RRGIBBS

A Program to estimate variance components for simple random regression models using Gibbs Sampling

USER NOTES

Karin Meyer

Printed September 4, 2002

Contents

1 Introduction

Random regression (RR) models have become a popular choice for the analysis of longitudinal data or 'repeated' records. Typically, analyses require numerous parameters, i.e. (co)variances between RR coefficients and measurement error variances, to be estimated, especially if the model of analysis includes additional random effects such as maternal effects.

Programs for RR model analysis using restricted maximum likelihood (REML) are available [e.g. 3, 5]. However, the high computational demands of REML analyses for RR models severely limit the feasibility of RR analyses for data sets sufficiently large to support estimation of the pertaining (co)variance components, in particular for models fitting many RR coefficients.

Bayesian analyses using Gibbs sampling provide an alternative which is markedly simpler to implement than REML, and have been recommended for the estimation of parameters required for test-day models in dairy cattle [6]. Whilst the range of models which can be accommodated via Gibbs sampling may be more restrictive and the total computing time required may be longer than for corresponding REML analyses, memory requirements are substantially less. Hence Bayesian methodology readily facilitates large scale analyses. Apart from these practical advantages, of course, it provides estimates of complete sampling distributions rather than just simple point estimates.

Purpose

RRGIBBS performs a single task : the analysis of a simple class of RR models using Bayesian methodology. Models may involve

- multiple fixed effects, including cross-classified effects and 'standard' covariables, as well as fixed regression(s) on Legendre polynomials of the meta-meter;
- sets of random regression coefficients, regressing on orthogonal polynomials or user-defined functions of a single, continuous covariable, the so-called"meta-meter";
- multiple random effects, distributed proportionally to an identity matrix of the numerator relationship matrix between animals,
- different orders of polynomial fit for each random effect,
- homogeneous or heterogeneous measurement error variances modelled as a step function of the covariable,
- a single trait only.

Model specification is via a parameter file. The run time behaviour of RRGIBBS can be modified by a number of command line options.

Results

RRGIBBS is a "no-frills" program. It basically offers little more than a reasonably efficient Gibbs sampler for a range of random regression analyses. The main output from RRGIBBS are files with the successive samples of (co)variance components drawn, ready for your favourite post-Gibbs analysis.

In addition, limited summary information is produced, including estimates of covariance matrices among RR coefficients and measurement error variances obtained as means over samples (after "burn-in"), and approximate 95% highest posterior density regions.

2 Methods

2.1 Sampling strategy

RRGIBBS generates a Markov Chain for the parameters of a RR model, sampling from their fully conditional posterior distributions. Full details are given, for instance, by Jamrozik and Schaeffer [4] and Rekaya et al. [7] for RR analyses of test-day records in dairy cattle. These extend readily to models involving additional random effects.

Covariances among RR coefficients are sampled from inverted Wishart distributions; see Jamrozik and Schaeffer [4] and Rekaya et al. [7] for formulae and Sorensen [8, section 9.2] for an outline of how to draw a sample from such distributions.

Measurement errors are assumed to be uncorrelated and the corresponding variances are thus sampled from inverted χ^2 distributions.

Samples for the location parameters, i.e. fixed and random effects, can be drawn either in 'scalar' or 'multivariate' mode.

2.1.1 "Scalar" Gibbs sampler

In scalar mode, effects are sampled one at a time. If a set of regression coefficients is fitted for an effect, these are sampled simultaneously or in a block, with block size equal to the order of polynomial fit, from a multivariate normal distribution.

This is the standard procedure applied in most instances and formulae for fully conditional distributions given by Jamrozik and Schaeffer [4] pertain to this sampling scheme.

2.1.2 "Multivariate" Gibbs sampler

In multivariate mode, all location parameters are sampled simultaneously. This requires sampling of all RR coefficients for the given data and pedigree structure and iterative solution of the mixed model equations for each Gibbs sample. Full details are given by García-Cortés and Sorensen [2] and Sorensen [8, section 9.5]

2.2 Highest Posterior Density region

Assume samples (after "burn-in") are ordered according to value. Approximate 95% highest posterior density regions are then determined simply as the points between the highest of the 2.5% lowest sample values and the next value higher

than it for the lower limit, and, analogously as the point between the lowest of the 2.5% highest samples and the next value lower than it for the upper limit [1].

2.3 Mixed model equations

The symmetric coefficient matrix of the mixed model equations typically is large and sparse. Hence only the non-zero elements of the lower triangle are stored in a series of linked lists.

Iterative solutions for fixed and random effects fitted required for the multivariate Gibbs sampler are obtained using a preconditioned conjugate gradient algorithm [9, 10], with the preconditioning matrix equal to the inverse of diagonal blocks corresponding to sets of RR coefficients.

3 Installation

3.1 Availability

RRGIBBS can be obtained *only* by downloading the material required from its web page : **http://agbu.une.edu.au/∼kmeyer/rrgibbs.html**

Material available comprises source code (complete with 'Makefile' for a UNIX environment), a manual and a worked example, as well as pre-compiled programs for several computing environments.

3.1.1 Conditions of use

RRGIBBS is available to the scientific community free of charge.

RRGIBBS is available under the conditions that it remains my copyright, that it is not modified other than to adapt it to the local computing environment or for personal research. Users are required to credit its use in any publications.

Permission is granted to redistribute RRGIBBS under the condition that this done using the complete and unmodified package.

3.1.2 Disclaimer

While every effort has been made to ensure that RRGIBBS does what it claims to do, there is absolutely no guarantee for its correct performance. You are using RRGIBBS entirely at your own risk.

As it is freeware, there is no user-support service. However, I do invite constructive criticism and genuine, informative bug reports to :
`kmeyer@didgeridoo.une.edu.au`.

3.2 Manual

The manual (this document) for RRGIBBS comprises more than 30 A4 pages. It is available as a PDF file :

- **RRGmanual.pdf**

3.3 Compiled programs

3.3.1 LINUX

RRGIBBS has been compiled under LINUX Redhat 7.2 using the Lahey/Fujitsu F95 compiler (Version 6.00a). This can be downloaded as :

- **RRGibbs.Linux.gz**

As the extension ".gz" indicates, this file has been compressed using `gzip`. To install, uncompress by issuing
`gunzip RRGibbs.Linux.gz`
the rename
`mv RRGibbs.Linux rrgibbs`
and move to the directory where you keep your executable programs.

3.3.2 UNIX

A compiled version of RRGIBBS for a Compaq True64 workstation (UNIX V5.0A; Compaq FORTRAN V5.5-1877) is available as :

- **RRGibbs.Compaq64.gz**

Again, this file has been compressed using `gzip`. To install, uncompress by issuing
`gunzip RRGibbs.Compaq64.gz`
then rename
`mv RRGibbs.Compaq64 rrgibbs`
and move to the directory where you keep your executable programs.

A corresponding file for a 32 bit machine, compiled using the Compaq FORTRAN compiler version 5.3-915 is available as :

- **RRGibbs.Compaq32.gz**

3.3.3 Solaris

3.4 Source code

RRGIBBS is written in standard FORTRAN 95 and is self-contained, except for public domain routines to generate random samples from the normal and χ^2 distribution.

The source code can be downloaded as :

- **RRGibbs.tar.gz**

As the extension ".tar.gz" indicates, this is a UNIX 'tape archive' which has been compressed using `gzip`. To install, uncompress and unpack the archive by issuing

```
gunzip RRGibbs.tar.gz
tar -xvf RRgibbs.tar
```

This will create a directory `GibbsRR` which contains the relevant FORTRAN files and a UNIX type `Makefile`.

3.4.1 Random number routines

RRGIBBS is set up to use the RANDLIB90 package from the Department of Biomathematics of the M.D. Anderson Cancer Centre at the University of Texas to the obtain random samples required by the Gibbs sampler.

If you choose to use this software, download the source code from :

ftp://odin.mdacc.tmc.edu/pub/source/randlib90.tar.gz

unzip and unpack as above, and follow the instructions given for installation.

The `Makefile` provided expects to load the compiled subroutines from RANDLIB90 from a library file named `librand90.a`, and to find the pertinent `.mod` file in a directory `randlib90/SOURCE` which is a sub-directory of `GibbsRR`. If your set-up differs, you need to change `Makefile` accordingly !

> *N.B.* : I had a problem with `random_multivariate_normal_mod.f90` from this package. It appeared that a vector was not allocated leading to a 'Segmentation Fault' at run time. I have notified the people concerned, but not received a reply. I have thus included my modified version of this routine with the source code for RRGIBBS. If you encounter similar problems, try replacing the file provided in RANDLIB90 with this one.

Otherwise, you need to edit the file `random.f` – this provides an interface for the random number generation routines used – and substitute the pertinent routines from your chosen software.

3.4.2 Editing the Makefile

If your system set-up provides a `make` utility, the next step is to edit the `Makefile` provided. Changes required are directory names and compiler options. The customisable section in `Makefile` is clearly marked.

3.4.3 Compilation

A simple 'make' should then create the executable file `rrgibbs` in the directory specified.

If you do not have a 'make' utility or if you are having trouble adapting the Makefile provided, you can always concatenate all the source code files into one big file and compile 'manually'.

Do not contact me for advise if you are having problems compiling RRGIBBS *getting 'make' to work or setting up random number routines – I will only tell you to consult your local computing experts !*

3.5 Testing

Whether you downloaded a precompiled program or compiled your own, you should test your installation by working through the example provided (see chapter 8), making sure that you get comparable results.

> *N.B.* If you are compiling RRGIBBS yourself, it is good practice to do so first up without optimisation and checks and profiling switched on, and to recompile with full optimisation only after a successful test run.

4 Input Files

RRGIBBS uses four types of input files. The first three must be supplied by the user. All input files are expected to be "formatted" in a FORTRAN sense, i.e. plain ASCII or text files.

4.1 Data File

This file contains the data to be analysed, and all information on effects in the model of analysis.

Name

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

Format

- The data file must be 'formatted' (in a FORTRAN sense), i.e. a plain text of ASCII file. Non-ASCII characters may cause problems.
  All variables must be stored in fixed width columns, separated by spaces.
- All fixed and random effects codes must be integer variables, i.e. contain digits only. The maximum value allowed for a code is 2,147,483,647.
- All other variables are read as real values, i.e. may contain digits, plus or minus signs, and FORTRAN type formatting directives only.
- No special codes for 'missing values' are available.
- Any alphanumeric strings in the part of the data file to be read by RRGIBBS are likely to produce errors.

The data file must be sorted according the 'subject' for which repeated measurements are taken (ascending order), and according to the value of the meta-meter within subject (e.g. animal and age at recording within animal).

4.2 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.

Name

- There is no 'default' name for the pedigree file. File names up to 30 characters long are accommodated.
- The pedigree file must contain one line for each animal in the data.
- Additional lines for parents without records can be included.

Layout

- Each line is expected to contain three integer variables : the animal code, the code for the animal's sire, and the code for the animal's dam.

Coding

- All codes must be integer values.
- All animals must have a numerically higher code than either of their parents.
  Unknown parents are to be coded as "0".
- If maternal genetic effects are to be fitted in the model of analysis, all dams of animals in the data must be known.

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.

4.3 Parameter File

RRGIBBS acquires all information on the model of analysis from a parameter file. This must be set up following the (complicated) set of rules described in detail in chapter 5.

4.4 Other Files

RRGIBBS will check for existence of several, one-line files with standard names in the working directory and, if they exist, acquire information from them rather than using a program defined default.

4.4.1 Initial random number

RRGIBBS uses two integer values to initialise the random number generator. These are expected to be read from a file called `randno`. If this file does not exist in the current directory, initial numbers are instead derived from the date and time of day.

4.4.2 Size of numerator relationship matrix

RRGIBBS sets the maximum number of non-zero off-diagonals in the numerator relationship matrix (lower triangle) to a pre-defined multiple of the number of animals in the analysis. For most applications, this should be sufficient. If a larger number is required, however, this can be obtained by supplying a file `nrmzhz` which contains this (integer) number.

4.4.3 Size of coefficient matrix

RRGIBBS sets the maximum number of non-zero off-diagonals in the coefficient matrix of the mixed model equations (lower triangle only) to a pre-defined multiple of the number of equations. In some cases, this can be too small. In other

cases it can be a very large number and RRGIBBS may not be able to allocate a correspondingly large matrix. In either case, the default value can be over-ridden by supplying a file `maxzhz` which contains the desired value for this (integer) number.

> *Hint* : Running RRGIBBS with the "–A" option (set-up steps only, see section 6.1), will generate `maxzhz` (and overwrite any existing file) and write the exact number required for the current analysis – this helps to avoid allocation of excess, unused space during the actual run and thus keep the memory requirements of RRGIBBS as small as possible.

4.4.4 User defined functions

If regression on user-defined functions has been chosen for one or more of the random effects, RRGIBBS expects to read these from additional input file(s). The required form for these files is :

- Files are formatted files (in a FORTRAN sense).
- There should be one row for each value of the meta-meter.
- Row should correspond to values of the meta-meter in ascending order.
- The number of columns must be equal to the order of fit specified for the random effect.
- The elements of each row should be the user-defined functions evaluated for the pertaining value of the meta-meter.

  Simple example :
  Assume the meta-meter has possible values of 1, 3, 5, 7 and 9, and that we want to fit a quadratic regression on 'ordinary' polynomials. In this case, RRGIBBS would expect to read a file with 5 rows and 3 columns :

```
1   1   1
1   3   9
1   5  25
1   7  49
1   9  81
```

4.4.5 Output from a previous run

If a file `nrminv.bin` exist in the working directory, RRGIBBS will attempt to read the inverse of the numerator relationship matrix from it, instead of constructing it anew.

If a continuation run is specified, RRGIBBS expects to read a complete Gibbs sample form which to continue sampling the Markov chain from a file `chain1.bin`

or `chain2.bin`. These are written at regular intervals and at the end of each 'sampling run' of RRGIBBS (see section 7.4.2).

5 Specifying the Model

RRGIBBS acquires all information on the model of analysis and layout of the data file through a *parameter file*. The name of this file must have extension '.par'. This section describes the format required.

The syntax of the parameter file follows loosely that used by ASREML [3]. Parsing of the options given, however, is fairly elementary. Hence it is important that the following rules are strictly adhered to – RRGIBBS will fail without explanatory messages otherwise.

RRGIBBS does not carry out consistency checks on the variables and models specified – and, again, will fail without explanation if these are conflicting.

> *Hint* : Using the –V run time option, RRGIBBS will echo each line in the parameter file read – this will allow you to determine at which line it fails.

5.1 General rules

- Length : only columns 1 to 72 are considered, anything beyond column 72 is ignored.
- Start : except for lines giving the list of variables (see below), all input should start in column 1.
- Case : variable names are case sensitive; use the same spelling for an effect in the list of variables and the specification of the model.
- Spaces : individual items of information on the same line must be separated by spaces.
- Names : all file names can have up to 30 characters.
- Continuation line : A " \ " (backslash) signifies that the next line is to be treated as a continuation of the current line. It must be separated from the last item by space(s).

5.2 Comments

The first line of the parameter file is treated as a comment only.

5.3 Variables

The second and following lines describe the variables, defined as space separated columns, in the data file. These lines must be intended by at least one character – on encountering a line with a non-blank first column, the program assumes that the list of variables is complete.

There should be a line for each variable. Up to 3 items are read :

1. The name of the variable – this can have up to 20 characters. This must be given.
2. The number of levels for this variable, if it is to be fitted as a fixed or random effect in the analysis.
   - This can be a number greater than the actual number of levels.
   - For variables with the "!P" option (see below), this can be given as 0, as the program determines the number of levels for these effects from the pedigree file.
3. An option "!P" to denote that this variable is a random effect with variance matrix proportional to the numerator relationship matrix.

5.4 Pedigree file

If a "!P" option has been given, RRGIBBS expects to read the name of the pedigree file from the line immediately following the last line specifying a variable. This must begin in column 1.

5.5 Data file

The next line must contain the name of the data file.

5.6 Linear Model

The next line(s) must specify the model to be fitted. This should be given in the form :

Trait ∼ Fixed covariables Fixed effect(s) !R Random effects

If necessary, this can span several lines (using \ to indicate a continuation line).

5.6.1 Trait

This is the name of the dependent variable. It must be separated by the form the effects in the model by " ∼ " (note the spaces surrounding ∼).

5.6.2 Fixed covariables

RRGIBBS allows for 'regular' covariables and covariables which are functions of the meta-meter, as for the random effects (see below).

'Regular' covariables are fitted as regressions of the trait on polynomials of the variable selected. These are specified in the form

LIN(covar)

for a *linear* regression on variable 'covar', or

POL(covar,*n*)

for a higher order polynomial regression on 'covar', fitted to order *n*. POL(covar,1) and LIN(covar) are equivalent.

It is common practice in random regression analyses, to fit a (fixed) regression on similar functions of the meta-meter as are used in the random part of the model, to model changes in means. Currently, RRGIBBS only allows for Legendre polynomials of the meta-meter. A corresponding fixed regression is specified as

LEG(meta,*k*)

where 'meta' is the variable representing the meta-meter and *k* is the order of fit. *k* includes an intercept, i.e. the regression involves orthogonal polynomials of meta to the power $k-1$. Hence, $k=2$ yields a linear regression, $k=3$ a quadratic regression, etc.

Covariables can be fitted nested within a fixed effect. This is denoted as

fixeff.LEG(meta,*k*) or fixeff.POL(covar,*n*)

where 'fixeff' is the variable representing the fixed effect code.

5.6.3 Fixed effects

Fixed effects to be fitted are simply specified as

fixeff

where, as above, 'fixeff' is the variable representing the fixed effect code. No facility to code interactions between fixed effects within RRGIBBS are currently implemented.

5.6.4 Random effects

Specification of random effects to be fitted must be preceded by the " !R " qualifier (again, note the spaces).

It is assumed, that sets of random regression coefficients on functions of the meta-meter(s) are fitted for all random effects. The order of fit is specified later, together with degrees of freedom etc. (see below). The default is a regression on Legendre

polynomials of the meta-meter. Random effects using the default setting can simply be specified as

rndeff

where 'rndeff' is the variable representing the random effect code. Alternatively, this can be specified explicitly as

rndeff.LEG

In addition, RRGIBBS allows regression on user-defined functions of the meta-meter. This is specified as

rndeff.USR

If this is chosen, RRGIBBS requires an input file with the function evaluated for all values of the meta-meter occurring in the data.

5.7 Subject and "meta-meter"

The next line must provide information on the 'subject', i.e. the random effect levels for which there are repeated measurements (commonly animal or individual) and the covariable with which variances and covariances are assumed to change, the so-called "meta-meter". These are given as

subject meta meta(*ll*1-*ll*2)

where 'subject' is the name of the random effect code and 'meta' the name of the covariable.

RRGIBBS provides the facility to restrict the range of the covariable for which any additional random effects (other than subject) affect the trait to be analysed. Only if this is required, does the third variable need to be given, with *ll*1 and *ll*2 denoting the lower and upper limits (inclusive) for the covariable to be effective. Needless to say, these values must be within the range of values for 'meta' which is covered in the data (no checks performed).

5.8 Covariances for random regression coefficients

Next, the parameter file has to supply information on order of fit, starting values of covariances and hyperparameters for all random effects in turn.

The following lines are required for each random effect :

1. A line given the name of the random effect, the number of random regression coefficients for each level (= order of fit), and the degrees of freedom for the prior distribution of the covariance matrix between random regression coefficients.

   rndeff *kk* *df*

If a fourth integer variable is found after df, it is interpreted as the running number of the meta-meter to be used ($i = 1, 2, \ldots$), i.e. as a rule, this variable only needs to be given if regression on a restricted range of the meta-meter is required.

However, if a user-defined function of the meta-meter has been chosen for the random effect, this fourth integer must be given, followed by the name of the file containing the user function evaluated for the values of the meta-meter, i.e.

rndeff kk df mm ff

with mm an integer and ff a file name.

2. From the next line onwards, RRGIBBS expects to read the starting values for the matrix of covariances among random regression coefficients. The $kk \times (kk+1)/2$ elements of the *upper* triangle of this symmetric matrix must be given in sequence, i.e. element $(1,1)$, $(1,2)$, ..., $(1,kk)$, $(2,2)$, $(2,3)$, ..., (kk,kk). Continuation lines (\ ) can be used as required.

3. If df has been set to $-(kk+1)$, a flat prior distribution for the covariance matrix is assumed, and the corresponding scale matrix is set to zero. Otherwise, RRGIBBS expects to read $kk \times (kk+1)/2$ elements of the *upper* triangle of the scale matrix next, starting with element $(1,1)$ on a new line.

5.9 Measurement error variances

Finally, the parameter file needs to specify how many measurement error variances are to be fitted, and, for a heterogeneous model, how different classes are defined.

1. The first line must start with 'Residual' or 'Error', followed by the number of error variances to be fitted, the associated degrees of freedom and, for multiple error variances a qualifier which selects how the step function is defined.

   Residual nn df QUA

   Valid values for QUA are FILE, SPEC or EQUAL. The latter is the default value, i.e. if QUA is omitted and $nn > 1$, it is set to EQUAL.

2. The following line(s) required depend on the value of QUA.

EQUAL : This assumes the range of values of 'meta' found in the data is to be subdivided into 'equal' subclasses. This is determined by dividing the range of values by the number of error variances to be fitted. If there is a remainder, the last class is increased in size to accommodate these values.
The nn starting values for measurement error variances are then read

successively, multiple values per line, with continuation lines as required.
If df is not equal to -2, the corresponding scale parameters for the prior distributions are read next. The first value must be on a new line, subsequent values can be on the same line or on continuation lines.

SPEC : This qualifier indicates that the ranges of values for each error variance class are specified in the following.
RRGIBBS then expects to read *nn* lines from the parameter file, one for each measurement error variance, with up to four variables :

(a) The first variable must be the starting value for the measurement error variance.

(b) The second value is the scale parameter of the prior distribution. This is only expected if df has not been specified as -2.

(c) The next two values must be integers, and give the lower limit and upper limit (inclusive values) of the range of meta values for which this error variance is assumed to apply.

FILE : This qualifier tells RRGIBBS to read the assignment of meta values to measurement error variance classes from a separate file. The file name must follow on the same line, separated from FILE by space(s). Starting values and, if applicable, scale parameters of the prior distributions are then read from the parameter file as described above for EQUAL.
The file specified must have one line for each value of meta which occurs in the data, in ascending order of meta values. Each line must have an integer number giving the running number $(1,\ldots,nn)$ of the measurement error variance assigned to the corresponding meta value.

Any further lines in the parameter file are ignored.

Example

The following parameter file gives an example for a random regression analysis of weight records in beef cattle from birth to 800 days. Random regressions for genetic and permanent environmental effects of both the animals and dams are fitted, assuming heterogeneous measurement error variance and restricting the influence of maternal effects to 600 days. Line numbers are shown for ease of reference only and are not part of the parameter file.

Line 1 is a comment only. Lines 2 – 11 describe the variables in the data file. Note the indentation of these lines, and that upper limits for the numbers of levels are not required for 'animal' and 'gendam', which have the '!P' qualifier, and 'age' and 'wt' which represent the meta-meter and the trait to be analysed, respectively.

```
Parameter file for RR Gibbs analysis
   animal 0 !P
   gendam 0 !P
   btype 2
   sex 3
   damage 6
   icg 4000
   pedam 5000
   age
   wt
   animal2 10000
fort.33
fort.22
wt ~ sex.leg(age,4) damage.leg(age,4) btype icg !R \
animal gendam animal2 pedam
animal age age(0-600)
animal 4 -5
324.51 140.949 -12.35 5.75 \
76.399 -1.120 -1.386 \
5.199 8.8175E-002 2.506
gendam 3 -4 2
116.046 20.248 -16.22 9.59 -1.2485 4.68
animal2 4 -5
653.78 267.33 -27.896 32.867 \
186.53 21.832 -6.22 \
58.6975 23.49 \
35.867
pedam 3 -4 2
89.792 18.558 -5.84 10.299 3.462 7.954
Residual 13 -2 SPEC
2.733 0 0
35.16 1 60
31.41 61 120
20.12 121 180
39.73 181 240
67.47 241 300
151.5 301 360
124.3 361 420
103.1 421 480
131.7 481 540
144.1 541 600
132.6 601 660
157.0 661 800
```

Line 12 gives the name of the pedigree file, starting in column 1, and line 13 gives the name of the data file.

The model of analysis is specified in lines 14 and 15. The trait to be analysed is 'wt'. Fixed effects fitted are regressions on cubic Legendre polynomials of 'age' ($kk = 4$) nested within 'sex' and nested within 'damage' classes. In addition, 'btype' and 'icg' are fitted as cross-classified fixed effects. Random regressions are fitted for 'animal', 'gendam', 'animal2' and 'pedam'. Here, 'animal' and 'gendam' are the genetic effects while 'animal2' and 'pedam' represent the permanent environmental effects of animals and dams represented in the data.

Line 16 specifies the subject and meta-meter(s) explicitly. The subject is 'animal', i.e. there are repeated records for levels of 'animal' and the data file is expected to be ordered according to 'animal'. The meta-meter is 'age'. For some random effects, only a subset of age is to be considered, namely 0 to 600 days.

Lines 17 to 20 give information relating to the covariance among regression coefficients for the first random effect, 'animal'. This is to be fitted to order $kk = 4$, i.e. a cubic regression. The associated degrees of freedom are given as -5, i.e. a flat prior distribution is assumed. No fourth variable is given on line 17, i.e. regressions for 'animal' are fitted for the full range of ages in the data. Lines 18 to 20 give the $4 \times (4+1)/2 = 10$ elements of the upper triangle of the symmetric matrix of starting values for covariances among random regression coefficients. A corresponding scale matrix is not given since $df = -(kk+1)$ have been specified.

Lines 21 to 22 contain corresponding details for 'gendam'. This is fitted to order $kk = 3$, with 6 covariance components specified. Again, flat priors are assumed. Line 21 has a "2" as fourth variable, indicating that regressions are to be fitted for the "second" meta-meter 'age(0-600)', i.e. that records taken after 600 days are assumed to be unaffected by 'gendam'.

Lines 23 to 27 and lines 28 to 29 give similar details for the third and fourth random effect, 'animal2' and 'pedam', respectively.

Lines 30 to 43 give details on the measurement error variances to be fitted. As shown in line 30, there are 13 classes. There are -2 degrees of freedom for each component, i.e. flat priors are assumed here as well. The qualifier SPEC is given, indicating that the next 13 lines give the starting values and age ranges for each variance. Ranges specified show that a separate error variance is to be fitted for records taken at birth (age 0), while error variances are subsequently assumed to change every 2 months (60 days), except for the last class which spans 140 days.

6 Run Time Options

The behaviour of RRGIBBS is determined by a number of options. These determine, for instance, the number of Gibbs samples to be drawn, the "burn-in" period and the type of output to be generated.

All options have default values, but can be modified through *command line arguments*. The syntax used is UNIX (LINUX)-like, and the program fussily requires things to be specified *just right* :

- Each argument must begin with a "–" sign, immediately followed by a letter and, if applicable, the value of the option chosen.
- Blanks between the – sign, the letter and the option are not allowed.
- Multiple options after a – sign are not recognised.
- Options should be separated by spaces.

In addition, RRGIBBS requires the name of a parameter file, from which all information about the layout of the data file and model of analysis is acquired. This file must have extension ".par". In specifying the file name, the extension can be omitted. The name of the parameter file must be the *last* command line argument. If not given, RRGIBBS will look for a parameter file with standard name "`gibbs.par`".

6.1 Options for Gibbs Sampling

–A : If given, only the "set-up" steps in RRGIBBS are carried out.

–B : followed by an integer number *nn*, which specifies the length of the "**B**urn-in" period, i.e. the number of samples ignored when calculating means over samples at the end of the analysis.
Default : *nn* =10,000, or *nn* =0 if –C is given.

–C : If given, this requests **C**ontinuation of a previous analysis.
This requires a file containing the complete chain from a previous sample to exist. It sets the "burn-in" period to 0.

–G : followed by an integer number *nn*, which specifies the total number of **G**ibbs samples to be drawn.
Default : *nn* =50,000.

–L : followed by an integer number *nn*. This option selects that sample values for the first *nn* **L**ocation parameters are to be written out in addition to

values for variance components.
Default : *nn* =0.

–M : requests that **M**eans for location parameters across samples are calculated, and written out together with the means for variance components after all samples have been obtained.

–O : followed by three letters gives an **O**ption determining the sampling scheme to be used : BLK selects a "scalar" Gibbs sampler, sampling in blocks of size equal to the order of polynomial fit; MUL selects a multivariate Gibbs sampler, sampling all location parameters jointly.
Default : BLK

–S : followed by an integer number *nn*, which specifies the interval at which the current chain is **S**aved to a file.
Default : $nn = 100$.

–V : requests **V**erbose mode. This entails expanded output to the screen during the set-up phase (useful to check the parameter file), and printing of variance components for every Gibbs sample.

6.2 Options for summary analyses

RRGIBBS provides a limited range of post-Gibbs analysis options.

–D : followed by an integer *mm* between 0 and 100 requests calculation of approximate (1-$mm/100$) Highest Posterior **D**ensity regions, with *mm* the error probability.
Default : $mm = 5$.

–F : followed by a file name, ff, signifies that the Gibbs samples in the chain to be summarised are stored in several **F**iles (obtained by successive runs of RRGIBBS using the –C option), rather than in the standard file "samples.rr". The file specified (ff) must contain the names of the sample files in the order in which they were obtained.

–H : followed by an optional, integer number *mm*. Used in conjunction with –Z, this option requests the output of frequency distributions (**H**istograms) for the samples of variance components (after "burn-in"). *mm* determines the approximate number of classes used.
Default : $mm = 40$.

–I : requests that the samples for all **I**nvidual parameters are written out to separate files.

–Z : specifies that RRGIBBS is to be run in "summary mode".
Additional options –F, –H and –I select which calculations are performed, and are only effective in conjunction with –Z.

6.3 Other options

–P : requests that RRGIBBS obtains solutions for all fixed and random effects fitted for the (co)variance components given.
If "–P" is followed by three letters, these select the iterative solution scheme to be used : GAU chooses Gauss-Seidel iterations with successive overrelaxation; PCG chooses a Preconditioned Conjugate Gradient algorithm.
Default : PCG

–Q : runs RRGIBBS as a simulation program. If "–Q" is followed by additional characters, these are assumed to represent the name of the output file to which the simulated data is written.
Assuming the RR model given is the true model and the covariances between RR coefficients specified are the population values, records for the given model, data and pedigree structure are simulated and written out to a file. Samples are drawn from multivariate normal distributions with means of zero. No systematic differences between animals (fixed effects) are simulated. The output file has the same layout as the input data file – replacing the original observations by the simulated values – but all variables are written out as INTEGER values (15 characters wide) unless they have been specified to be covariables (including the meta-meter) variables or traits. The latter are written as REAL values (20 characters wide).
Default file : `NewRRdata.dat`

Examples

```
rrgibbs -V
```

specifies an analysis with 50,000 Gibbs samples, a "burn-in" period of 10,000, saving of all parameters (location and variance components) every 100 samples, parameter file "gibbs.par", and verbose output.

```
rrgibbs -G1000 -B200 -S50 -M -L10 rrm.par
```

runs an analysis comprising 1000 Gibbs samples with a "burn-in" of 200 samples, saving every $50-th$ sample to a file for use in a potential continuation run. Values for the first 10 location parameters are written out for each sample, and mean sample values for all fixed and random effects fitted are to be calculated after sampling has been completed. The parameter file for the analysis is "rrm.par".

```
rrgibbs -Z -Fsfile -B30000 -H50 -D rrm.par
```

summarises results from several runs of RRGIBBS for "rrm.par". Names of files with the sample values for variance components are listed in "sfile". Calculation of frequency distributions of samples with approximately 50 classes and Highest Posterior Density regions (for a default error probablity of 5%) omitting the first 30,000 samples are requested.

7 Output Files

7.1 "Sampling" runs

7.1.1 Sample values for (co)variance components

The primary output from RRGIBBS is a file with the values for the covariances among RR coefficients and measurement error variances for each Gibbs sample. This file is a binary file – in a FORTRAN sense – with one record per sample, each record consisting of the running number of the sample (INTEGER) and the values for the (co)variances (REAL*8).

The output file has a standard name of `samples.rr`. However, if this file already exists in the current directory, RRGIBBS will not overwrite it, but attempt to open a file `samples.rr01` instead. If this exists as well, file names of `samples.rr02`, `samples.rr03`, ..., `samples.rr19` are tried in sequence.

RRGIBBS is set up to read sample values from a number of files in a 'summary' run (see run time option "–F" in section 6.2).

7.1.2 Sample values for location parameters

Optionally RRGIBBS will write out the sample values for selected 'location parameters' (run time option "–L"; see section 6.1). These are written in the same form as described above (section 7.1.1) for the variance components, but to file(s) with standard names of `locpars.rr`, `locpars.rr01`, ..., `locpars.rr19`.

7.1.3 Report file

Summary information on the model of analysis and data and pedigree structure encountered is written to a formatted file `RRGibbs.out`. If RRGIBBS completes the number of samples specified, means and ranges over samples after the "burn-in" phase are also given.

7.2 "Summary" runs

Summary runs – specified using the "–Z" run time options – can produce a multitude of output files.

7.2.1 Results file

Results from a summary run are written to a file called `RRSummary.out`.

Results given vary with the options specified. They include means, approximate Highest Posterior Density regions and ranges for the variance components estimated, and covariance and correlation matrices for each random effect together with their eigenvalues.

7.2.2 Estimates file

Estimates of variance components are written to a file called `Estimates.out`.

If this file is found in the working directory for a BLUP or simulation run, this file is read and the estimates given replace those specified in the parameter file. This allows estimates obtained to be used for such runs without the need for (tedious) editing of the parameter file.

In addition, this file is the direct equivalent to `DF17#DAT` generated by DXMRR [5], complete with dummy log likelihood. Hence it can be used directly as input file for any programs set up to read estimates from such file, e.g. to evaluate covariances and correlations between records at selected ages or eigenfunctions of the resulting covariance functions.

7.2.3 Variances file

RRGIBBS calculates the variances for each random effect for the ages represented in the data, together with the total variance at each age and the pertaining variance ratios. These are written to a file called `Variances.out`.

7.2.4 Samples file

If samples have been obtained in several, continued runs of RRGIBBS these are consolidated into a single file `samples.all`, eliminating any duplicate samples. Like `samples.rr`*nn*, this is an unformatted file with one record per sample, consisting of the sample number and the values for the vector of variance components.

7.2.5 Individual parameters

RRGIBBS provides the option of creating formatted files with the complete chain sampled for individual variance components. This is requested using run time option "–I" (see section 6.2). If given, RRGIBBS will write formatted files `par001.dat`, `par002.dat`, ..., `par`*nnn*`.dat` with *nnn* the total number of variances to be estimated.

7.2.6 GNUPLOT command files

Inspection of chains for individual parameters or the distribution of sampling values can provide valuable clues to the progress of the analysis and the ability of the data to support estimation for the model specified. As described above (section 7.2.5), a summary run of RRGIBBS can be used to produce files with chains for individual parameters (together with their running means) of frequency distributions. Plotting the resulting curves or histograms can be tedious when we have numerous parameters.

GNUPLOT is a versatile freeware plotting program. It is available for UNIX and LINUX platforms and often part of the default installation. There is also a Windows version. The GNUPLOT home page can be found at http://www.gnuplot.info. GNUPLOT is ideal to produce quick, 'working' plots. Hence RRGIBBS has been set up to write out command files to plot the results for individual parameters :

- `Parameters.plt` is written when the "–I" (see section 6.2) run time option is given. It will create a plot for each variance component, plotting the sample value and corresponding running mean every $50-$th sample.
- `Histograms.plt` is written when "–H" (see 6.2) is selected. It will create a plot for each variance component, plotting the frequency distribution of all samples after the "burn-in" phase.

7.3 "Other" runs

7.3.1 Solutions

When RRGIBBS is run as a BLUP program only (run time option "–P", see section 6.3), the solutions for fixed and random effects for the given values of variance components are written to a file called `Solutions.out`. This is a formatted file with one row per effect listing the equation number in the mixed model equations and the corresponding solution. In addition, the solutions are written by effect together with the original codes found in the data or pedigree file to `RRGibbs.out`.

7.3.2 Simulated data

If a simulation run is requested, the simulated data are written to a formatted file. If a name for the output file has been specified, this will be used. Otherwise the output file is called `NewRRdata.dat`.

7.4 Files to be used in subsequent runs

RRGIBBS writes several files with information which may be used in a subsequent run, with the aim of reducing overall time and memory requirements.

7.4.1 Inverse of the numerator relationship matrix

RRGIBBS writes the non-zero elements of the inverse of the numerator relationship matrix (NRM) between animals to a binary file with standard name ‘” `nrminv.bin` ”. This is read for each Gibbs sample.

At the beginning of each run, RRGIBBS will look for a file `nrminv.bin` in the current working directory. If it exits, RRGIBBS will read the number of animals and the identities of the first and last animal and their parents in the run which created this file. If these agree with corresponding values from the current run (derived reading the pedigree file specified), RRGIBBS will use this file rather than setting up the NRM afresh.

7.4.2 Intermediate Markov chains

Even the most reliable computer will crash at times. Hence we cannot always expect a Gibbs sampling run to complete. With potentially long “burn-in” and Markov chains required for RR analyses, it is essentially that RRGIBBS can recover from an unexpected interruption with little loss of work being carried out so far. This is achieved by written out the complete Gibbs sample at regular intervals. This interval is set by the “–S” run time option (see section 6.1) and should be the smaller the longer a sample takes to complete. Two files, `chain1.bin` and `chain2.bin` are used in turn to store this information.

If RRGIBBS is to be restarted after a crash, a continuation run (option “–C”) should be specified. This will cause RRGIBBS to look for these files and read the last chain saved from the more recent of them. In a summary run with multiple files of samples to be processed, RRGIBBS will account for such interruptions and potentially duplicate samples.

7.4.3 Other information

As outlined in section 4.4, RRGIBBS will attempt to read information on the array size required and initial random numbers to be used from files `randno`, `maxzhz` and `nrmzhz`. In turn, these are updated in each run.

> *Hint* : It is good practice to carry out each analysis in a separate working directory and to delete any files created by RRGIBBS before starting a new analysis.

8 Worked example

The worked example provided is the same as for DXMRR.

Test data given in file `mrrtst.dat` consist of 1626 January weights of 436 beef cows (from the Wokalup selection experiment in Western Australia), recorded between 19 and 82 months of age. There are 24 different ages (in months) in the data and up to 6 records per cow. The total number of animals in the analysis (including parents without records) is 784. The analysis is carried out within 83 contemporary groups (year-paddock-age of weighing subclasses), fitted as fixed effects.

Corresponding pedigree information is supplied in file `mrrped.dat`.

A polynomial order of fit of 3, i.e. a quadratic function, is chosen for the random regression analysis, and a corresponding fixed regression on Legendre polynomials of age is fitted. Six different measurement error variances, corresponding to the 6 years of age, are fitted. This yields a total of 18 parameters to be estimated.

The parameter file, `gibbs.par`, for this analysis using the same starting values for variance components as in DXMRR and flat priors, is as follows :

```
Parameter file for RR Gibbs analysis
   no1 1
   animal 0 !p
   sire 0
   gendam 0
   icg 90 !i
   animal2 500 !i
   wt 0
   age 0
mrrped.dat
mrrtst.dat
wt ~ no1.leg(age,3) icg !r animal animal2
animal2 age
animal 3 -4
4500 800 10 300 10 100
animal2 3 -4
1500 10 10 100 10 50
residual 6 -2 SPEC
800 19 30
1000 31 42
```

```
1400 43 54
1300 55 66
1200 67 78
1000 79 999
```

Assume we have a file `randno` in the current working directory which sets the two initial random numbers to 111111 and 777777.

A run in 'sampling mode' drawing 1000 samples and treating the first 250 samples as burn in can be performed using :

```
rrgibbs -G1000 -B250 gibbs.par
```

The screen output from this run has been capture in file `SampleRun.log`. The corresponding results file has been copied to `SampleRun.out`. Note that no files `nrminv.bin`, `maxzhz` or `nrmzhz` existed in the directory. Hence RRGIBBS assigned arrays much too large to store the coefficient matrix in the mixed model equations.

A simple run in 'summary mode' can be carried out as :

```
rrgibbs -Z -B250 gibbs.par
```

This writes the summary of results to file `RRSummary.out` and also creates the files `Estimates.out` and `Variances.out`.

The first five six of `Variances.out` for this example are :

```
19 996.33 265.71 224.08 1486.1 0.670 0.179
20 997.39 265.66 224.08 1487.1 0.671 0.179
21 1003.5 270.44 224.08 1498.0 0.670 0.181
22 1014.3 279.63 224.08 1518.0 0.668 0.184
31 1262.7 500.34 1695.2 3458.1 0.365 0.145
32 1301.1 534.12 1695.2 3530.4 0.369 0.151
```

The first column gives the age at recording, columns 2–5 give the additive genetic, permanent environmental, measurement error and phenotypic variance, respectively, and columns 6 and 7 give the heritability and permanent environmental varaince as a proportion of the total. Clearly, with only 1000 samples, the results are not very plausible.

A continuation run to create another 9000 samples could be started as :
`rrgibbs -C -G9000 rrgibbs.par`
This would write the new samples to a files `samples.rr01`. To carry out a summary run utilising all samples, you need to first create a file which lists the names of all sample files to be considered, `samples.rr` and `samples.rr01` in thsi case. Call this `samfiles`. A summary run disgarding the first 2000 samples as "burn-

in" could then be done using :

```
rrgibbs -Z -B2000 -Fsamfiles gibbs.par
```

or

```
rrgibbs -Z -I -H -B2000 -Fsamfiles gibbs.par
```

if output files with individual chains and frequency distributions of samples were required (Files for the latter two steps not included in example).

Files for the example can be downloaded from
http://agbu.une.edu.au/∼kmeyer/rrgibbs.html

as

- **Example.tar.gz**

As the extension ".tar.gz" indicates, this is a UNIX 'tape archive' wich has been compressed using `gzip`. To install, uncompress by issueing
`gunzip Example.tar.gz`
and unpack the archive using
`tar -xvf Example.tar`
This will create a directory `GibbsRR/Example` which contains the files described above.

> *N.B.* : The example run provided was carried out on a Compaq Alpha True64 machine. Runs on other architectures (32bit) may yield different random numbers for identical 'seeds' of the random number generator, and thus somewhat different results !

Bibliography

[1] Chen M.H., Shao Q.M., Monte Carlo estimation of Bayesian credible and HPD intervals, J. Comp. Graph. Stat. 8 (1999) 69–92.

[2] García-Cortés L., Sorensen D., On a multivariate implementation of the Gibbs sampler, Genet. Select. Evol. 28 (1996) 121–126.

[3] Gilmour A., Cullis B.R., Welham S.J., Thompson R., ASREML (1999), mimeo 177pp.

[4] Jamrozik J., Schaeffer L.R., Estimates of genetic parameters for a test day model with random regressions for yield traits of first lactation Holsteins, J. Dairy Sci. 80 (1997) 762–770.

[5] Meyer K., "DXMRR" - a program to estimate covariance functions for longitudinal data by restricted maximum likelihood., in: Proceedings Sixth World Congr. Genet. Appl. Livest. Prod., Armidale, Australia, vol. 27 (1998), pp. 465–466.

[6] Misztal I., Strabel T., Jamrozik J., Mantyssari E.A., Meuwissen T.H.E., Strategies for estimating the parameters needed for different test-day models, J. Dairy Sci. 83 (2000) 1125–1134.

[7] Rekaya R., Carabaño M.J., Toro M.A., Use of test day yields for the genetic evaluation of production traits in Holstein-Friesian cattle, Livest. Prod. Sci. 57 (1999) 203–217.

[8] Sorensen D., Gibbs Sampling in Quantitative Genetics, Tech. Rep. Internal report no. 82, Danish Institute of Animal Science (1996).

[9] Stranden I., Lidauer M., Solving large mixed linear models using preconditioned conjugate gradient iteration, J. Dairy Sci. 82 (1999) 2779–2787.

[10] Tsuruta S., Misztal I., Stranden I., Use of preconditioned conjugate gradient algorithm as generic solver for mixed-model equations in animal breeding applications, J. Anim. Sci. 79 (2001) 1166–1172.