```
--aireml30,0.001
```

limits the number of iterates carried out to 30, and stops estimation when the change in log likelihood is less than 10^{-3}.

--nostrict

By default, the AI algorithms enforces an increase in log likelihood in each iterate. This can be switched off using the option **--nostrict**. This can improve convergence in some cases.

--modaim In this case, a modification of the AI matrix can result in better performance of the AI algorithm. Four procedures have been implemented in WOMBAT. These are selected by specifying **--modaim**n, with $n = 1, 2, 3, 4$ selecting modification by setting all eigenvalues to a minimum value ($n = 1$), by adding a diagonal matrix ($n = 2$, the default), or through a modified ($n = 3$) [37, 38] or partial ($n = 4$) [12] Cholesky decomposition of the AI matrix; see Section A.5 for details.

--emalg Option **--emalg** selects estimation using a standard EM algorithm. As for **--aireml**, this option can be followed by n or n, C to specify the maximum number of iterates allowed and the convergence criterion.

--pxem Option **--pxem** then selects estimation using the PX-EM algorithm. As above, this can be followed by n or n, C.

--pxai Option **--pxai** specifies a hybrid algorithm consisting of a few initial rounds of PX-EM, followed by AI REML. This is the default for full rank estimation (unless **-c** is specified without any explicit definition of the algorithm to be used). This option can have up to three sequential arguments, m denoting the number of PX-EM iterates, and n and C as above.

Example

```
---pxai6,50,0.001
```

[1]

defines 6 PX-EM iterates, followed by up to 50 AI iterates and a convergence criterion of 10^{-3} for the log likelihood.

--emai Option **--emai** is like **--pxai** except that the initial iterates are carried out using the standard EM algorithm. This is the default for reduced rank estimation. Optional arguments are as for **--pxai**.

--cycle Option **--cycle**, followed by an optional argument n, is used in conjunction with **--pxai** or **--emai**. If given, it will repeat the number of (PX-)EM and AI iterates specified by these options n times. If n is omitted, the number of cycles is set to 100. This option is useful for reduced rank analyses, where cycling algorithms appears to be beneficial.

--simplex Finally, WOMBAT incorporates two choices for a derivative-free algorithm. While little used, these can be usefully to check for convergence in 'tough' problems. Option **--simplex** selects the simplex or polytope algorithm due to Nelder and Mead [33], as previously implemented in DfReml. This option can have three arguments, n and C as above (but with the convergence criterion C describing the maximum variance among the log likelihood values in the polytope

allowed at convergence), and S the initial step size. Option **--powell** invokes maximisation using Powell [35]'s method of conjugate directions, again ported from DfReml and with optional parameters n, C (as for **--aireml**) and S.

--powell

Not really a REML algorithm: Option **--like1** selects a single likelihood evaluation for the current starting values, or, if combined with **-c**, the currently best point as contained in **BestPoint**. When combined with **-v**, the individual components of $\log \mathcal{L}$ are printed to the standard output (screen).

--like1

Similarly, option **--valid** does nothing more than calculate likelihood values. It is provided to aid in the use of cross-validation to estimate the tuning factor for penalised estimation. This involves obtaining estimates for a range of tuning factors for a 'training' data set. In doing so, WOMBAT creates a file **PenBestPoints.dat** (see Section 7.2.11). The corresponding likelihood values in the 'validation' data set are then easily obtained with the **--valid** run time option: for each set of estimates in **PenBestPoints.dat**, WOMBAT calculates the likelihood and writes it together with the tuning factor to a file **ValidateLogLike.dat**.

--valid

5.3.3 Parameterisation

By default, WOMBAT reparameterises to the elements of the Cholesky factor of the covariance matrices to be estimated (except for full rank analyses using the PX-EM or EM algorithm, which estimate the covariance components directly).

As a rule, the Cholesky factorisation is carried out pivoting on the largest diagonal element. This can be switched off with the **--noreord** option.

--noreord

Reparameterisation to the elements of the Cholesky factors removes constraints on the parameter space. Strictly speaking, however, it leaves the constraint that the diagonal elements should be non-negative. This can be removed by transforming the diagonal elements to logarithmic scale. This is selected using the option **--logdia**, which applies the transformation to all covariance matrices. Conversely, the option **--nologd** prevents WOMBAT from carrying out this transformation. If neither option is set (the default) and WOMBAT encounters small diagonal elements in any covariance matrices to be estimated during iterations, it will switch to taking logarithmic values of the diagonal elements for the respective matrices only.

--logdia

--nologd

5.3.4 Matrix storage mode

--dense

By default, WOMBAT assumes that the mixed model equations are sparse and carries out most computations using sparse matrix storage. In certain cases, this may be inappropriate and lead to substantial overheads – a typical example is the case where a random effect has a user-defined covariance matrix which is dense (e.g. a genomic relationship matrix). For this case, the option **--dense** is provided – specifying this option causes the one triangle of coefficient matrix to be stored assuming all elements are non-zero. Similarly, all calculations are carried out without checking for zero elements - this is done loading subroutines from the BLAS [7] and LAPACK [2] libraries, using packed storage of the coefficient matrix. For n equations, the latter requires $n(n+1)/2$ elements to be stored. With integer*4 addressing, this sets an upper limit of $n = 65,535$. For larger problems, the option **--dense2** is available which uses storage in two-dimensional arrays instead. Note that use of **--dense2** may require substantial amounts of RAM and that it is not yet implemented for all algorithms (e.g. not for PX(EM)).

NEW

--dense2

5.3.5 Sparse matrix factorisation, auto-differentiation and inversion

--old

By default, WOMBAT now employs a 'super-nodal' approach in the sparse matrix manipulations. This involves gathering dense sub-blocks and processing these using BLAS [7] and LAPACK [2] library routines. While this readily allows processing utilising multiple cores in these steps, the library routines require matrices to be strictly positive definite. Occasionally, they may fail where the previous procedure used may have circumvented the problem. The run time option **--old** is available to switch back to the latter.

5.3.6 Other

5.3.6.1 Pedigree pruning

--noprun

By default, WOMBAT 'prunes' the pedigree information supplied, i.e. eliminates any uninformative individuals (without records and links to only one other individual), before calculating the inverse of the numerator relationship matrix. Exceptions are prediction runs (option **--blup**) and models which fit correlated additive genetic effects. In some instances, however, it may be desirable not to 'prune' pedigrees. Pruning can be switched off through the run time option **--noprune**.

5.3.6.2 Overriding defaults for pedigree reduction

--redped

For sire model analyses, the default is not to carry out any pruning or pedigree reduction. The option **--redped** is provided to switch on pedigree reduction for sire model analyses, i.e. elimination of all individuals without a link to an individual in the data from the pedigree file.

--norped

Occasionally – for instance to compute breeding values for progeny of animals in the data – it is desirable not to reduce the pedigree prior to analysis. Providing the option **--norped** eliminates this step and switches off 'pruning' at the same time. Option **--redped** is invoked automatically for runs with options **--solvit** or **--s1step**.

5.3.6.3 No programmed pause please

--batch

Generally, WOMBAT does not require any interactive input. An exception is a 'pause' after a warning message on a constellation of options which is new or can be problematic. These programmed pauses can be switched off using the run time option **--batch**.

5.3.6.4 Centering data

--nocent

By default, WOMBAT 'centers' data, i.e. subtracts the trait-specific mean from each observation. For some (rare) tasks this is undesirable. Hence, the option **--nocenter** is provided which eliminates this step.

5.3.6.5 Choosing the algorithm to calculate inbreeding

--maxgen

By default, WOMBAT uses the algorithm of Tier [39] to compute inbreeding coefficients before setting up the inverse of the numerator relationship matrix between individuals. This algorithm is implemented with a limit of **23** generations. Occasionally, this results in insufficient space, both for pedigrees with mode generations and very large pedigrees. The option **--maxgen**, followed immediately by two digits giving the new value (e.g. maxgen25) allows this limit to be increased, up to a maximum value of 30. Note that this may result in WOMBAT trying to assign some very large arrays, and may slow calculations.

--quaas
--meuw

Alternative algorithms, due to Quaas [36] and Meuwissen and Luo [20], which do not have this limitation but are slower and use very little memory, can be selected using the run time option **--quaas** and **--meuw**, respectively.

5.3.6.6 Using only the first n observations

--nonly

Sometimes, it is useful to carry out a preliminary analysis considering a small(ish) subset of the data only. For instance, we may want to test that we have fitted the model correctly or we may want to estimate suitable starting values. A quick way to do this is available through the option **--nonly**N, where N is an integer number specifying the number of records to be used (no space between nonly and N). For example, `--nonly1000` selects an analysis utilising the first 1000 records only.

5.3.6.7 Omitting the preliminary LSQ analysis

--nolsq

One of the preliminary steps carried out by WOMBAT is a simple least-squares analysis of the fixed part of the model – this aims at identifying any dependencies amongst fixed effects levels which might not have been correctly specified. However, if there are many cross-classified effects or covariables, this can be very slow, especially for large data sets (code for this step is very old and probably the weakest part of WOMBAT and not as highly optimised as the main part of the calculations). Option **--nolsq** is available to skip this step and is recommended for large analyses with many cross-classified effects – but it does rely on all rank deficiencies in the part of the coefficient matrix for fixed effects being identified: the run is likely to fail with an appropriate error message otherwise.

A scenario where this option is useful is when we fit different random effects for a given data set where the fixed part of the model remains constant – typically, it is then sufficient to carry out the least-squares step only for the first run. Alternatively, it is beneficial for simulations studies where the data is re-sampled, so that data and pedigree structure are the same for all replicates.

5.3.6.8 Omit writing out solutions

--nosolut

Run option **--nosolut** will skip writing out files with fixed and random effects solutions at the end of an estimation run. This is useful for simulation studies where these are not of interest. -

5.4 Parameter file name

If a parameter file other than **wombat.par** is to be used, this can be given as the *last* entry in the command line. The extension **.par** can be omitted.

Example

```
wombat  weights
```

specifies a run where WOMBAT expects the parameter file **weights.par**.

Table 5.2: 'Advanced' run options for 𝕎𝕆𝕄𝔹𝔸𝕋- part I

Option	Purpose
Numerical settings	
--zero	Change value of operational **zero**
--pivot	Change value for smallest **pivot** in Cholesky factor (covariance matrix)
Ordering strategies for mixed model matrix (MMM)	
--metis	Use multi-level nested dissection procedure (**MeTiS**) to re-order MMM
--mmd	Use **m**ultiple **m**inimum **d**egree method to re-order MMM
--amd	Use **a**pproximate **m**inimum **d**egree algorithms for re-ordering
--choozhz	Assign initial size of mixed model matrix interactively
Specific REML algorithms	
--aireml	Use the **AI** algorithm
--nostrict	Allow the **AI** algorithm to take steps decreasing the log likelihood
--modaim	Choose method to **mod**ify the **AI** **m**atrix to ensure it is positive definite
--emalg	Use the standard **EM**-**alg**orithm
--pxem	Use the **PX**-**EM** algorithm
--emai	Use a few rounds of **EM** followed by **AI** REML
--pxai	Use a few rounds of **PX** followed by **AI** REML *(default)*
--cycle	Repeat **cycle**s of (PX-)EM and AI iterates
--simplex	Use the **Simplex** procedure for derivative-free (DF) maximisation
--powell	Use **Powell**'s method of conjugate directions (DF)
--force	**Force** derivative-free search after AI-REML steps
--like1	Carry out a single **like**lihood evaluation only
--valid	Carry out likelihood evaluations for multiple estimates
Parameterisations	
--logdia	Select **log** transformation of **dia**gonal elements of Cholesky factor
--nologd	**No log** transformation of **d**iagonal elements of Cholesky factor
--reorder	Pivot on largest diagonal elements of covariance matrices during Cholesky factorisation *(default)*
--noreord	Carry out Cholesky decomposition of covariance matrices sequentially
--reconstr	Allow change in order of pivots of Cholesky factors of covariance matrices and **reconstr**uction of mixed model matrix
--noreconst	Do not allow changes in order of pivots and reconstruction of mixed model matrix

Table 5.3: 'Advanced' run options for WOMBAT- part II

Option	Purpose
Miscellaneous	
--nonly$N$	Consider only the first N observations
--noprune	Do not prune pedigrees
--nocenter	Do not subtract mean from observations
--nolsq	Skip least-squares analysis for fixed effects
--batch	Do not pause after warning messages
--maxgen	Increase the space available in Tier"s algorithm to set up NRM inverse
--quaas	Use algorithm of Quaas [36] to set up NRM inverse
--meuw	Use algorithm of Meuwissen and Luo [20] to set up NRM inverse
--redped	Switch on reduction of pedigree file for a sire model.
--norped	Do not carry out pedigree reduction.

6 Input files for WOMBAT

6.1 Format

All input files supplied by the user are expected to be 'formatted' (in a Fortran sense), i.e. should be plain text of ASCII file.

Non-standard characters may cause problems !

Hint

Take great care when switching from DOS/Windows to Linux or *vice versa*: Remember that these operating systems use different end-of-line coding - this means that you may have to 'translate' files using a utility like `dos2unix` (`unix2dos`) or `fromdos` (`todos`).

6.2 Data File

The data file is mandatory. It gives the traits to be analysed, and all information on effects in the model of analysis. It is expected to have the following features:

File name

1. There is no 'default' name for the data file. File names up to 30 characters long are accommodated.

Format

2. Variables in the data file should be in fixed width columns, separated by spaces.
3. Each column, up to the maximum number of columns to be considered (= number of variables specified in the parameter file),

must have a numerical value – even if this column is not used in the analysis, i.e. no 'blank' values !

4. All codes of effects to be considered (fixed, random or 'extra' effects) must be positive integer variables, i.e. consist of a string of digits only.
   The maximum value allowed for a code is $2\,147\,483\,647$, i.e. just over 2 billion.

integer codes

5. All traits and covariables (including control variables) are read as real values, i.e. may contain digits, plus or minus signs, and Fortran type formatting directives only.

real variables

> *N.B.*
> Calculations in WOMBAT use an operational zero (default value: 10^{-8}), treating all smaller values as zero. To avoid numerical problems, please ensure your traits are **scaled** so that their variances are in a moderate range (something like 10^{-5} to 10^{5}).

6. Any alphanumeric strings in the part of the data file to be read by WOMBAT are likely to produce errors !

7. For multi-trait analyses, there should be one record for each trait recorded for an individual[1]. The trait number for the record should be given in the first column.

Missing values

No special codes for 'missing values' are available – missing traits are simply absent records in the data file.

8. The data file must be *sorted* in ascending order, according to :

Order of records

   i) the individual (or 'subject') for which traits are recorded, and
   ii) according to the trait number within individual.
   iii) For RR analyses, records are expected to be sorted according to the value of the control variable (within individual and trait number) in addition.

   > *N.B.*
   > WOMBAT does not allow 'repeated' records for individual points on the trajectory in RR analyses, i.e. you can not have multiple observations for an individual with the same value of the control variable.

9. For multivariate analyses combining traits with repeated and single records, the traits with repeated records need to have a

[1] Yes, this may result in some duplication of codes, if the model is the same for all traits !

lower trait number than those with single records only.

Annotation

To facilitate annotation of the data file (e.g. column headers, date of creation, source), WOMBAT will skip lines with a '#' (hash sign) in column 1 at the beginning of the file - there is no limit on the number, n, of such lines, but they must represent the first n lines (any '#' elsewhere will cause an error).

6.3 Pedigree File

If the model of analysis contains random effect(s) which are assumed to be distributed proportional to the numerator relationship matrix, a pedigree file is required. It is expected to have the following features:

File name

1. There is no 'default' name for the pedigree file. File names up to 30 characters long are accommodated.

Parents

2. The pedigree file must contain one line for each animal in the data. Additional lines with pedigree information for parents without records themselves can be included.

Layout

3. Each line is expected to contain three integer variables :

   (a) the animal code,
   (b) the code for the animal's sire,
   (c) and the code for the animal's dam.

   All codes must be valid integer in the range of 0 to $2\,147\,483\,647$. Additional, optional variables in the fourth or fifth column can be:

   (d) the animal's inbreeding coefficient (real variable between 0 and 1),
   (e) a code of 1 (males) or 2 (females) (integer), defining the number of X chromosomes, if a corresponding relationship matrix is to be set up.

Coding

4. All animals must have a numerically higher code than either of their parents.
   Unknown parents are to be coded as "0".

5. If maternal genetic effects are to be fitted in the model of analysis, all dams of animals in the data must be 'known', i.e. have codes > 0.

Order

6. The pedigree file does not need to be sorted. However, sorting according to animal code (in ascending order) is desirable, since it will yield slightly reduced processing time.

As for the data file, any lines at the beginning of the pedigree file with a '#' (hash sign) in column 1 are ignored.

6.4 Parameter File

WOMBAT acquires all information on the model of analysis from a parameter file.

Rules to set up the parameter file are complex, and are described in detail in a separate chapter (Chapter 4).

6.5 Other Files

Depending on the model of analysis chosen, additional input files may be required.

6.5.1 General inverse file

For each random effect fitted for which the covariance option **GIN** (see Section 4.8.1.2) has been specified, WOMBAT expects a file set up by the user which contains the *inverse* of the matrix (such as relationship or correlation matrix) which determines the 'structure' of the covariance matrix for the random effect. The following rules apply :

File name

1. The file name should be equal to the name of the random effect, with the extension **.gin**. For example, **mother.gin** for a random effect called **mother**.
   For random effect names containing additional information in round brackets, for instance in RR analysis, only the part preceding the '(' should be used. In this case, be careful to name the effects in the model so that no ambiguities arise!
2. The first line of the file should contain a real variable with value equal to the log determinant of the covariance/general relationship matrix (NB: This is the log determinant of the matrix, not of the inverse; this can generally be calculated as a 'by-product' during inversion).
   This comprises a constant term in the (log) likelihood, i.e. any value can be given (e.g. zero) if no comparisons between models are required.

NEW

Optionally, this can be followed (separated by space(s)) by the keyword "DENSE". If given, WOMBAT will store the elements of the general relationship matrix in core, assuming it is dense, i.e. for n levels, an array of size $n(n+1)/2$ is used. This can require substantial additional memory, but reduces the overhead

incurred by re-reading this matrix from disk for every iteration, and may be advantageous if the matrix is (almost) dense, such as the inverse of a genomic relationship matrix.

3. The file should then contain one line for each non-zero element in the inverse. Each line is expected to contain three space-separated variables :
   (a) An integer code for the 'column' number
   (b) An integer code for the 'row' number
   (c) A real variable specifying the element of the inverse

   Here 'row' and 'column' numbers should range from 1 to N, where N is the number of levels for the random effect.
   Only the elements of the *lower* triangle of the inverse should be given and given 'row-wise', i.e. WOMBAT expects a 'column' number which is less than or equal to the 'row' number.

Layout

6.5.1.1 Codes for GIN levels

By default, WOMBAT determines the number of levels for a random effect with covariance option **GIN** from the data, renumbering them in ascending numerical order. In some cases, however, we might want to fit additional levels, not represented in the data. A typical example is am additional genetic effect, which can have levels not in the data linked to those in the data through covariances arising from co-ancestry.

If WOMBAT encounters row or column numbers greater than the number of random effect levels found in the data, it will take the following action:

1. It is checked that this number does not exceed the maximum number of random effects levels as specified in the parameter file. If it does, WOMBAT stops (change parameter file if necessary).
2. WOMBAT looks for a file with the same name as the **.gin** file but extension .codes; e.g. **mother.codes** for the random effect **mother**. This file is expected to supply the codes for all levels of the random effect: There has to be one line for each level with two space separated integer variables, the running number (1st) and the code for the level (2nd).
3. If such file is not found, WOMBAT will look for a genetic effect (i.e. a random effect with covariance option **NRM**) which has the same number of levels as the current random effect. If found, it will simply copy the vector of identities for that effect and proceed. (Hint: you may have to use run time **--noprune** to utilise this feature).
4. Finally, if neither of these scenarios apply, WOMBAT will assume

File name

the random levels are coded from 1 to N and try to proceed without any further checking – this may cause problems!

6.5.2 Basis function file

If a regression on a user- defined set of basis functions has been chosen in the model of analysis by specifying the code **USR** for a covariable (or 'control' variable in a RR analysis), file(s) specifying the functions need to be supplied.

The form required for these files is:

1. The name of the file should be the name of the covariable (or 'control' variable), as given in the parameter file (model of analysis part), followed by **_USR**, the number of coefficients, and the extension **.baf**.

File name

Example

If the model of analysis includes the effect **age** and the maximum number of regression coefficients for age is 7, the corresponding input file expected is **age_USR7.baf**

N.B.

The file name does not include a trait number.
This implies, that for multivariate analyses the same basis function is assumed to be used for a particular covariable across all traits. The only differentiation allowed is that the number of regression coefficients may be different (i.e. that a subset of coefficients may be fitted for some traits); in this case, the file supplied must correspond to the largest number of coefficients specified.

2. There should be one row for each value of the covariable.

3. Rows should correspond to values of the covariable in ascending order.

4. The number of columns in the file must be equal to (or larger than) the number of regression coefficients to be fitted (i.e. the order of fit) for the covariable.

5. The elements of the $i-$th row should be the user-defined functions evaluated for the $i-$th value of the covariable.

Example

Assume the covariable has possible values of 1, 3, 5, 7 and 9, and that we want to fit a cubic regression on 'ordinary' polynomials, including the intercept. In this case, WOMBAT would expect to find a file with 5 rows (corresponding to the 5 values of the covariable) and 4 columns (corresponding to the 4 regression coefficients, i.e. intercept, linear, quadratic and cubic):

```
1   1   1    1
1   3   9   27
1   5  25  125
1   7  49  343
1   9  81  729
```

Note that there is no leading column with the value of the covariable (you can add it as the last column which is ignored by WOMBAT, if you wish) – the association between value of covariable and user defined function is made through the order of records.

6.5.3 File with allele counts

For an analysis using the run option **--snap**, an additional input file is required which supplies the counts for the reference allele for each QTL or SNP to be considered. This has the default name **QTLAllels.dat** or **QTLAllelsR.dat**, depending whether integer or or real input is chosen. If both exist in the working directory, WOMBAT will utilize the former and ignore the latter.

- **QTLAllels.dat** must be a formatted file with one row per QTL. Each row should contain a single digit (usually 0, 1, 2) for all individuals in the data, without any spaces between them! At present, there is no provision for missing genotypes (these are readily imputed). In contrast to most other input files used by WOMBAT, information is obtained in a Fortran formatted read and a blank line is treated as a line of zeros. For example, if there are 1000 individuals, each line should be 1000 characters long. The number of SNPs processed is given by the number of records (rows) in the file.
- **QTLAllesR.dat** accommodates the situation where – for some reason or other – a format with space separated values is preferred. This removes the restriction of a single digit. 'Counts' are read as real values, i.e. can contain decimals. Values for a SNP can be spread over any number of rows, but counts for each new SNP must begin on a new row.

6.5.4 Files with results from part analyses

6.5.4.1 List of partial results

For a run with option **--itsum** or **--pool**, WOMBAT expects a number of files with results from part analyses as input. Typically, these have been generated by WOMBAT when carrying out these analyses; see Section 7.2.6 for further details.

6.5.4.2 Single, user generated input file

For run option **--pool**, results can be given in a single file instead. For each part analysis, this should contain the following information:

1. A line giving (space separated):
   a) The number of traits in the part analysis
   b) The (running) numbers of these traits in the full covariance matrix.
   c) The relative weight to be given to this part; this can be omitted and, if not given, is set to 1.
2. The elements of the upper triangle of the *residual* covariance matrix, given row-wise.
3. For each random effect fitted, the elements of the upper triangle, given row-wise. Each matrix must begin on a new line and the matrices must given in the same order as the corresponding **VAR** statements in the parameter file.

6.5.5 'Utility' files

WOMBAT will check for existence of other files with default names in the working directory and, if they exist, acquire information from them.

6.5.5.1 File **RunOptions**

This file can be used as an alternative to the command line to specify run options (see Chapter 5).
It must have one line for each run option specified, e.g.

```
-v
--emalg
```

to specify a run with verbose output using the EM-algorithm.

6.5.5.2 File **FileSynonyms**

In some cases, WOMBAT expects input files with specific names. If files with different default names have the same content, duplication

can be avoided by setting up a file **FileSynonyms** to 'map' specific files to a single input file. This file should contain one line for each input file to be 'mapped' to another file. Each line should give two file names (space separated) :

(a) The default name expected by WOMBAT.
(b) The name of the replacement file

Example

```
age.baf      mybasefn.dat
damage.baf   mybasefn.dat
```

[Not yet implemented !]

6.5.5.3 File RandomSeeds

To simulate data, WOMBAT requires two integer values to initialise the random number generator. If the file **RandomSeeds** exists, it will attempt to read these values from it. Both numbers can be specified on the same or different lines. If the file does not exist in the working directory, or if an error reading is encountered, initial numbers are instead derived from the date and time of day.

WOMBAT writes out such file in each simulation run, i.e. if **RandomSeeds** exists, it is overwritten with a new pair of numbers !

6.5.6 File SubSetsList

For a run with option **--itsum**, WOMBAT expects to read a list of names of files with results from subset analyses in a file with the standard name **SubSetsList**. This has generated by WOMBAT (see Section 7.3.9) if the part analyses have been carried out using WOMBAT, but may need editing. In particular, if a weighted summation is required, the default weights of '1.000', need to be replaced 'manually' by appropriate values, selected by the user !

6.5.7 File(s) Pen*(.dat)

6.5.7.1 File PenTargetMatrix

For penalty options **COVARM** and **CORREL** a file with this name must be supplied which gives the shrinkage target. This must be a positive definite matrix. The file should be a plain text file and contain the elements of the *upper* triangle of the matrix. It is read in 'free' format, i.e. variable numbers of elements per line are allowed.

6.5.7.2 File **PenBestPoints.dat**

A run with the option **--valid** expects to read sets of estimates from a file with this name. This is generated by WOMBAT when penalized estimation is specified, but can be edited to suit or generated by other means. For each tuning factor, it should contain:

(a) A line with the tuning factor (realvariable) at the beginning
(b) The elements of the *upper* triangle of estimate the residual covariance matrix (or equivalent) for this tuning factor. This is read in 'free' format, i.e. can be given over as many lines suitable.
(c) Starting on a new line: The elements of the *upper* triangle of estimate the genetic covariance matrix (or equivalent) for this tuning factor. Again, this is read in 'free' format.

7 Output generated by WOMBAT

This chapter describes the files written out by WOMBAT. These comprise 'internal' files generated by WOMBAT for use within a run (and thus of limited interest to the user), and various output files. Most files have default names, independent of the analysis.

7.1 Main results files

These are formatted summary files to be read (rather than processed by other programs). They have the extension **.out**.

7.1.1 File SumPedigrees.out

If a pedigree file is specified, this file gives some summary statistics on the pedigree information found.

7.1.2 File SumModel.out

This file gives a summary about the model of analysis specified and the corresponding features of the data found. Statistics given include means, standard deviations and ranges for traits and covariables, and numbers of levels found for the effects in the model of analysis.

It is written after the first processing of the input files, i.e. during the set-up steps.

7.1.3 File SumEstimates.out

This files provides estimates of the parameters as estimated, and the resulting covariance matrices and their eigenvalues and, for reduced rank analyses, the corresponding matrices of eigenvectors. WOMBAT also writes out the corresponding matrices of correlations and variance ratios. In addition, values for any user-defined functions of covariances (see Section 4.10.4) are written out.

Standard errors

If the final estimates were obtained using the AI algorithm, WOMBAT provides approximate sampling errors for the parameters and covariance components estimated, as given by the inverse of the respective average information matrices. In addition, sampling errors of variance ratios and correlations are derived, as described in Section A.4.2.

7.1.4 File BestSoFar.out

This is an abbreviated version of **SumEstimates.out**, written out when the command line option **--best** is specified. It gives matrices of estimates pertaining to the set of parameters with the highest likelihood found so far.

7.1.5 File FixSolutions.out

This file lists the generalised least-squares solutions for all fixed effects fitted, together with 'raw' means and numbers of observations for individual subclasses.

Hint

If this file is the 'by-product' of an estimation run using the AI-REML algorithm (default), no standard errors for fixed effects are given. The reason is that the AI algorithm does not involve calculation of the inverse of the coefficient matrix of the mixed model equations. Asymptotic lower bound standard errors are written out, however, if the (PX-)EM algorithm is used in the last iterate performed, or if a BLUP run is specified, as both evaluate this inverse. Hence, to obtain standard errors, carry out one iterate using the EM algorithm and specifying a continuation run:

wombat -c -emalg1 parfile.par

(replacing parfile.par by the name of your parameter file). This also provides a check on convergence - the increase in $\log \mathcal{L}$ from the EM step should be very small. Note though that for large and very large problems, the EM iterate can require substantially longer (CPU time) than the AI algorithm. Alternatively, specify a BLUP run:

wombat -c -blup parfile.par

Again, note that the latter does not involve pruning of the pedigree, i.e. WOMBAT will recalculate the inverse of the numerator relationship matrix and owerwrite the existing file(s) `nrminv`n`bin`.

7.1.6 File SumSampleAI.out

This file is only written out when specifying run option **--sample** (see Section 5.2.9. It gives a brief summary of the characteristics of the analysis and average information matrix for which samples were obtained, together with means and variances across replicates. In addition, large deviations between theoretical results from the information matrix and samples obtained are flagged.

7.2 Additional results

These are large files, most likely subject to further processing by other programs. Thus they contain minimum or no text. They have extension **.dat**.

7.2.1 File Residuals.dat

This files gives the residuals for all observations, for the model fitted and current estimates of covariance components. It has the same order as the data file, and contains 2 space separated columns :

(a) Column 1 contains the estimated residual.
(b) Column 2 gives the corresponding observation, as deviation from the trait mean.

Summary statistics about the distribution of residuals can be readily obtained using standard statistical packages. For example, the following R commands compute means, standard deviations and quartiles, and plot the two columns against each other as well as a distribution histogram for the residuals :

Example

```
res<-read.table(‘‘Residuals.dat’’)
summary(res); sd(res)
par(mfrow=c(1,2))
plot(res); hist(res[,1])
```

7.2.2 File(s) RnSoln_rname.dat

Solutions for each random effect are written to a separate file. These files have names **RnSoln_rname.dat**, with **rname** representing the name of the random effect. Columns in these files are :

(a) Column 1 gives the running number for the level considered
(b) Column 2 gives the corresponding original code; this is only printed for the first ’effect’ fitted for the level.
(c) Column 3 gives the ’effect’ number, where, depending on the analysis, ’effect’ is a trait, principal component or random regression coefficient.
(d) Column 4 gives the solution.
(e) Column 5 gives the sampling error of the solution, calculated as the square root value of the corresponding diagonal element of the coefficient matrix in the mixed model equations. This is only available, if the last iterate has been carried out using an EM or PX-EM algorithm.

For genetic random effects with covariance option **NRM**, WOMBAT calculates inbreeding coefficients from the list of pedigrees specified. For such effects, there may be an additional column containing these coefficients (in %). This should be the *l*ast column in the **RnSoln_rname.dat** file (NB: For multivariate analyses, this is only given once per individual, in the line corresponding to the first trait).

There may be up to 7 columns – ignore column 6 unless you recognize the numbers (calculations for column 6 are not fully debugged, but may be o.k. for simple models).

If you have carried out a reduced rank analysis, i.e. give the PC option for the analysis type, the solutions in **RnSoln_rname.dat** pertain to the principal components! You might then also be inter-

ested in the corresponding solutions on the original scale – WOMBAT endeavours to calculate these for you and writes them to the file **RnSoln_rname-tr.dat**. However, if you have carried out a run in which you have calculated standard errors for the effects fitted, these are ignored in the back-transformation and you will find that column 5 (in **RnSoln_rname-tr.dat**) consists entirely of zeros – this does not mean that these s.e. are zero, only that they have not been determined.

7.2.3 File(s) Curve_cvname(_trname).dat

At convergence, curves for fixed covariables fitted are evaluated and written to separate files, one per covariable and trait. These have names **Curve_cvname.dat** for univariate analyses and **Curve_cvname_ trname.dat** for multivariate analyses, with **cvname** the name of the covariable as specified in the parameter file and, correspondingly, **trname** the name of the trait. Curves are only evaluated at points corresponding to nearest integer values of values found in the data. Each file has four columns :

(a) Column 1 gives the value of the covariable.
(b) Column 2 gives the point on the fitted curve.
(c) Column 3 contains the number of observations with this value of the covariable.
(d) Column 4 gives the corresponding raw mean.

Hint

To get most information from these files, it might be worth your while scaling your covariables prior to analysis!

7.2.4 File(s) RanRegname.dat

For random regression analyses, WOMBAT evaluates variance components (file **RanRegVariances.dat**), variance ratios (file **RanRegVarRatios.dat**, not written if more than one control variable is used) and selected correlations (**RanRegCorrels.dat**) for values of the control variable(s) occurring in the data. If approximate sampling variances of parameters are available, it is attempted to approximate the corresponding sampling errors. The general layout of the files is as follows :

(a) Column 1 gives the running number of the value of the control variable.
(b) Column 2 gives the corresponding actual value (omitted if more than one control variable).

(c) The following columns give the variance components, ratios or correlations.
   (i) If sampling errors are available, each source of variation is represented by two columns, i.e. value followed by the approximate lower bound sampling error, with additional spaces between 'pairs' of numbers.
   (ii) Random effects are listed in same order as the starting values for random effects covariances are given in the parameter file.
   (iii) If the same control variable is used for all random effects, it is attempted to calculate a total, 'phenotypic' variance and corresponding variance ratios and correlations.
   (iv) Correlations are calculated for 5 values of the control variable, corresponding to lowest and highest value, and 3 approximately equidistant intermediate values.

In addition, the files contain some rudimentary headings.

IF the special option `RRCORR-ALL` has been specified (see Section 4.10.6.13), a file **`RanRegCorrAll.dat`** is written out in addition. This contains the following columns:

(a) The name of the random effect
(b) The running number for trait one
(c) The running number for trait two
(d) The running number for the pair of traits
(e) The value of the control variable ("age") for trait one
(f) The value of the control variable for trait two
(g) The estimated covariance between traits one and two for the specified ages
(h) The corresponding correlation

7.2.5 Files `SimDatan.dat`

Simulated records are written to files with the standard name **`SimDatan.dat`**, where n is a three-digit integer value (i.e. 001, 002, ...). These files have the same number of columns as specified for the data file in the parameter file (i.e. any trailing variables in the original data file not listed are ignored), with the trait values replaced by simulated records. These variables are followed by the simulated values for individual random effects : The first of these values is the residual error term, the other values are the random effects as sampled (standard uni-/multivariate analyses) or as evaluated using the random regression coefficients sampled - in the same order as the corresponding covariance matrices are specified in the parameter file.

Except for the trait number in multivariate analyses (first variable), all variables are written out as real values.

7.2.6 Files EstimSubSet*n*+...+*m*.dat

If an analysis considering a subset of traits is carried out, WOMBAT writes out a file **EstimSubset***n*+...+*m***.dat** with the estimates of covariance matrices for this analysis. Writing of this file is 'switched on' when encountering the syntax **"m"->n**, specifying the trait number in the parameter file (see Section 4.8.2). The first two lines of **EstimSubset***n*+...+*m***.dat** gives the following information :

(a) The number of traits in the subset and their names, as given in the parameter file.
(b) The corresponding trait numbers in the 'full' analysis.

This is followed by the covariance matrices estimated. The first matrix given is the matrix of residual covariances, the other covariance matrices are given in the same order as specified in the parameter file.

(c) The first line for each covariance matrix gives the running number of the random effect, the order of fit and the name of the effect
(d) The following lines give the elements of covariance matrix, with one line per row.
    NB The number of rows written is equal to the number of traits in the subset; for random effects not fitted for all traits, corresponding rows and columns of zeros are written out.

Finally, **EstimSubset***n*+...+*m***.dat** gives some information on the data structure (not used subsequently) :

(e) The number of records for each trait
(f) The number of individuals with pairs of records
(g) The number of levels for the random effects fitted

7.2.7 Files PDMatrix.dat and PDBestPoint

These files give the pooled covariance matrices, obtained running WOMBAT with option **--itsum**.

PDMatrix.dat is meant to be readily pasted (as starting values) into the parameter file for an analysis considering all traits simultaneously. It contains the following information for each covariance matrix :

(a) A line with the qualifier **VAR**, followed by the name of the random effect and the order and rank of the covariance matrix.

(b) The elements of the upper triangle of the covariance matrix; these are written out as one element per line.

PDBestPoint has the same form as **BestPoint** (see Section 7.3.3). It is meant to be copied (or renamed) to **BestPoint**, so that WOMBAT can be run with option **--best** to generate a 'results' file (**BestSoFar**) with correlations, variance ratios and eigenvalues of the pooled covariance matrices.

7.2.8 Files PoolEstimates.out and PoolBestPoint

These files provided the results from a run with the option **--pool**:

1. **PoolEstimates.out** summarizes characteristics of the part estimates provided (input), options chosen, and results for all analyses carried out.
2. **PoolBestPoint** is the equivalent to **BestPoint**. If penalized analyses are carried out, copies labelled **PoolBestPoint_unpen** and **PoolBestPoint_t**xx, with xx equal to the tuning factor, are generated so that files for all sub-analyses are available at the end of the run.

7.2.9 Files MME*.dat

The two files **MMECoeffMatrix.dat** and **MMEEqNos+Solns.dat** are written out when the run option **--mmeout** is specified.

MMECoeffMatrix.dat : contains the non-zero elements in the lower triangle of the coefficient matrix in the MME. There is one line per element, containing 3 space-separated variables:

(a) The row number (integer); in running order from 1 to N, with N the total number of equations.
(b) The column number (integer); in running order from 1 to N.
(c) The element (real).

Hint

This file is in the correct format to be inverted using run option **--invert** or **--invrev**.

MMEEqNos+Solns.dat provides the mapping of equation numbers (1 to N) to effects in the model, as well as the right hand sides and solutions. This file has one line per equation, with the following, space separated variables:

(a) The equation number (integer).
(b) The name of the trait, truncated to 12 letters for long names.
(c) The name of the effect, truncated to 12 letters (For random effects and analyses using the PC option, this is replaced by

PCn).

(d) The original code for this level and effect (integer); for covariables this is replaced by the 'set number' ($= 1$ for non-nested covariables).
(e) The running number for this level (within effect).
(f) The right hand side in the MME (real).
(g) The solution for this effect (real).

7.2.10 File QTLSolutions.dat

This is the output file for a run using option **--snap**. It contains one line for each line found in the input file **QTLAllels.dat** containing

(a) The estimated SNP effect (regression coefficient).
(b) Its standard error (from the inverse of the coefficient matrix).
(c) The $t-$value, i.e. the ratio of the two variables.
(d) A character variable with a name for the line.

7.2.11 Files Pen*(.dat) and ValidateLogLike.dat

The following files are created in conjunction with penalized estimation. Some can be used by WOMBAT in additional steps.

7.2.11.1 File PenEstimates.dat

This file gives a brief summary of estimates together with log likelihood values for all values of the tuning parameter given.

7.2.11.2 File PenBestPoints.dat

This file collects the **BestPoint**'s for all tuning parameters. The format for each is similar to that for **BestPoint** (see Section 7.3.3), except that the first line has 3 entries comprising the tuning factor, the maximum penalized likelihood and the corresponding unpenalized value. It is suitable as input for additional calculations aimed at comparing estimates, and used as input file for 'validation' runs (see run option **--valid**, Section 5.3.2).

Output to this file is cumulative, i.e. if it exists in the working directory it is not over-written but appended to.

7.2.11.3 File PenCanEigvalues.dat

Similarly, this file collects the values of the tuning factors and corresponding penalized and unpenalized log likelihood values. It has one line for each tuning factor. For a penalty on the canonical eigenvalues,

estimates of the latter are written out as well (in descending order). Again, if this file exists in the working directory it is not over-written but appended to.

7.2.11.4 File PenTargetMatrix

If the option **PHENV** (see Section 4.10.7) is specified, WOMBAT writes out this file with a suitable target matrix. For penalty **COVARM** a covariance matrix and for **CORREL** the corresponding correlation matrix is given. For a multivariate analysis fitting a simple animal model, this is the phenotypic covariance (correlation) matrix. For a random regression analysis, corresponding matrices are based on the sum of the covariance matrices among random regression coefficients due the two random effects fitted, assumed to represent individuals' genetic and permanent environmental effects (which must be fitted using the same number of basis functions).

Written out is the *upper* triangle of the matrix.

7.2.11.5 File ValidateLogLike.dat

This file is the output resulting from a run with the option **--valid**. It contains one line per tuning factor with he following entries:

(a) A running number
(b) The tuning factor
(c) The unpenalized log likelihood in the validation data.
(d) The penalized log likelihood in the training data.
(e) The unpenalized log likelihood in the training data.

If this file exists in the working directory it is not over-written but appended to.

7.2.12 File CovSamples_*name*.dat

This (potentially large) file contains the samples drawn from the multivariate normal distribution of estimates, either for all random effects in the analysis (*name* = ALL) or for single, selected effect. The file contains one line per replicate, with covariance matrices written in the same sequence as in the corresponding estimation run (for ALL), giving the upper triangle for each matrix.

7.3 'Utility' files

In addition, WOMBAT produces a number of small 'utility' files. These serve to monitor progress during estimation, to carry information

over to subsequent runs or to facilitate specialised post-estimation calculations by the user.

7.3.1 File ListOfCovs

This file lists the covariance components defined by the model of analysis, together with their running numbers and starting values given. It is written out during the 'set-up' phase (see Section 5.2.3). It can be used to identify the running numbers needed when defining additional functions of covariance components to be evaluated (see Section 4.10.4)

7.3.2 File RepeatedRecordsCounts

This file gives a count of the numbers of repeated records per trait and, if option **TSELECT** is used, a count of the number of pairs taken at the same time.

7.3.3 File BestPoint

Whenever WOMBAT encounters a set of parameters which improves the likelihood, the currently 'best' point is written out to the file **BestPoint**.

The first line of **BestPoint** gives the following information :

(a) The current value of the log likelihood,
(b) The number of parameters

This is followed by the covariance matrices estimated.

1. Only the upper triangle, written out row-wise, is given.
2. Each covariance matrix starts on a new line. '
3. The first matrix given is the matrix of residual covariances. The other covariance matrices are given in the same order as the matrices of starting values were specified in the parameter file.

N.B.

BestPoint is used in any continuation or post-estimation steps – do not delete is until the analysis is complete !

7.3.4 File Iterates

WOMBAT appends a line of summary information to the file **Iterates** on completion of an iterate of the AI, PX-EM or EM algorithm. This can be used to monitor the progress of an estimation

run – useful for long runs in background mode. Each line gives the following information :

(a) Column 1 gives a two-letter identifying the algorithms (AI, PX, EM) used.
(b) Column 2 gives the running number of the iterate.
(c) Column 3 contains the log likelihood value at the end of the iterate.
(d) Column 4 gives the change in log likelihood from the previous iterate.
(e) Column 5 shows the norm of the vector of first derivatives of the log likelihood (zero for PX and EM)
(f) Column 6 gives the norm of the vector of changes in the parameters, divided by the norm of the vector of parameters.
(g) Column 7 gives the Newton decrement (absolute value) for the iterate (zero for PX and EM).
(h) Column 8 shows a) the step size factor used for AI steps, b) the average squared deviation of the matrices of additional parameters in the PX-EM algorithm from the identity matrix, c) zero for EM steps.
(i) Column 9 gives the CPU time for the iterate in seconds
(j) Column 10 gives the number of likelihood evaluations carried out so far.
(k) Column 11 gives the factor used to ensure that the average information matrix is 'safely' positive definite
(l) Column 12 identifies the method used to modify the average information matrix (0: no modification, 1: modify eigenvalues directly, 2: add diagonal matrix, 3: modified Cholesky decomposition, 4: partial Cholesky decomposition – see Section A.5).

7.3.5 File `OperationCounts`

This small file gathers accumulates the number of non-zero elements in the Cholesky factor (or inverse) of the mixed model matrix together with the resulting operation count for the factorisation. This can be used to compare the efficacy of different ordering strategies for a particular analysis. The file contains one line per ordering tried, with the following information :

(a) The name of the ordering strategy (**`mmd`**, **`amd`** or **`metis`**).
(b) For **`metis`** only : the values of the three options which can be set by the user, i.e. the number of graph separators used, the 'density' factor, and the option selecting the edge matching strategy (see Section 5.3.1).
(c) The number of non-zero elements in the mixed model matrix after factorisation.

(d) The operation count for the factorisation.

7.3.6 Files AvInfoParms and AvinfoCovs

These files are written out when the AI algorithm is used. After each iterate, they give the average information matrix (not its inverse !) corresponding to the 'best' estimates obtained by an AI step, as written out to **BestPoint**. These can be used to approximate sampling variances and errors of genetic parameters.

N.B.

If the AI iterates are followed by further estimates steps using a different algorithm, the average information matrices given may not pertain to the 'best' estimates any longer.

AvInfoParms contains the average information matrix for the parameters estimated. Generally, the parameters are the elements of the leading columns of the Cholesky factors of the covariance matrices estimated. This file is written out for both full and reduced rank estimation.

For full rank estimation, the average information is first calculated with respect to the covariance components and then transformed to the Cholesky scale. Hence, the average information for the covariances is available directly, and is written to the file **AvInfoParms**.

Both files give the elements of the upper triangle of the symmetric information matrix row-wise. The first line gives the log likelihood value for the estimates to which the matrix pertains – this can be used to ensure corresponding files of estimates and average information are used. Each of the following lines in the file represents one element of the matrix, containing 3 variables :

(a) row number,
(b) column number, and
(c) element of the average information matrix.

N.B.

Written out are the information matrices for all parameters. If some parameters (or covariances) are not estimated (such as zero residual covariances for traits measured on different animals), the corresponding rows and columns may be zero.

7.3.7 Files Covariable.baf

For random regression analyses, file(s) with the basis functions evaluated for the values of the control variable(s) in the data are written out. These can be used, for example, in calculating covariances of predicted random effects at specific points.

The name of a file is equal to the name of the covariable (or 'control' variable), as given in the parameter file (model of analysis part), followed by the option describing the form of basis function (**POL**, **LEG**, **BSP**; see Section 4.8.1.1) the maximum number of coefficients, and the extension **.baf**. The file then contains one row for each value of the covariable, giving the covariable, followed by the coefficients of the basis function.

NB. These files pertain to the *random* regressions fitted! Your model may contain a fixed regression on the same covariable with the same number of specified regression coefficients, n, but with intercept omitted. If so, the coefficients in this file are not appropriate to evaluate the fixed regression curve.

7.3.8 File LogL4Quapprox.dat

For analyses involving an additional parameter, the values used for the parameter and the corresponding maximum log likelihoods are collected in this file. This is meant facilitate estimation of the parameter through a quadratic approximation of the resulting profile likelihood curve via the run option **--quapp**. Note that this file is appended to at each run.

7.3.9 File SubSetsList

If analyses considering a subset of traits are carried out, WOMBAT writes out files **EstimSubset**$n+\ldots+m$**.dat** (see Section 7.2.6), to be used as input files in a run with option **--itsum**. In addition, for each run performed, this file name this appended to **SubSetsList**. This file contains one line per 'partial' run with two entries: the file name (**EstimSubset**$n+\ldots+m$**.dat**) and a weight given to the corresponding results when combining estimates. The default for the weight is unity.

7.4 Miscellaneous

7.4.1 File ReducedPedFile.dat

As one of the first steps in analyses fitting an animal model, WOMBAT checks the pedigree file supplied against the data file and, if applicable, deletes any individuals not linked to the data in any way. The new, reduced pedigree is written to this file. Like the original pedigree file, it contains 3 columns, i.e. codes for animal, sire and dam.

7.4.2 Files PrunedPedFile$n$.dat

The second 'pedigree modification' carried out is 'pruning'. This is performed for each genetic effect separately (provided they are assumed to be uncorrelated). The corresponding pedigrees are written to these files, with $n = 1, 2, \ldots$ pertaining to the order in which these effects are specified in the model part of the parameter file. Each has 7 columns: columns 1 to give the animal, sire and dam recoded in running order, columns 4 to 6 give the corresponding original identities, and column 7 contains the inbreeding coefficient of the individual.

7.4.3 File WOMBAT.log

This file collects 'time stamps' for various stages of a run, together explanatory messages for programmed stops. The content largely duplicates what is written to the screen. However, the file is closed after any message written – in contrast to a capture of the screen output which, under Linux, may be incomplete. It is intended for those running a multitude of analyses and wanting to trace hat went on. N.B. This file is appended too, i.e. will collect information for multiple runs in the current directory if not deleted between runs.

8 ‘Internal’ files used by WOMBAT

WOMBAT generates a number of files for ‘internal’ use only. All of these are binary files, and have default names and the extension **.bin**. A number of these files are re-used in continuation runs or additional runs for the same analysis if a matching file is found, thus saving computational steps.

Files created are

nrminv$n$**.bin** : This file contains the inverse of the numerator relationship matrix, with $n = 1, 2$.

adjacency.bin : This file contains the adjacency structure of the mixed model matrix.

eqnsorder.bin : This file contains information on the best order of equations in the mixed model equations.

symbfact.bin : This file gives the structure (non-zero elements) of the mixed model matrix after factorisation, together with the vectors determining the sparse matrix compressed storage scheme.

recoded$n$**.bin** : These files, with $n = 1, 4$ contain the data in a recoded form.

Hint

Checks for a match between an existing **.bin** file and the current model of analysis and data file are rather elementary and should not be relied upon. When starting a ‘new’ analysis, it is good practice to switch to a new, ‘clean’ directory, copying any **.bin** files which are known to match, if appropriate.

9 Worked examples

A number of worked examples are provided to illustrate the use of WOMBAT and, in particular, show how to set up the parameter files.

Installation for the suite of examples is described in section 3.1.4. This generates the directory **WOMBAT/Examples** with subdirectories **Example**n ($n = 1, \ldots, 9$).

Each subdirectory contains the data and pedigree files for a particular example, a file **WhatIsIt** with a brief description of the example, and one or subdirectories for individual runs, **A**, **B**, **C**, ...

Each 'run' directory (**A**, **B**, ...) contains :

(a) A parameter file (**.par**)
(b) The file **typescript** (generated using the **script** command) which contains the screen output for the run.
Run time options used can be found at the top of this file.
(c) The numerous output files generated by WOMBAT.

N.B.
The example data sets have been selected for ease of demonstration, and to allow fairly rapid replication of the example runs. Clearly, most of the data sets used are too small to support estimation of all the parameters fitted, in particular for the higher dimensional analyses shown !

N.B.
Examples runs have been carried out on a 64-bit machine (Intel I7 processor, rated at 3.20Ghz) ; numbers obtained on 32-bit machine may vary slightly.
Note further that all the example files are Linux files - you may need to 'translate' them to DOS format if you plan to run the examples under Windows.

Example 1

This shows a univariate analysis for a simple animal model, fitting a single fixed effect only.

Source: Simulated data; Example 1 from DfReml

Example 2

This shows a bivariate analysis for the case where the same model is fitted for both traits and all traits are recorded for all animals. The model of analysis includes 3 cross-classified fixed effects and an additional random effect

A Analysis fitting an animal model
B Analysis fitting a sire model

Source: Data from Edinburgh mouse lines; Example 2 from DfReml

Example 3

This example involves up to six repeated records for a single trait, recorded at different ages. The model of analysis is an animal model with a single fixed effect. Data are analysed :

A Fitting a univariate 'repeatability' model, with age as a covariable
B Fitting a multivariate analysis with 6 traits
C Fitting a univariate random regression model

Source: Wokalup selection experiment; Example 3 from DfReml

Example 4

This example shows a four-variate analysis for a simple animal model. Runs show:

A A 'standard' full rank analysis
B A reduced rank analysis, fitting the first two principal components only
C A reduced rank analysis using the EM algorithm

Source: Australian beef cattle field data

Example 5

Similar to example 4, but involving 6 traits. Runs show:

A A 'standard' full rank analysis
B A reduced rank analysis, fitting the first four principal components
C An analysis fitting a factor-analytic structure for the genetic covariance matrix
D A full rank analysis, illustrating use of penalized REML ('bending') for a chosen tuning parameter

Source: Australian beef cattle data

Example 6

This example involves 4 measurements, with records on different sexes treated as different traits. This gives an eight-variate analysis, with 16 residual covariances equal to zero. The model of analysis is a simple animal model.

A A full rank analysis with 'good' starting values
B A full rank analysis with 'bad' starting values

Source: Australian beef cattle field data

Example 7

This example illustrates the analysis of 4 traits, subject to genetic and permanent environmental effects. The model of analysis involves several crossclassified fixed effects, nested covariables and different effects for different traits.

A Univariate analysis for trait 1
B Univariate analysis for trait 2
B1 As B but reworked by setting up NRM inverse externally to illustrate use of GIN option
B2 As B, but fitting fixed effects only and a user-defined basis function for the covariable dam age
C Univariate analysis for trait 2, allowing for a non-zero direct-maternal genetic covariance
C1 As C but using GIN instead of NRM option
D Bivariate analysis for traits 1 and 2
E Bivariate analysis for traits 1 and 2, allowing for a non-zero direct-maternal genetic covariance
F Trivariate analysis for traits 1, 2 and 3
G Fourvariate analysis of all traits
H Fourvariate analysis of all traits, not fitting maternal effects for trait 4
I Reduced rank, fourvariate analysis of all traits, not fitting maternal effects for trait 4

Source: Wokalup selection experiment

Example 8

This is an example of a model where different random effects are fitted for different traits. It is a bivariate analysis of mature cow weights together with gestation length. Mature cow weight involves repeated records per animal, and a permanent environmental effect of the animal is thus fitted for this trait. Gestation length is treated as trait of

the calf and assumed to be affected by both genetic and permanent environmental effects.

A Standard model
B Equivalent model, using the **PEQ** option for permanent environmental effects of the animal.

Source: Australian beef cattle field data

Example 9

This example illustrates random regression analyses fitting an additional random effect, using B-splines as basis functions and imposing rank restrictions on estimated covariance functions. Data are monthly records for weights of calves from birth to weaning.

A Full rank analysis
A1 As A, but evaluating the basis functions externally to illustrate the use of user-defined basis functions.
B Reduced rank analysis
C Reduced rank analysis with different ranks

Source: Wokalup selection experiment

Example 10

In a recent paper, Wilson et al. [40] presented a tutorial on 'animal model' analyses of data from a wild population. This example replicates the analyses shown (and expands it by demonstrating how to evaluate likelihood fixing the genetic covariance at zero, in order to carry out a likelihood ratio test for this component).

The tutorial and all data (& pedigree) files used in this example are **not** included – you have to download these from their web site:

http://wildanimalmodels.org

A Simple univariate analysis
B Bivariate analysis
B1 Bivariate analysis, fixing the genetic covariance at zero
C Repeated records model

Source: The Wild Animal Models Wiki

Example 11

This example demonstrates simple bi-variate random regression analyses.

A Using a RR model to carry out a multivariate analysis with repeated records where the pattern of temporary environmental covariances is determined through the control variable.

A1 Multivariate analysis corresponding to A

B Bi-variate RR analysis fitting a quadratic regression on Legendre polynomials and homogeneous measurement error variances.

Source: Simulated records

Example 13

This example illustrates multivariate analyses with repeated records per trait, especially some new features:

- WOMBAT now insists that the option **RPTCOV** is specified in the parameter file!
- WOMBAT writes out a file **RepeatedRecordsCounts** with some basic information on how many animals have how many records.
- There is now a mechanism – through **RPTCOV TSELECT** – to specify which records are measured at the same time and thus have a non-zero error covariance and which are not.
- Trait numbers need to be assigned so that any traits with repeated records have a lower number than traits with single records.

The data were obtained by simulating records for 4 traits recorded on 800 animals at 5 different times. A missing value indicator (999) is used to create different pattern of missing records - note that analyses in the different sub-directories analyze different columns in the data file.

A Demonstrates an analysis without missing records, i.e. where all traits are recorded at the same time. This implies that there are non-zero error covariances between all traits and that the **ALIGNED** option is appropriate.

B Shows the analysis when some records are missing, but in a systematic fashion: Traits 1 and 2 have records at all 5 times, but traits 3 and 4 are only recorded for times 1 and 2. As the 'missing' observations only occur for the later times, the option **ALIGNED** is still appropriate.

C Similar to B, but measurements for traits 3 and 4 are taken at times 2 and 4. This means that a time of recording indicator needs to be used to model the residual covariance structure correctly. This is done specifying **TSELECT** together with the name of the column in the data file which contains the time variable.

C1 As C, but using a multivariate random regression analysis.

D Illustrates the scenario where we have a trait with repeated records analysed together with traits with single records and where traits with single and repeated records are measured at different times so that

i) there are no error covariances between these groups of traits and

ii) that we can 'use' the error covariance to model covariances between traits due to permanent environmental effects of the animal.

For this example, we use records taken at times 1 to 4 for trait 1, and records taken at time 5 for traits 2 to 4. For this case a model fitting a permanent environmental effects due to the animal for trait 1 only together with the **INDESCR** option is appropriate. Estimates of the error covariances betwen trait 1 and traits 2, 3 and 4 then reflect the permanent environmental covariance, while estimates of the (co)variances among the latter represent the sum of temporary and permanent environmental covariances.

E Shows the case where we have a trait with repeated records analysed together with traits with single records, but where the single records are taken at the same time as one of the repeated records, so that we need to model non-zero error covariances. Here we consider records for trait 1 at all 5 times, and records for traits 2 to 3 taken at time 5. Again we need the **TSELECT** option to model this properly. In addition, we need to use the equivalent model invoked via the **PEQ** option in order to separate temporary and permanent envirionmental covariances between trait 1 and the other traits. Note that permanent environmental effects are fitted for all 4 traits, but that only the corresponding covariance components which can be disentangled from the environmental covariances are reported.

Source: Simulated records

Example 14

This gives a toy example illustrating the use of the run option **--snap** for a simple GWAS type analysis.

Source: Simulated records

Example 15

This example illustrates how to pool estimates of covariance components by (penalized) maximum likelihood.

The example is comprised of 14 traits, with 6 traits measured on few

animals and the remaining records representing 4 traits with measures on males and females treated as different traits [from 25]. There are results from 76 bivariate and one six-variate analysis to be combined. Due to traits measured on different subsets of animals (sexes) there are a number of residual covariances which are to be fixed at zero.

A All part analyses have been carried out using WOMBAT and a 'full' parameter file is available.
B Results from part analyses are summarized in a single file and a 'minimum' parameter file is used.

Example 16

This example shows what options are available in WOMBAT to fit 'social interaction' type models.

A This directory shows an example run for simulated data for a dilution factor of 0, treating direct and social genetic effects as uncorrelated.
B As A, but allowing for a non-zero genetic covariance
C As B, but treating social group and residual variances as heterogeneous.
D Fitting the same model as in B to data simulated with a non-zero dilution factor, this directory shows the multiple runs required to estimated this factor using a quadratic approximation to the profile likelihood.
Z Larry Schaeffer has some notes which contain a very simple example. This subdirectory shows how to obtain the BLUP solutions given using WOMBAT. www.aps.uoguelph.ca/ lrs/ABModels/NOTES/SSocial.pdf

Example 17

This example shows how WOMBAT can be used to approximate the sampling distribution of estimates by sampling from their asymptotic, multivariate normal distribution, as described by Meyer and Houle [27].

A Sample estimates for the genetic covariance matrix in a 5-trait analysis (Note that 100 samples is used for illustration only - more should be used in practice!)

Example 18

This example illustrates the use of the single-step option, **`--s1step`**

A Univariate analysis

B As A, using –blup
C Bivariate analysis
D Univariate analysis, fitting "explicit" genetic groups

Example 19

This example illustrates the use of *simple* penalties to reduce sampling variances in REML estimation.

0 Parameter file to simulate data

A Standard unpenalized analysis
B Unpenalized analysis, parameterised to elements of the canonical decomposition
C Penalty on canaonical eigenvalues with ESS=8
D Penalty on genetic partial correlations with ESS=8, shrinking towards phenotypic values

Z Script to simulate replicates & Fortran code to summarize results

A Technical details

This chapter explains some of the assumptions made, strategies and statistical procedures used in WOMBAT.

A.1 Ordering strategies

An essential set-up set in REML analyses is to find a permutation of the rows and columns in the coefficient matrix of the mixed model equations that makes the 'fill-in' arising during factorisation small and thus minimises the computational efforts per likelihood evaluation and REML iterate. This needs to be done only once per analysis (WOMBAT saves the results from this step for re-use in any subsequent steps). As it can have a dramatic impact on the time and memory required per analysis, it is well worth considerable effort to find the 'best' order. Especially for analyses involving large data sets or multiple random effects, the time spend trying several, or even numerous alternatives is readily recouped within the first few iterates [23]. WOMBAT selects a default ordering strategy based on the number of equations in the analysis.

Three different stratgies are implemented :

1. The multiple minimum degree procedure [18] as implemented in the widely used public domain subroutine genmmd. This is the strategy which has been used in DfReml. For WOMBAT, it is the default for small analyses involving up to $25\,000$ equations.
2. The approximate minimum degree ordering of Amestoy et al. [1]. This tends to produce orderings of similar quality to the multiple minimum degree procedure, but is considerably faster. Implementation is through the public domain subroutine amd (version 1.1) available at

www.cise.ufl.edu/research/sparse/amd. This is the default for analyses involving more than $25\,000$ and up to $50\,000$ equations.

3. A multilevel nested dissection procedure, as implemented in subroutine
   metis_nodend from the MeTiS package (public domain) of Karypis and Kumar [16], available at at **www.cs.umn.edu/∼karypis/metis**. This is the default for large analyses.

A.2 Convergence citeria

𝕎𝕆𝕄𝔹𝔸𝕋 calculates four different criteria to determine whether an analysis has converged. The first two are simple changes, available for all maximisation algorithms, the other two are based on the derivatives of the log likelihood function, i.e. can only be obtained for the AI algorithm. The criteria are :

1. The increase in log likelihood values between subsequent iterates, i.e. $\log \mathcal{L}^t - \log \mathcal{L}^{t-1}$, with $\log \mathcal{L}^t$ the log likelihood for iterate t.
2. The change in the vector of parameter estimates from the last iterate. This is evaluated as

$$\sqrt{\sum_{i=1}^{p} \left(\hat{\theta}_i^t - \hat{\theta}_i^{t-1}\right)^2 / \sum_{i=1}^{p} \left(\hat{\theta}_i^t\right)^2} \tag{A.1}$$

   where $\hat{\theta}_i^t$ denotes the estimate of the $i-$th parameter from iterate t, and p is the number of parameters.
3. The norm of the gradient vector, i.e., for $g_i^t = \partial \log \mathcal{L}^t / \partial \theta_i$,

$$\sqrt{\sum_{i=1}^{p} \left(g_i^t\right)^2} \tag{A.2}$$

4. The 'Newton decrement', i.e.

$$-\sum_{i=1}^{p} \sum_{j=1}^{p} g_i^t g_j^t H_{ij}^t \tag{A.3}$$

   where H_{ij}^t is the $ij-$the element of the inverse of the average information matrix for iterate t. This gives a measure of the expected difference of $\log \mathcal{L}^t$ from the maximum, and has been suggested as an alternative convergence criterion [5].

Default values for the thresholds for these criteria used in 𝕎𝕆𝕄𝔹𝔸𝕋

Table A.1: Default thresholds for convergence criteria

	Algorithm		
Criterion	AI	(PX-) EM	Powell
Change in $\log \mathcal{L}$	$< 5 \times 10^{-4}$	$< 10^{-5}$	$< 10^{-4}$
Change in parameters	$< 10^{-8}$	$< 10^{-8}$	$< 10^{-8}$
Norm of gradient vector	$< 10^{-3}$	–	–
Newton decrement	not used	–	–

are summarised in Table Table A.1.

N.B.
Current values used are rather stringent; 'softer' limits combining several criteria may be more appropriate for practical estimation.

A.3 Parameterisation

A.4 Approximation of sampling errors

A.4.1 Sampling covariances

At convergence, the inverse of the AI matrix gives estimates of lower bound sampling covariances among the parameters estimated. These are used to approximate sampling errors of covariance components and genetic parameters.

For full rank analyses parameterising to the elements of the Cholesky factors of the corresponding covariance matrices, the AI matrix is obtained by first calculating the AI matrix for the covariance components. This is then transformed to the AI matrix for the parameters to be estimated by pre- and postmultiplying it with the corresponding Jacobian and its transpose, respectively. Full details are given in Meyer and Smith [31]. This implies that sampling covariances among the covariance components can be obtained directly by simply inverting the corresponding AI matrix.

For reduced rank analyses, however, the AI matrix for the parameters to be estimated are calculated directly, as outlined by Meyer and Kirkpatrick [28]. Hence, sampling covariances among the corresponding covariance components need to be approximated from the inverse of the AI matrix. WOMBAT estimates the leading columns of the Cholesky factor ($\mathbf{L}$) of a matrix to obtain a reduced rank estimate, $\mathbf{\Sigma} = \mathbf{L}\mathbf{L}'$. Let l_{ir} (for $i \leq r$) denote the non-zero elements of $\mathbf{L}$. The

inverse of the AI matrix then gives approximate sampling covariances $Cov\,(l_{ir}, l_{js})$. The $ij-$th covariance component, σ_{ij}, is

$$\sigma_{ij} = \sum_{r=1}^{q(i,j)} l_{ir}\; l_{jr}$$

with $q(i,j) = min(i,j,t)$ and t the rank which the estimate of $\mathbf{\Sigma}$ is set to have. The covariance between two covariances, σ_{ij} and σ_{km} is then

$$\mathrm{Cov}\,(\sigma_{ij}, \sigma_{kl}) = \sum_{r=1}^{q(i,j)} \sum_{s=1}^{q(k,m)} \mathrm{Cov}\,(l_{ir}l_{jr}, l_{ks}l_{ms})$$

Using a first order Taylor series approximation to the product of two variables, this can be approximated as

$$\begin{aligned}\mathrm{Cov}\,(\sigma_{ij}, \sigma_{kl}) \approx \sum_{r=1}^{q(i,j)} \sum_{s=1}^{q(k,m)} [\, & l_{jr}l_{ms}\,\mathrm{Cov}\,(l_{ir}, l_{ks}) + l_{jr}l_{ks}\,\mathrm{Cov}\,(l_{ir}, l_{ms}) \\ & + l_{ir}l_{ms}\,\mathrm{Cov}\,(l_{jr}, l_{ks}) + l_{ir}l_{ks}\,\mathrm{Cov}\,(l_{jr}, l_{ms})\,]\end{aligned} \tag{A.4}$$

Equation A.4 extends readily to two covariance components belonging to different covariance matrices, $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$, and their respective Cholesky factors.

A.4.2 Sampling errors of genetic parameters

Let $\boldsymbol{\sigma}$ denote the vector of covariance components in the model of analysis and $\mathbf{V}(\boldsymbol{\sigma})$ its approximate matrix of sampling covariances, obtained as described above (Section A.4.1). The sampling covariance for any pair of linear functions of $\boldsymbol{\sigma}$ is then simply

$$\mathrm{Cov}\,(\mathbf{w}_1'\boldsymbol{\sigma}, \mathbf{w}_2'\boldsymbol{\sigma}) = \mathbf{w}_1'\mathbf{V}(\boldsymbol{\sigma})\mathbf{w}_2 \tag{A.5}$$

with $\mathbf{w}_i$ the vector of weights in linear function i.

Sampling variances of non-linear functions of covariance components are obtained by first approximating the function by a first order Taylor series expansion, and then calculating the variance of the resulting linear function. For example, for a variance ratio

$$\mathrm{Var}\left(\frac{\sigma_1^2}{\sigma_2^2}\right) \approx \left[\sigma_2^4\,\mathrm{Var}(\sigma_1^2) + \sigma_1^4\,\mathrm{Var}(\sigma_2^2) - 2\sigma_1^2\sigma_2^2\,\mathrm{Cov}(\sigma_1^2, \sigma 2^2)\right]/\sigma_2^8 \tag{A.6}$$

Similarly, for a correlation

$$\begin{aligned}\operatorname{Var}\left(\frac{\sigma_{12}}{\sqrt{\sigma_1^2\sigma_2^2}}\right) \approx \Big[& 4\sigma_1^4\sigma_2^4 \operatorname{Var}(\sigma_{12}) + \sigma_{12}^2\sigma_2^4 \operatorname{Var}(\sigma_1^2) + \sigma_{12}^2\sigma_1^4 \operatorname{Var}(\sigma_2^2) \\ & - 4\sigma_{12}\sigma_1^2\sigma_2^4 \operatorname{Cov}(\sigma_{12}, \sigma_1^2) - 4\sigma_{12}\sigma_1^4\sigma_2^2 \operatorname{Cov}(\sigma_{12}, \sigma_2^2) \\ & + 2\sigma_{12}^2\sigma_1^2\sigma_2^2 \operatorname{Cov}(\sigma_1^2, \sigma_2^2)\Big] / \left(4\sigma_1^6\sigma_2^6\right) \end{aligned} \tag{A.7}$$

A.5 Modification of the average information matrix

To yield a search direction which is likely to improve the likelihood, or, equivalently, decrease $-\log \mathcal{L}$, the Hessian matrix or its approximation in a Newton type optimisation strategy must be positive definite. While the AI matrix is a matrix of sums of squares and crossproducts and thus virtually guaranteed to be positive definite, it can have a relatively large condition number or minimum eigenvalues close to zero. This can yield step sizes, calculated as the product of the inverse of the AI matrix and the vector of first derivatives, which are too large. Consequently, severe step size modifications may be required to achieve an improvement $\log \mathcal{L}$. This may, at best, require several additional likelihood evaluations or cause the algorithm to fail. Modification of the AI matrix, to ensure that it is 'safely' positive definite and that its condition number is not excessive, may improve performance of the AI algorithm in this instance.

Several strategies are available. None has been found to be 'best'.

1. Schnabel and Estrow [37] described a modified Cholesky decomposition of the Hessian matrix. This has been implemented using algorithm 695 of the TOMS library (**www.netlib.org/toms**. This is the) [9], but using a factor of $\epsilon^{-1/2}$ (where ϵ denotes machine precision) to determine the critical size of pivots, which is intermediate to the original value of $\epsilon^{-2/3}$ and the value of $\epsilon^{1/3}$ suggested by Schnabel and Estrow [38].
2. A partial Cholesky decomposition has been suggested by Forsgren et al. [12]. This has been implemented using a factor of $\nu = 0.998$.
3. Modification strategies utilising the Cholesky decomposition have been devised for scenarios where direct calculation of the eigenvalues is impractical. For our applications,however, computational costs of an eigenvalue decomposition of the AI matrix are negligible compared to those of a likelihood evaluation. This allows a modification where we know the minimum eigenvalue of the resulting matrix. Nocedahl and Wright [34, Chapter 6] described two variations, which have been implemented.

(a) Set all eigenvalues less than a value of δ to δ, and construct the modified AI matrix by pre- and postmultiplying the diagonal matrix of eigenvalues with the matrix of eigenvectors and its transpose, respectively.

(b) Add a diagonal matrix $\tau\mathbf{I}$ to the AI matrix, with $\tau = max(0, \delta - \lambda_{min})$ and λ_{min} the smallest eigenvalue of the AI matrix. This has been chosen as the default procedure, with δ bigger than $3 \times 10^{-6} \times \lambda_1$, and λ_1 the largest eigenvalue of the AI matrix.

Choice of the modification can have a substantial effect on the efficiency of the AI algorithm. In particular, too large a modification can slow convergence rates unnecessarily. Further experience is necessary to determine which is a good choice of modification for specific cases.

A.6 Iterative summation of expanded part matrices

Consider the $q \times q$ covariance matrix $\mathbf{V}$ (among q traits) for a random effect, e.g. additive genetic effects. Assume we have S analyses of subsets of traits, with the $s-$th analysis comprising k_s traits. Further, let $\mathbf{C}_s$ denote the $k_s \times k_s$ matrix of estimates of covariance components for the random effect from analysis s. The pooled matrix $\mathbf{V}$ is constructed by iterating on

$$\mathbf{V}^{t+1} = \sum_{s=1}^{S} w_s \Big\{ \mathbf{V}^t \left(\mathbf{P}_s \mathbf{P}_s{}' \mathbf{V}^t \mathbf{P}_s \mathbf{P}_s{}'\right)^- \mathbf{P}_s \mathbf{C}_s \mathbf{P}_s{}' \left(\mathbf{P}_s \mathbf{P}_s{}' \mathbf{V}^t \mathbf{P}_s \mathbf{P}_s{}'\right)^- \mathbf{V}^t + \left[\left(\mathbf{I} - \mathbf{P}_s \mathbf{P}_s{}'\right) \left(\mathbf{V}^t\right)^{-1} \left(\mathbf{I} - \mathbf{P}_s \mathbf{P}_s{}'\right)\right]^- \Big\} / \sum_{s=1}^{S} w_s \quad \text{(A.8)}$$

until $\mathbf{V}^t$ and $\mathbf{V}^{t+1}$ are virtually identical, with $\mathbf{V}^t$ the value of $\mathbf{V}$ for the $t-$th iterate.

Other components of (Equation A.8) are w_s, the weight given to analysis s, $\mathbf{I}$, an identity matrix of size $q \times q$, and transformation matrix $\mathbf{P}_s$ for analysis s, of size $q \times k_s$. $\mathbf{P}_s$ has k_s elements of unity, $p_{ij} = 1$, if the i-th trait overall is the j-th trait in analysis s, and zero otherwise.

Starting values ($\mathbf{V}^0$) are obtained initially by decomposing covariance matrices $\mathbf{C}_s$ into variances and correlations, and calculating simple averages over individual analyses. If the resulting correlation matrix is not positive definite, it is modified by regressing all eigenvalues towards their mean, choosing a regression factor so that the smallest eigenvalue becomes 10^{-6}, and pre- and post-multiplying the diagonal matrix of modified eigenvalues with the matrix of eigenvectors and

its transpose, respectively. Using the average variances, $\mathbf{V}^0$ is then obtained from the (modified) average correlation matrix. $\mathbb{WOMBAT}$ is set up to perform up to $100\,000$ iterates. Convergence is assumed to be reached when

$$\frac{2}{q(q+1)} \sum_{i=1}^{q} \sum_{j=i}^{q} \left(v_{ij}^{t+1} - v_{ij}^{t}\right)^2 \leq 10^{-7}$$

with v_{ij}^t denoting the $ij-$th element of $\mathbf{V}^t$.

Bibliography

[1] Amestoy P.R., Davis T.A., Duff I.S. An approximate minimum degree ordering algorithm. SIAM J. Matr. Anal. Appl. 17 (1996) 886–905.

[2] Anderson E., Bai Z., Bischof C., Blackford S., Demmel J., Dongarra J., Du Croz J., Greenbaum A., Hammarling S., McKenney A., Sorensen D. LAPACK Users' Guide. Society for Industrial and Applied Mathematics, Philadelphia, PA, Third edn. (1999). ISBN 978-0-89871-447-0.

[3] Bijma P. Multilevel selection 4: Modeling the relationship of indirect genetic effects and group size. Genetics 186 (2010) 1029–1031. doi: 10.1534/genetics.110.120485.

[4] Bondari K., Willham R.L., Freeman A.E. Estimates of direct and maternal genetic correlations for pupa weight and family size of Tribolium. J. Anim. Sci. 47 (1978) 358–365.

[5] Boyd S., Vandenberghe L. Convex Optimization. Cambridge University Press (2004).

[6] Dennis J.E., Schnabel R.B. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. SIAM Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia (1996).

[7] Dongarra J.J., Croz J.D., Hammarling S., Hanson R.J. An extended set of FORTRAN Basic Linear Algebra Subprograms. ACM Trans. Math. Softw. 14 (1988) 1–17. doi: 10.1145/42288.42291.

[8] Erisman A.M., Tinney W.F. On computing certain elements of the inverse of a sparse matrix. Commun. ACM 18 (1975) 177–179. ISSN 0001-0782. doi: 10.1145/360680.360704.

[9] Eskow E., Schnabel R.B. Algorithm 695: Software for a new modified Cholesky factorization. ACM Trans. Math. Software 17 (1991) 306–312.

[10] Fernando R.L., Grossman M. Genetic evaluation with autosomal and X-chromosomal inheritance. Theor. Appl. Genet. 80 (1990) 75–80. doi: 10.1007/BF00224018.

[11] Ferris M., Lucidi S., Roma M. Nonmonotone curvilinear line search methods for unconstrained optimization. Comput. Optim. Applic. 6 (1996) 117–136.

[12] Forsgren A., Gill P.E., Murray W. Computing modified Newton directions using a partial Cholesky factorization. SIAM J. Sci. Statist. Comp. 16 (1995) 139–150.

[13] Grippo L., Lampariello F., Lucidi S. A nonmonotone line search technique for

Newton's method. SIAM J. Numer. Anal. 23 (1986) 707–716. ISSN 0036-1429. doi: 10.1137/0723046.

[14] Gustavson F., Waśniewski J., Dongarra J., Langou J. Rectangular full packed format for Cholesky's algorithm: factorization, solution, and inversion. ACM Trans. Math. Softw. 37 (2010) 18. doi: 10.1145/1731022.1731028.

[15] Ihaka R., Gentleman R. R: A language for data analysis and graphics. J. Comp. Graph. Stat. 5 (1996) 299–314.

[16] Karypis G., Kumar V. MeTis A software package for partioning unstructured graphs, partitioning meshes, and computing fill-in reducing ordering of sparse matrices Version 4.0. Department of Computer Science, University of Minnesota, Minneapolis, MN 55455 (1998). 44 pp.

[17] Koivula M., Negussie E., Mäntysaari E.A. Genetic parameters for test-day somatic cell count at different lactation stages of Finnish dairy cattle. Livest. Prod. Sci. 90 (2004) 145–157.

[18] Liu J.W.H. Modification of the minimum degree algorithm by multiple elimination. ACM Trans. Math. Soft. 11 (1985) 141–153.

[19] Mäntysaari E.A. Derivation of multiple trait reduced random regression (RR) model for the first lactation test day records of milk, protein and fat. In: Proceedings of the 50th Annual Meeting of the European Association of Animal Production. Europ. Ass. Anim. Prod. (1999).

[20] Meuwissen T.H.E., Luo Z. Computing inbreeding coefficients in large populations. Genet. Sel. Evol. 24 (1992) 305–313.

[21] Meyer K. DfReml — a set of programs to estimate variance components under an individual animal model. In: Proceedings Animal Model Workshop, vol. 71 Supplement 2 of *J. Dairy Sci.* Edmonton, Canada, June 25–26, 1988 (1988), pp. 33–34. doi: 10.1016/S0022-0302(88)79977-4.

[22] Meyer K. DfReml version 3.0. CD-ROM of the Sixth World Congress on Genetics Applied to Livestock Production (1998).

[23] Meyer K. Ordering strategies to reduce computational requirements in variance component estimation. Proc. Ass. Advan. Anim. Breed. Genet. 16 (2005) 282–285.

[24] Meyer K. Random regression analyses using B–splines to model growth of Australian Angus cattle. Genet. Sel. Evol. 37 (2005) 473–500. doi: 10.1051/gse:2005012.

[25] Meyer K. Multivariate analyses of carcass traits for Angus cattle fitting reduced rank and factor–analytic models. J. Anim. Breed. Genet. 124 (2007) 50–64. doi: 10.1111/j.1439-0388.2007.00637.x.

[26] Meyer K. A penalized likelihood approach to pooling estimates of covariance components from analyses by parts. J. Anim. Breed. Genet. 130 (2013) 270–285. doi: 10.1111/jbg.12004.

[27] Meyer K., Houle D. Sampling based approximation of confidence intervals for functions of genetic covariance matrices. Proc. Ass. Advan. Anim. Breed. Genet. 20 (2013) 523–526.

[28] Meyer K., Kirkpatrick M. Restricted maximum likelihood estimation of genetic principal components and smoothed covariance matrices. Genet. Sel. Evol. 37 (2005) 1–30. doi: 10.1051/gse:2004034.

[29] Meyer K., Kirkpatrick M. Better estimates of genetic covariance matrices by 'bending' using penalized maximum likelihood. Genetics 185 (2010) 1097–1110. doi: 10.1534/genetics.109.113381.

[30] Meyer K., Kirkpatrick M., Gianola D. Penalized maximum likelihood estimates of genetic covariance matrices with shrinkage towards phenotypic dispersion. Proc. Ass. Advan. Anim. Breed. Genet. 19 (2011) 87–90.

[31] Meyer K., Smith S.P. Restricted maximum likelihood estimation for animal models using derivatives of the likelihood. Genet. Sel. Evol. 28 (1996) 23–49. doi: 10.1051/gse:19960102.

[32] Meyer K., Tier B. "SNP Snappy": A strategy for fast genome wide association studies fitting a full mixed model. Genetics 190 (2012) 275–277. doi: 10.1534/genetics.111.134841.

[33] Nelder J.A., Mead R. A simplex method for function minimization. Computer J. 7 (1965) 308–313.

[34] Nocedahl J., Wright S.J. Numerical Optimization. Springer Series in Operations Research. Springer Verlag, New York, Berlin Heidelberg (1999). ISBN 0-38798793-2.

[35] Powell M.J.D. An efficient method for finding the minimum of a function of several variables without calculating derivatives. Computer J. 7 (1965) 155–162.

[36] Quaas R.L. Computing the diagonal elements of a large numerator relationship matrix. Biometrics 32 (1976) 949–953.

[37] Schnabel R.B., Estrow E. A new modified Cholesky factorization. SIAM J. Sci. Statist. Comp. 11 (1990) 1136–1158.

[38] Schnabel R.B., Estrow E. A revised modified Cholesky factorization algorithm. SIAM J. Opt. 9 (1999) 1135–1149.

[39] Tier B. Computing inbreeding coefficients quickly. Genet. Sel. Evol. 22 (1990) 419–425.

[40] Wilson A.J., Reale D., Clements M.N., Morrissey M.B., Postma E., Walling C.A., Kruuk L.E.B., Nussey D.H. An ecologist's guide to the animal model. Journal of Animal Ecology 79 (2010) 13–26. doi: 10.1111/j.1365-2656.2009.01639.x.

Index