WOMBAT

A program for Mixed Model Analyses by Restricted Maximum Likelihood

Version 1.0

USER NOTES

Karin Meyer

Animal Genetics and Breeding Unit,
University of New England
Armidale, NSW 2351,
AUSTRALIA
kmeyer@didgeridoo.une.edu.au

MEAT & LIVESTOCK AUSTRALIA

ANIMAL GENETICS AND BREEDING UNIT
A joint unit of NSW DPI and UNE

ILRIC

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

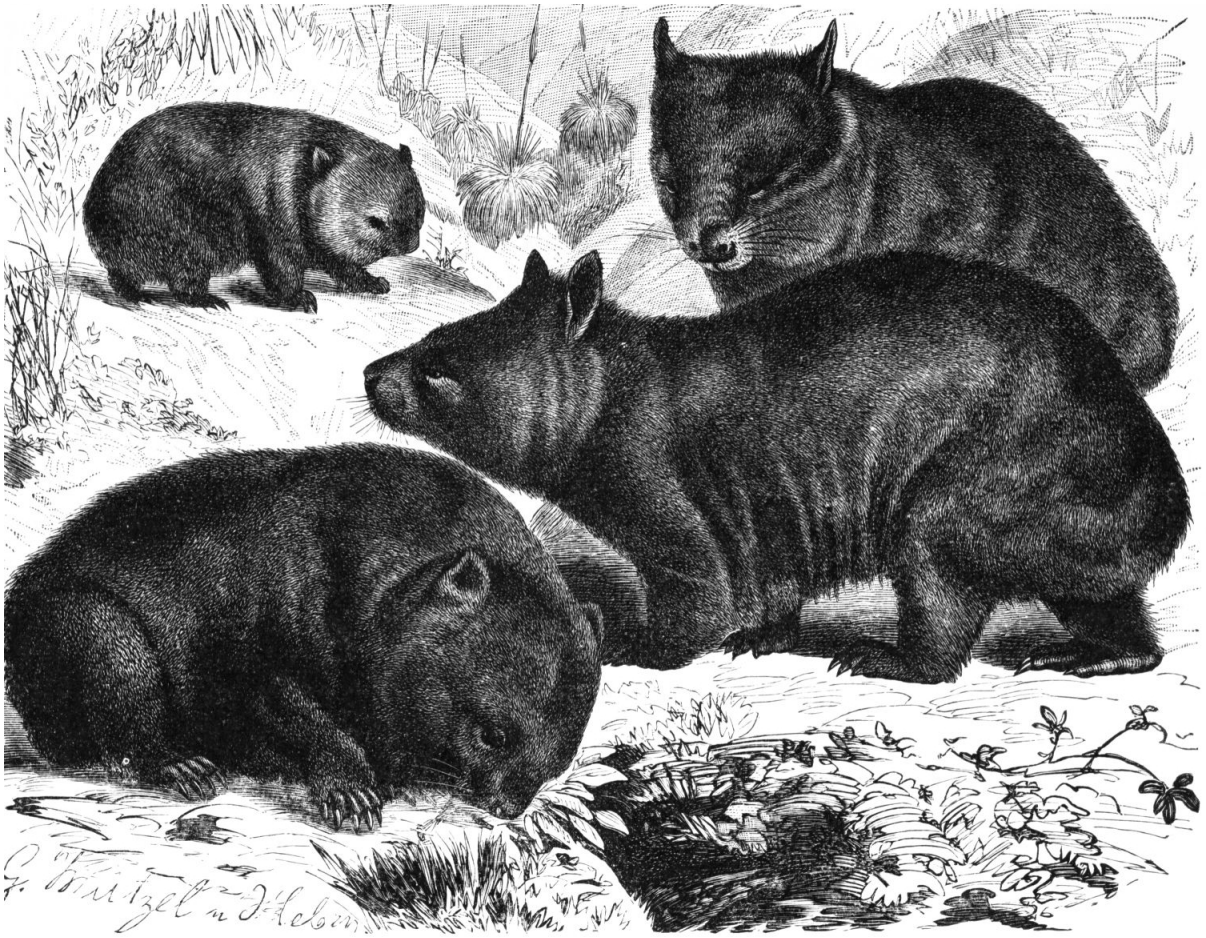
This gives a document which is fully navigable - all references to other sections and citations are 'clickable' within document links, and all links to external URLs can be accessed from within the PDF viewer (if this is configured to do so).

This copy of the manual has been produced on April 11, 2008.

Contents

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, *Phascolomys ursinus G. Cuv.* (links), und Breitstirnwombat, *Ph. latifrons Owen* (rechts), 1/8 natürlicher Größe."

Translation (partly): "Coarse-haired wombat, *Vombatus ursinus G. Cuv.* (left), and Southern hairy-nosed wombat, *Lasiorhinus 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 [14, 15] 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 [10], 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 [9].

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.

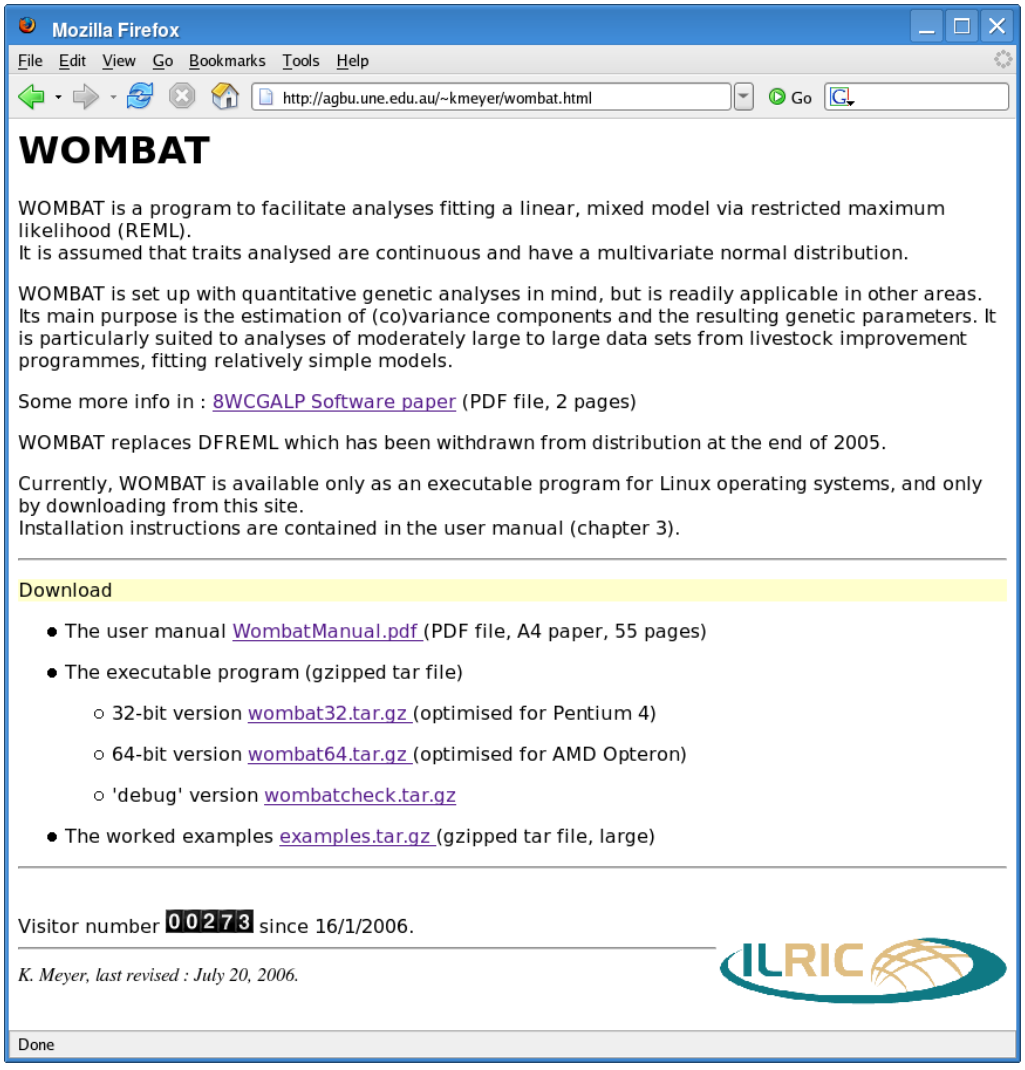
Throughout the manual, it is assumed that users have a thorough knowledge of mixed linear models and a sound understanding of maximum likelihood inference in general. Numerous textbooks covering these topics are available. To get the best from WOMBAT, users should also be familiar with some of the technical 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://agbu.une.edu.au/~kmeyer/wombat.html**

Figure 2.1: WOMBAT Home Page



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, WINDOWS versions are

provided, both to be run under CYGWIN and in a CMD window. These have been restricted to smaller analyses; see compilation notes in Section 3.1.2.

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 forms of citation:

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

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 currently only as a pre-compiled executable file[1]. Installation instructions below are for LINUX operating systems (including CYGWIN under WINDOWS). Versions for 32-bit and 64-bit personal computers are available.

Installation is simple :

1. Download the version appropriate to your machine from **http://agbu.une.edu.au/~kmeyer/wombat.html**
   - `wombat32.tar.gz` for standard 32-bit PCs
   - `wombat64.tar.gz` for 64-bit PCs
   - `wombatwin.tar.gz` for 32-bit PCs emulating a LINUX environment under Windows through CYGWIN
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 system can find WOMBAT, by doing *one* of the following :
   - Add `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.

[1] "Open Source" to be considered later

Done !

3.1.1 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.

1. Download `examples.tar.gz` from **http://agbu.une.edu.au/∼kmeyer/wombat.html**
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 `WOMBAT/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.

3.1.2 Compilation notes

The LINUX versions of WOMBAT have been compiled under Fedora Core 5, using the Pathscale Ekopath™ FORTRAN 95 compiler, version 2.2.1. The 32 bit and 64 bit versions have been optimised for Pentium 4 and AMD Opteron processors, respectively. The additional 32 bit LINUX versions (for debugging or older machines and LINUX versions) have been compiled using the Lahey™ LF95 compiler (version 6.20c) under Fedora

Core 2. Compiled programs have been tested under Fedora Core 2 to 5, as well as Ubuntu 7.04.

The WINDOWS versions of WOMBAT have been compiled using the g95 FORTRAN compiler. Execution times for programs generated with this free compiler (available from **http://www.g95.org**) tend to be slower than those produced by commercial compilers. Programs have been compiled both under CYGWIN and under MINGW. These versions of WOMBAT are primarily intended for evaluation and teaching purposes. Hence, the executables have been restricted to smaller analyses, allowing for models with a maximum of 150 000 equations and at most 50 000 animals in the analysis.

3.1.3 Updates

Expiry

WOMBAT is set up to 'expire' after a certain date. Usually this is the end of the calendar year.[2] 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 6.1 and 6.2 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.[3] The detailed rules given in chapter 4 are best understood following some example(s). The suite of examples provided templates for all types of analyses performed by WOMBAT, and a range of different models.

[2] At present, during active development, WOMBAT may only be 'valid' for a few months
[3] A browser interface to aid with this task is planned

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[4].

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://agbu.une.edu.au/~kmeyer/wombat.html**

   ❏ `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 machines.

2. Use `wombat_chk` for run(s) to demonstrate the error.

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

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.

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

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

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.

File name

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.

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

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
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
VARIANCE	–	Name of random effect
RESIDUAL	–	
ERROR	–	
END		

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

```
PEDS pedigrees.dat
```

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: 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.

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

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

4.1 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.

4.2 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 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.

4.3 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. [Not yet implemented !]

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 reduded rank, multivariate analysis for 8 traits

4.4 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 4.6.1.2). Only one pedigree file can be given. The format of the pedigree file required by WOMBAT is described in detail in Section 6.2.

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.

4.5 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 6.1.

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

4.5.1 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 TRn 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: This shows the block for an analysis reading records for 2 traits from the file `mydata.dat`.

```
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
```

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

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

4.6 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.

4.6.1 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).

4.6.1.1 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 !]

COV : This specifies a fixed covariable. The effect name should have the form "vn(n,BAF)", where vn is a variable name, n gives the degree of fit and BAF stands for a three-letter code describing the basis functions to be used in the regression.
Valid codes for basis functions 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

The INTEGER variable n gives the degree of fit (rather than the order of fit), i.e. the number of regression coefficients. For instance, for polynomials $n = 2$ denotes a quadratic regression. By default, with POL WOMBAT does not fit an intercept for each covariable. Fitting of an intercept can be enforced by preceding n with a minus sign.
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.
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

4.6.1.2 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 :

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.
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 covariance 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

Genetic effect

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.

RRC : This codes specifies a 'control variable' in a random regression analysis.

4.6.1.3 '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.

4.6.2 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]

4.7 Special information

4.7.1 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.

[1] This may sound contradictory to Section 6.1 - 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.

(c) The trait number; this can be omitted for univariate analyses.

4.7.2 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 4.7.1). 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.

4.7.3 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) 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) The running number of the covariance component in the numerator.
   (c) 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) The running number of the covariance component in the numerator.
   (c) The running number of the first covariance component in the denominator.
   (d) The running number of the second covariance component in the denominator.

EXAMPLE: 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 σ^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.

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

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.

4.8 Covariance components

Finally, 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 must be of full rank or, for PC analyses, have rank which is at least equal to number of principal components to be fitted.

4.8.1 Residual covariances

4.8.1.1 'Standard' multivariate analyses

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 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).
2. The $q(q+1)/2$ elements of the *upper* triangle of the residual covariance matrix, given row-wise (i.e. σ^2_1, σ_{12}, ..., σ_{1q}, σ^2_2, σ_{23}, ..., σ^2_q). 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.

4.8.1.2 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.

FUN : This code specifies that error covariances change as a smooth function of the control variable. [Not yet implemented !]

2. 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. Each should begin on a new line and be preceded by two INTEGER values (space separated) indicating the upper and lower limits (inclusive) of the interval of the control variable for which this matrix is applicable.
   - ❏ For FUN the type of function and its coefficients need to be given, as well as the covariance matrix at the lowest value of the control variable.

4.8.2 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

Correlated random effects

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].

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

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

5 Run options

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

```
wombat
```

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 and Table 5.2.

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 4.1). Any such options are read after options specified on the command line, i.e. may overide these.

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.

5.1 Basic run options

5.1.1 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

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
`--simul`	Select a run **simul**ating data only
`--subset`	Write out parameter files for analyses of **subset**s of traits
`--itsum`	**It**erative **sum**mation of partial covariance matrices
`--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
`--invspa`	**Inv**ert a **spa**rse symmetric matrix
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

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.

5.1.2 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

5.1.3 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.

5.1.4 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

`--bad` 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.

5.1.5 Intermediate results

WOMBAT writes out the currently best values of estimates of covariance components to the file `BestPoint` whenever the likelihood is increased.

`--best` 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.

5.1.6 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 5.2.1).

5.1.7 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 6.4.4.3). 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 4.6.1.2).

5.1.8 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
```

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.

`--invspa` 4. 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 5.2.1) 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`.

5.1.9 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 and, subsequently, combine the resulting estimates. 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 run time options to assist with these tasks :

--subset Option `--subset`n is aimed at making 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. These are based on the data (and pedigree) file for the 'full' multivariate analysis, using the facility for automatic renumbering of traits and subset selection, i.e. no edits of these parameter files are necessary.

On analysis, encountering the syntax for trait renumbering (see Section 4.6.2) 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 !

--itsum Option `--itsum` selects a run to combine estimates from partial analyses, using the 'iterative summing of expanded part matrices' approach of Mäntysaari [13] (see also Koivula et al. [11]), 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 7.3.7). 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 4.6.2 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.

These options are not available for analyses involving random regression

models, correlated random effects or permanent environmental effects fitted as part of the residuals.

5.1.10 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.* Often, 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.

5.2 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).

5.2.1 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.

Table 5.2: 'Advanced' run options for WOMBAT

Option	Purpose
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 *(default)*
`--pxai`	Use a few rounds of **PX** followed by **AI** REML
`--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)
`--pownew`	Use Powell [22]'s **new** method for DF maximisation
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
Miscellaneous	
`--noprune`	Do not prune pedigrees

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

`--metis` Option `--metis` selects an ordering using a multilevel nested dissection procedure. Up to three 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 [16] 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.

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 or accessible (some of the 32-bit versions are restricted to 2GB) and thus failing to run while, in reality, much less space is required for the analysis concerned. Hence, the run time option `--choozhz` is supplied: This cases WOMBAT to pause, write out the default settings, and read in (from the terminal) an alternative value specified by the user.

5.2.2 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 ($n = 3$) [23, 24] or partial ($n = 4$) [7] 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: `--pxem6,50,0.001`
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.

Finally, WOMBAT incorporates three choices for a derivative-free algorithm. While little used, these can be usefully to check for convergence

--simplex in 'tough' problems. Option `--simplex` selects the simplex or polytope algorithm due to Nelder and Mead [19], 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
--powell using Powell [21]'s method of conjugate directions, again ported from DfReml and with optional parameters n, C (as for `--aireml`) and S.
--pownew Last, `--pownew` selects Powell [22]'s [Not yet implemented !] new procedure UOBYA (no arguments).

5.2.3 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).

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

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
--logdia 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`
--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.

5.2.4 Other

By default, WOMBAT 'prunes' the pedigree information supplied, i.e. eliminates any uninformative individuals (without records and links to
--noprun 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 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. For example

```
wombat weights
```

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

6 Input files for WOMBAT

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 !

6.1 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 !

INTEGER codes

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.

REAL variables

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.

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.

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

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.

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).

6.2 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 $2\,147\,483\,647$.

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.

6.3 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).

6.4 Other Files

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

6.4.1 General inverse file

For each random effect fitted for which the covariance option GIN (see Section 4.6.1.2) has been specified, WOMBAT expects a file set up by the user which contains the inverse 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, the user should 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 matrix.
   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.

Layout

3. The file should 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 'row' number
   (b) An INTEGER code for the 'column' 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.

6.4.1.1 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).

File name

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

6.4.2 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

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.

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`

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
```

6.4.3 Files with results from part analyses

For a run with option `--itsum` 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.

6.4.4 '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.

6.4.4.1 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.

6.4.4.2 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 !]

6.4.4.3 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 !

6.4.5 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 7.3.7) 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 !

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 4.7.3) 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.

7.2 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 :

```
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.

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.

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.

7.2.5 Files SimData$n$.dat

Simulated records are written to files with the standard name SimData$n$.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$n + \ldots + m$.dat

If an analysis considering a subset of traits is carried out, WOMBAT writes out a file EstimSubset$n + \ldots + m$.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 4.6.2). The first two lines of EstimSubset$n + \ldots + m$.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$n + \ldots + m$.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 7.3.2). 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.

7.3 '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 5.1.3). It can be used to identify the running numbers needed when defining additional functions of covariance components to be evaluated (see Section 4.7.3)

7.3.2 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 !

7.3.3 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).

7.3.4 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 5.2.1).
(c) The number of non-zero elements in the mixed model matrix after factorisation.
(d) The operation count for the factorisation.

7.3.5 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.

7.3.6 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 4.6.1.1) 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.

7.3.7 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.

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 $n = 1, 2$.

`cholnrm`n`.bin` : This file contains the Cholesky factor 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 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 3.1.1. 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; numbers obtained on 32-bit machine may vary slightly.

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 is an animal model with an additional random effect, and included 3 cross-classified fixed effects.

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.

A A 'standard' full rank analysis
B A reduced rank analysis, fitting the first four principal components

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
C Univariate analysis for trait 2, allowing for a non-zero direct-maternal genetic covariance
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
B Reduced rank analysis
C Reduced rank analysis with different ranks

Source: Wokalup selection experiment

Appendix 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 [16]. 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 [12] 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 [10], available at at **www.cs.umn.edu/∼karypis/metis**. This is the default for large analyses.

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	–	–

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 [2].

Default values for the thresholds for these criteria used in WOMBAT 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 [18]. 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 [17]. 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 (**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 **L**. The inverse of the AI matrix then gives approximate sampling covariances $Cov\left(l_{ir}, l_{js}\right)$. The $ij-$th covariance component, σ_{ij}, is

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

with $q(ij) = min(i, j, t)$ 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

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

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

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

Equation A.4 extends readily to two covariance components belonging to different covariance matrices, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, 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

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

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

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

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 [23] 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) [5], 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 [24].
2. A partial Cholesky decomposition has been suggested by Forsgren et al. [7]. 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 [20], 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

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

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

[3] 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).

[4] 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.

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

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

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

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

[13] 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: 50th Annual Meeting. Europ. Ass. Anim. Prod. (1999). Mimeo., 8pp.

[14] 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.

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

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

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

[18] Meyer K., Smith S.P. Restricted maximum likelihood estimation for animal models using derivatives of the likelihood. Genet. Select. Evol. 28 (1996) 23–49.

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

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

[21] 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.

[22] Powell M.J.D. UOBYQA : unconstrained optimization by quadratic approximation. Math. Programm. Ser. B 92 (2002) 555–582.

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

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