UNIVERSITY OF GEORGIA
College of Agricultural & Environmental Sciences
Animal Breeding and Genetics Group

Introduction to BLUPF90 software suite

Daniela Lourenco
BLUPF90 TEAM – 08/2024

UNIVERSITÀ DEGLI STUDI FIRENZE

BLUPF90 software suite

- Collection of software
  - Fortran ≥ 90
    - Fortran = Formula Translation System
    - Fortran = Formula Translator
    - First compiler in 1957 by IBM

FEATURE ARTICLE: SOFTWARE FOR SCIENTIFIC COMPUTING

The State of Fortran

Laurence J. Kedward, *University of Bristol, Bristol, BS8 1TH, U.K.*
Bálint Aradi, *University of Bremen, 28359, Bremen, Germany*
Ondřej Čertík, *Los Alamos National Laboratory, Los Alamos, NM, 87544, USA*
Milan Curcic, *University of Miami, Miami, FL, 33149, USA*
Sebastian Ehlert, *Institut für Physikalische und Theoretische Chemie, Universität Bonn, 53115, Bonn, Germany*
Philipp Engel, *Technische Universität Berlin, 10623, Berlin, Germany*
Rohit Goswami, *Quansight Austin, Austin, TX, 78735, USA*
Michael Hirsch, *Boston University, Boston, MA, 02215, USA*
Asdrubal Lozada-Blanco, *University of São Paulo, São Carlos, 13566-590, Brazil*
Vincent Magnin, *Université Lille, CNRS, Centrale Lille, Université Polytechnique Hauts-de-France, IEMN, 59000, Lille, France*
Arjen Markus, *Deltares Research Institute, 2629 HV, Delft, The Netherlands*
Emanuele Pagone, *Cranfield University, Cranfield, MK43 0AL, U.K.*
Ivan Pribec, *Technical University of Munich, 80333, Munich , Germany*
Brad Richardson, *Archaeologic, Inc., Berkeley, CA, 94707, USA*
Harris Snyder, *Structura Biotechnology Inc., Toronto, ON, M5G 1S5, Canada*
John Urban, *High-Performance Computing Consultant, USA*
Jérémie Vandenplas, *Wageningen University Research, 6700 AH, Wageningen, The Netherlands*

A community of developers has formed to modernize the Fortran ecosystem. In this article, we describe the high-level features of Fortran that continue to make it a good choice for scientists and engineers in the 21st century. Ongoing efforts include the development of a Fortran standard library and package manager, the fostering of a friendly and welcoming online community, improved compiler support, and language feature development. The lessons learned are common across contemporary programming languages and help reduce the learning curve and increase adoption of Fortran.

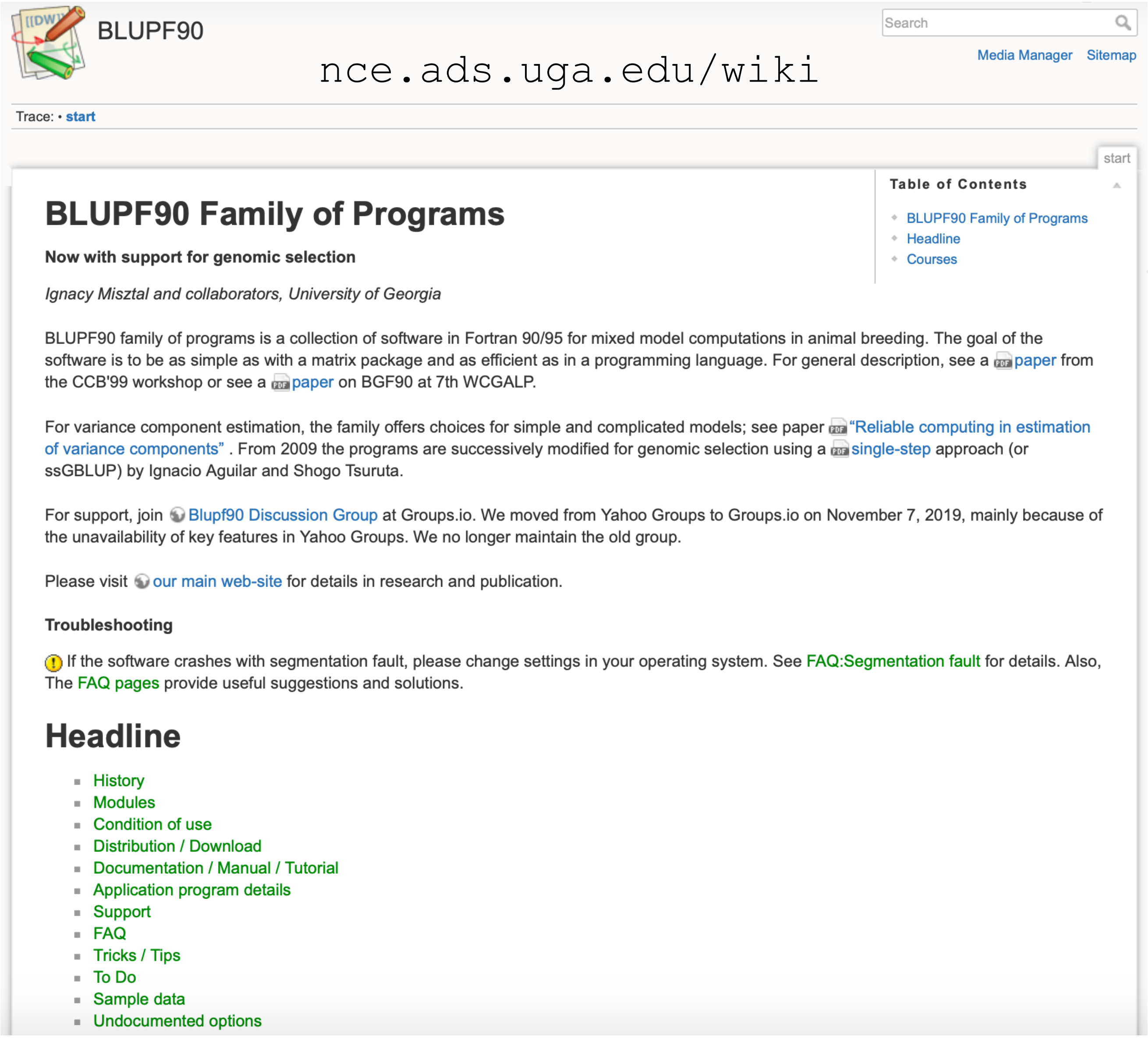
Fortran is a high-level programming language primarily used to solve scientific and engineering problems. It has been under active development since its inception under John Backus at IBM in 1954 to the present day. The initial goal was to ease the translation of mathematical formulas to optimized machine code instructions, a concept now known as compilation. The intuitive abstraction of mathematical procedures enabled rapid development of numerical solutions to scientific problems, at a time when most programs were still hand-coded in assembly language. Following the release of its first implementation in 1957, the language was adopted by the scientific and engineering communities for writing numerical programs. As a result, the language was quickly ported to several computer architectures such that Fortran is accepted as being the first cross-platform programming language.

The ISO Fortran Standard and its maintenance of backwards compatibility provide guarantees for

---

1521-9615 © 2022 IEEE
Digital Object Identifier 10.1109/MCSE.2022.3159862
Date of publication 16 March 2022; date of current version 27 May 2022.

BLUPF90 software suite



nce.ads.uga.edu/wiki

Table of Contents

BLUPF90 Family of Programs

Now with support for genomic selection

Ignacy Misztal and collaborators, University of Georgia

BLUPF90 family of programs is a collection of software in Fortran 90/95 for mixed model computations in animal breeding. The goal of the software is to be as simple as with a matrix package and as efficient as in a programming language. For general description, see a paper from the CCB'99 workshop or see a paper on BGF90 at 7th WCGALP.

For variance component estimation, the family offers choices for simple and complicated models; see paper "Reliable computing in estimation of variance components" . From 2009 the programs are successively modified for genomic selection using a single-step approach (or ssGBLUP) by Ignacio Aguilar and Shogo Tsuruta.

For support, join Blupf90 Discussion Group at Groups.io. We moved from Yahoo Groups to Groups.io on November 7, 2019, mainly because of the unavailability of key features in Yahoo Groups. We no longer maintain the old group.

Please visit our main web-site for details in research and publication.

Troubleshooting

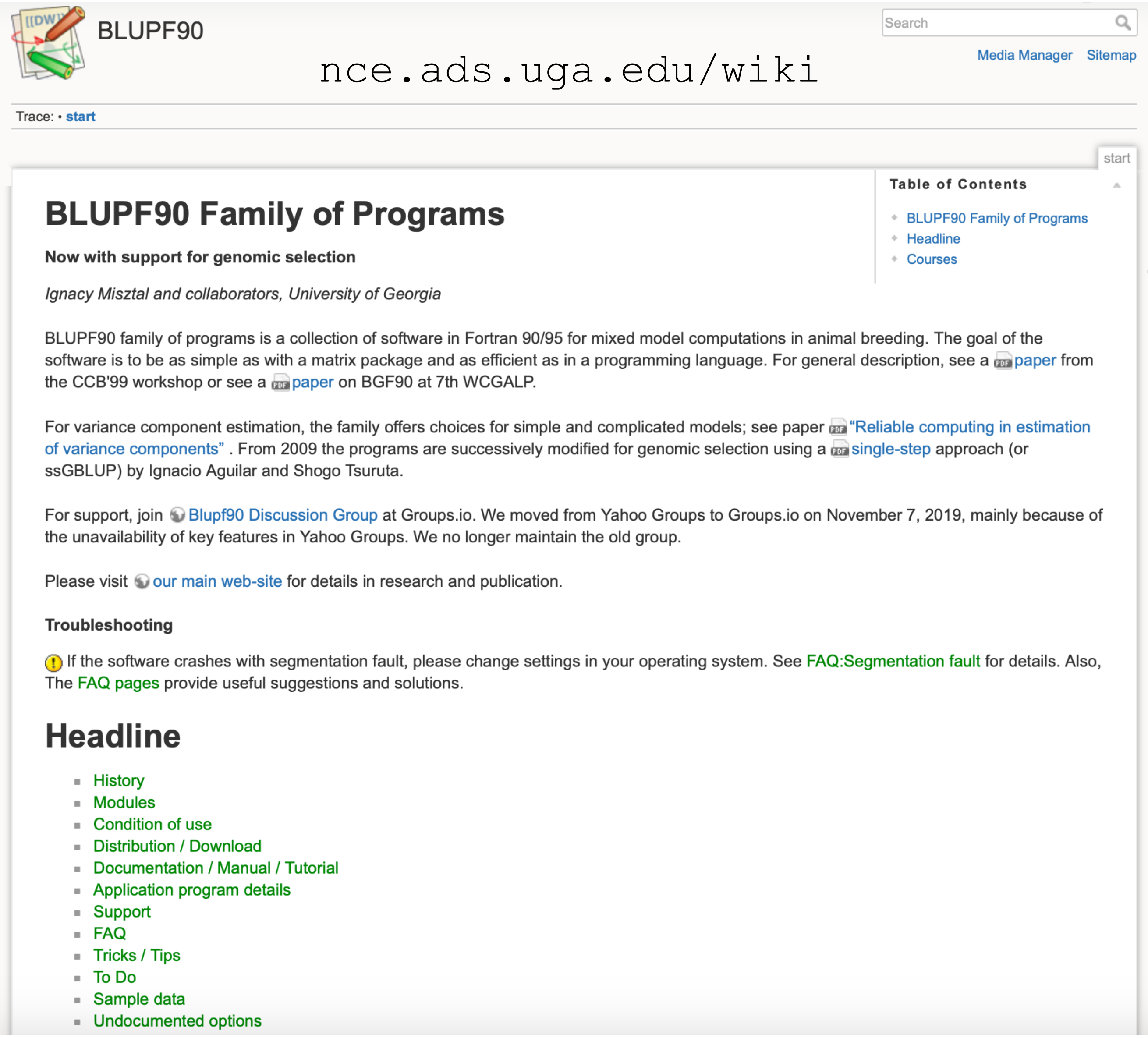
If the software crashes with segmentation fault, please change settings in your operating system. See FAQ:Segmentation fault for details. Also, The FAQ pages provide useful suggestions and solutions.

Headline

- History
- Modules
- Condition of use
- Distribution / Download
- Documentation / Manual / Tutorial
- Application program details
- Support
- FAQ
- Tricks / Tips
- To Do
- Sample data
- Undocumented options

- Collection of software
  - Fortran ≥ 90
  - Computations in AB & G
- Since 1997/1998 by Ignacy Misztal
- Several developers + collaborators
- Simple, efficient, and comprehensive
  - Very general models

BLUPF90 software suite



nce.ads.uga.edu/wiki

Table of Contents

BLUPF90 Family of Programs

Now with support for genomic selection

Ignacy Misztal and collaborators, University of Georgia

BLUPF90 family of programs is a collection of software in Fortran 90/95 for mixed model computations in animal breeding. The goal of the software is to be as simple as with a matrix package and as efficient as in a programming language. For general description, see a paper from the CCB'99 workshop or see a paper on BGF90 at 7th WCGALP.

For variance component estimation, the family offers choices for simple and complicated models; see paper "Reliable computing in estimation of variance components" . From 2009 the programs are successively modified for genomic selection using a single-step approach (or ssGBLUP) by Ignacio Aguilar and Shogo Tsuruta.

For support, join Blupf90 Discussion Group at Groups.io. We moved from Yahoo Groups to Groups.io on November 7, 2019, mainly because of the unavailability of key features in Yahoo Groups. We no longer maintain the old group.

Please visit our main web-site for details in research and publication.

Troubleshooting

If the software crashes with segmentation fault, please change settings in your operating system. See FAQ:Segmentation fault for details. Also, The FAQ pages provide useful suggestions and solutions.

Headline

- History
- Modules
- Condition of use
- Distribution / Download
- Documentation / Manual / Tutorial
- Application program details
- Support
- FAQ
- Tricks / Tips
- To Do
- Sample data
- Undocumented options

- No GUI (graphical user interface)!!!
- First idea: to solve the MME

$$\begin{bmatrix} \mathbf{X'X} & \mathbf{X'W} \\ \mathbf{W'X} & \mathbf{W'W}+\mathbf{A}^{-1}\frac{\sigma_e^2}{\sigma_a^2} \end{bmatrix} \boxed{\begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{u}} \end{bmatrix}} = \begin{bmatrix} \mathbf{X'y} \\ \mathbf{W'y} \end{bmatrix}$$

- First software: blupf90

- Second idea: VCE

$$\begin{bmatrix} \mathbf{X'X} & \mathbf{X'W} \\ \mathbf{W'X} & \mathbf{W'W}+\mathbf{A}^{-1}\boxed{\frac{\sigma_e^2}{\sigma_a^2}} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X'y} \\ \mathbf{W'y} \end{bmatrix}$$

- Software: remlf90, airemlf90 , gibbsf90

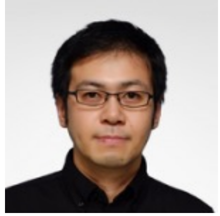
BLUPF90 software main developers

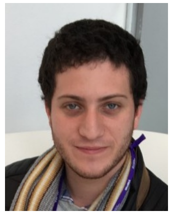

Ignacy
Misztal


Shogo
Tsuruta


Andres
Legarra


Ignacio
Aguilar


Yutaka
Masuda


Matias
Bermann

- + Several contributors
- Research turns into code

- Which programs?

BLUPF90 software suite

blupf90

BLUP with explicit equations

remlf90

Expectation Maximization REML

airemlf90

Average Information REML

gibbsXf90

Bayesian Analyses – linear traits

thrgibbsXf90

Bayesian Analyses – categorical traits

postgibbsf90

Post-analyses of Gibbs samples

Genomics

preGSf90

Processing of SNP data (QC + matrices)

qcf90 ✖

QC of large SNP data

postGSf90

Estimation of SNP effects and GWAS

predf90 ✖

Prediction of GEBV based on SNP effects

seekparentf90 ✖

Parentage verification (SNP and pedigree)

predictf90

Adjusted and predicted phenotypes

validationf90

Perform validation of predictions

Large-scale

blup90iod2

blup90iod2OMP1

blup90iod3

cblup90iod2

cblup90iod2OMP1

accf90

accf90GS

accf90GS2

accf90GS3

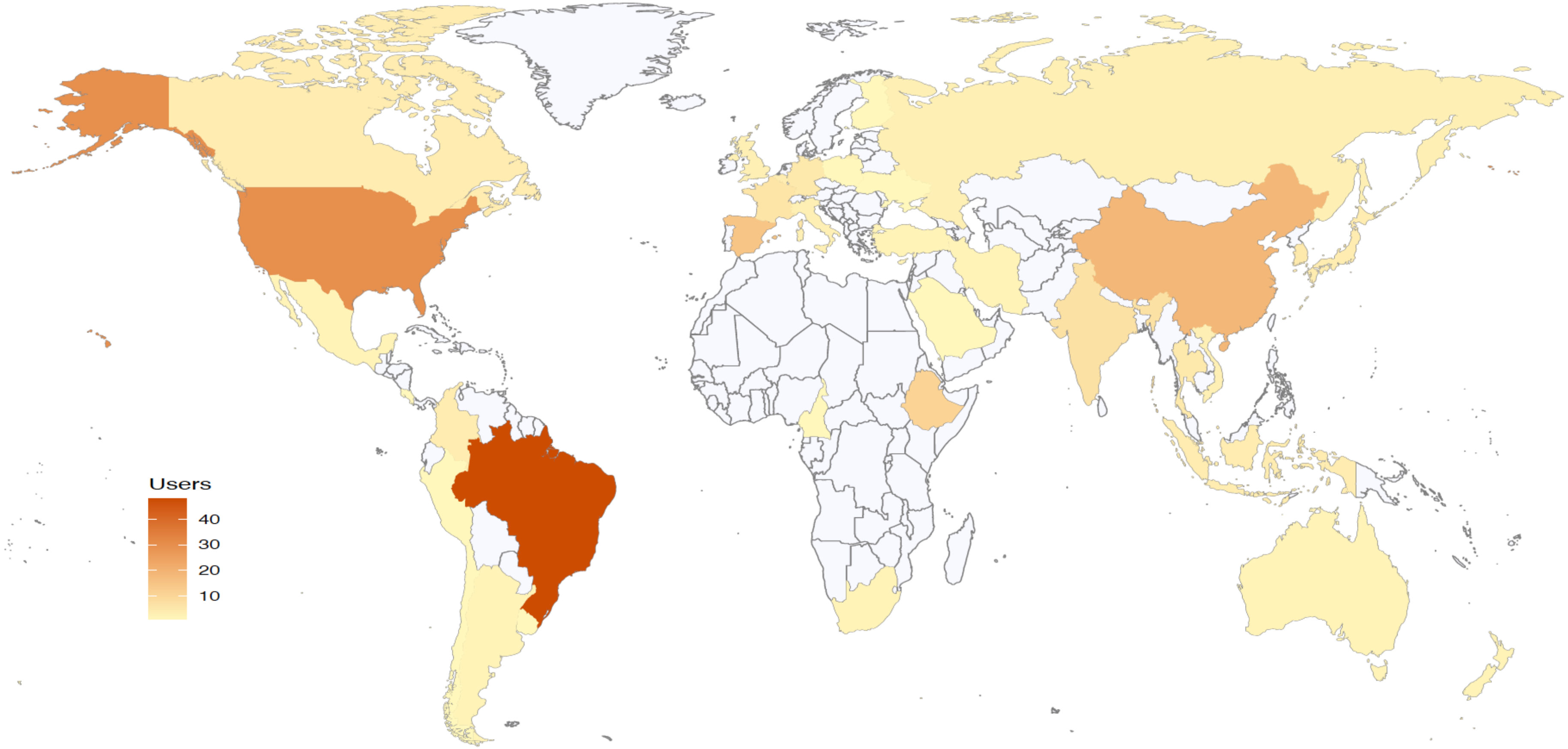
BLUPF90 software suite

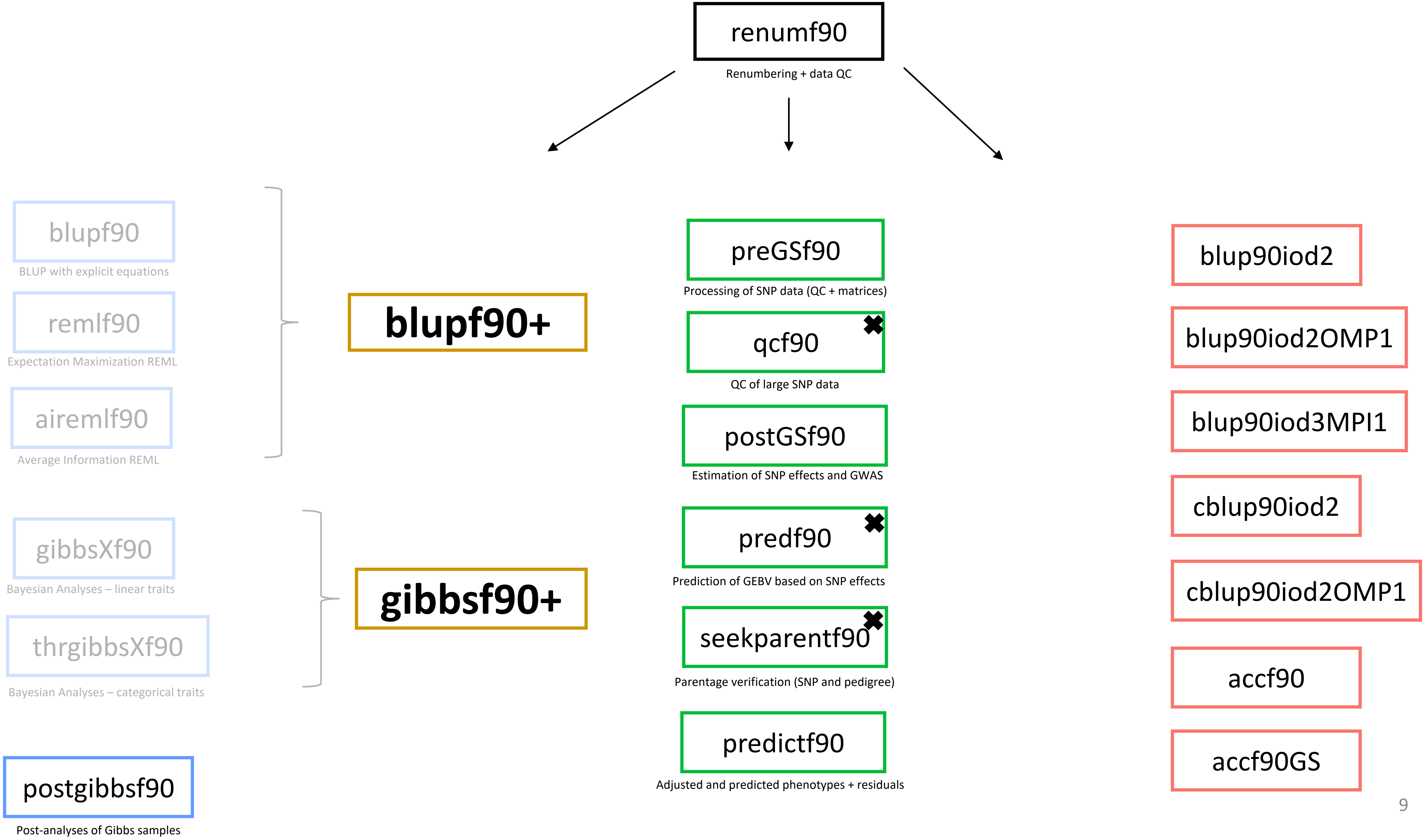
March/2024

- 1,684,059 accesses to the nce server
- 1074 true binaries downloads (without duplicates)
- 239 users (IP with at least one binary download)
- BLUPF90+ has the most downloads
- Brazil is the countries with the most users
- Windows is the most used OS

BLUPF90 software suite

Users around the world





renumf90

Renumbering + data QC

blupf90

BLUP with explicit equations

remlf90

Expectation Maximization REML

airemlf90

Average Information REML

blupf90+

gibbsXf90

Bayesian Analyses – linear traits

thrgibbsXf90

Bayesian Analyses – categorical traits

gibbsf90+

postgibbsf90

Post-analyses of Gibbs samples

preGSf90

Processing of SNP data (QC + matrices)

qcf90 ✖

QC of large SNP data

postGSf90

Estimation of SNP effects and GWAS

predf90 ✖

Prediction of GEBV based on SNP effects

seekparentf90 ✖

Parentage verification (SNP and pedigree)

predictf90

Adjusted and predicted phenotypes + residuals

blup90iod2

blup90iod2OMP1

blup90iod3MPI1

cblup90iod2

cblup90iod2OMP1

accf90

accf90GS

RENUMF90

The **renumbering software** for the BLUPF90 suite

RENUMF90

- **Renumbers data and pedigree**
- **Creates a parameter file for BLUPF90 family**
  - **Parameter file can be modified by the users for new models**
- **Traces back pedigree for individuals in the data**
- **Performs comprehensive pedigree checks**
- **Provides data statistics**
- **Creates an Xref file for genotyped individuals**
- **Computes inbreeding by default in v ≥ 1.157** NEW

RENUMF90

- **Supports**
  - virtually any dataset
  - multiple traits
  - different models (effects) per trait
  - alphanumeric and numeric fields
  - unknown parent groups
  - covariates for random regression models

RENUMF90 – Input files

- **Parameter file with specific keywords**
- **Data file and pedigree file as flat files**
    - Columns separated by at least one SPACE
    - No TABS !!!! (current version checks for it)
    - Input files cannot contain character #
    - Missing sire/dams must have code 0
    - code 00 is treated as a known animal

RENUMF90 – Output files

- **Creates files to be used by BLUPF90 family**
  - renf90.inb - file with inbreeding
  - renf90.tables - cross reference file with renumbered and original effects
  - renf90.fields - description of the effects in each field of renf90.dat

  - renf90.dat - renumbered data
  - renaddxx.ped - renumbered pedigree + statistics
  - renf90.par - new parameter file

RENUMF90 parameter file

MANDATORY

Keyword	possible value	description
DATAFILE	character	The name of data file to be processed
TRAITS	integer	Position for phenotype (trait) in the data file
FIELDS_PASSED TO OUTPUT	integer	Position for the columns in the original data that will be passed to the renumbered data without changes Keep empty if not needed
WEIGHT(S)	integer	The position(s) for weight in the data file Keep empty if not needed
RESIDUAL_VARIANCE	real value(s)	Residual (co)variance
EFFECT	(next slide …)	Description of an effect Repeatable – 1 for each effect in the model

RENUMF90 parameter file

Effects

Keyword	Possible value	effect type	form
EFFECT	integer (column where the effect is)	cross	alpha
			numer
		cov	

Keyword (only for covariables)	Possible value	form
NESTED	integer (column where the effect is)	alpha
		numer

RENUMF90 parameter file

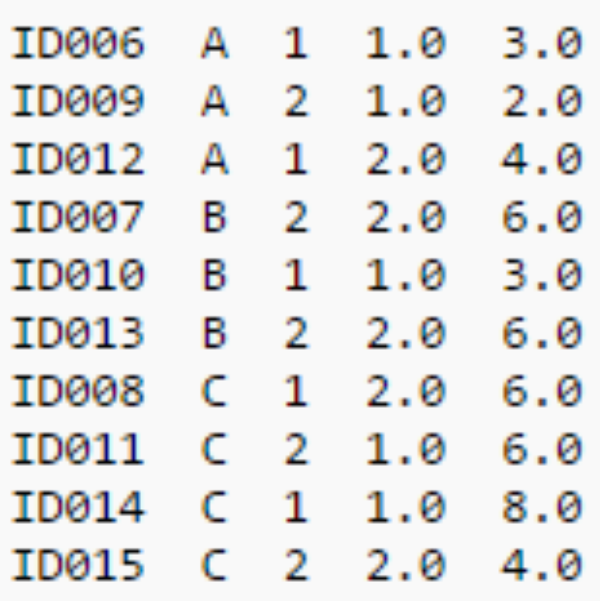
Model: $y = farm + sex + \beta\ age + e$

```
DATAFILE
data1.txt
TRAITS
5
FIELDS_PASSED TO OUTPUT
2
WEIGHT(S)

RESIDUAL_VARIANCE
1.0
EFFECT        #1st effect - farm
2 cross alpha
EFFECT        #2nd effect - sex
3 cross numer
EFFECT        #3rd effect - age
4 cov
```

Fixed linear model

data1.txt

ID farm sex age phen



```
ID006   A   1   1.0   3.0
ID009   A   2   1.0   2.0
ID012   A   1   2.0   4.0
ID007   B   2   2.0   6.0
ID010   B   1   1.0   3.0
ID013   B   2   2.0   6.0
ID008   C   1   2.0   6.0
ID011   C   2   1.0   6.0
ID014   C   1   1.0   8.0
ID015   C   2   2.0   4.0
```

RENUMF90 parameter file

Random Effects

Keyword after EFFECT	possible value	description
RANDOM	diagonal animal	Non-correlated Correlation structure among animals

Keyword	possible value	description
OPTIONAL	pe mat mpe	Permanent environmental Maternal Permanent environmental maternal (only if mat is used)

RENUMF90 parameter file

Random effects file section

Keyword after RANDOM (animal only)	**possible value**	**description**
FILE	character	Name of the pedigree file for animal models only

Keyword after FILE **(for RANDOM animal only)**	**possible value**	**description**
FILE_POS	integer	Specifies positions in the pedigree for ani sire dam alternate_dam yob Default: 1 2 3 0 0 *If maternal effect alternate_dam*

Keyword **(for RANDOM animal only)**	**possible value**	**description**
SNP_FILE	character	Optional: If genomic info is to be used Name of the SNP file Format: ID 011122211155152222

RENUMF90 parameter file

Pedigree options

Keyword **(for RANDOM animal only)**	**possible value**	**description**
PED_DEPTH	Integer	Optional Specifies the depth of pedigree search Default = 3 All pedigree = 0

RENUMF90 parameter file

Unknown Parent Group options

Keyword **(for RANDOM animal only)**	**possible value**	**description**
UPG_TYPE		Optional
	yob 1990 1992 …	UPG assigned based on yob
	in_pedigrees	Missing parent receives -x x is the UPG number
	group_unisex	UPG based on the information in pedigree Ex. UPG by breed **FILE_POS** 1 2 3 0 0 4 #the 6th field indicates which column the UPG code is in the pedigree
	group_sex	Separate UPG code for unknown sire and dam **FILE_POS** 1 2 3 0 0 4 5 #the 6th and 7th fields indicate which columns the UPG codes are in the pedigree

RENUMF90 parameter file

Inbreeding option

Keyword **(for RANDOM animal only)**	**possible value**	**description**
INBREEDING		Default in RENUMF90 ≥ v1.157
	pedigree	Calculates inbreeding code and saves it in the renumbered pedigree file (Default in RENUMF90 ≥ v1.157)
	File `<name>`	Reads inbreeding from an external file format: original_ID inbreeding (0 to 1)
	self `x`	Calculates inbreeding with selfing `x` is the column in the pedigree file with the number of selfing generations
	No-inbreeding	Turn inbreeding calculation off in RENUMF90 ≥ v1.157

RENUMF90 parameter file

(CO)VARIANCES for Random effects

Keyword	possible value	description
(CO)VARIANCES	real	(co)variance for the animal effect dimension should account for number of traits and random correlated effects

32.79	-7.22	-11.07
-7.22	258.06	87.66
-11.07	87.66	194.34

RENUMF90 parameter file

(CO)VARIANCES structure

- 3 trait (T1-T3) and 2 correlated effects (E1-E2)

		E1			E2		
		T1	T2	T3	T1	T2	T3
E1	T1						
	T2						
	T3						
E2	T1						
	T2						
	T3						

RENUMF90 parameter file

(CO)VARIANCES for Random effects

Keyword	possible value	description
(CO)VARIANCES	real	(co)variance for non-correlated random effects

Keyword	possible value	description
(CO)VARIANCES_PE	real	(co)variance for the PE effect if present

Keyword	possible value	description
(CO)VARIANCES_MPE	real	(co)variance for the MPE effect if present

RENUMF90 parameter file

Creating interaction between effects

Keyword	**possible value**	**description**
COMBINE	integer	Should come before DATAFILE COMBINE 7 2 3 4 Columns 2, 3, 4 are combined into 7 They can be numer or alpha

RENUMF90 parameter file

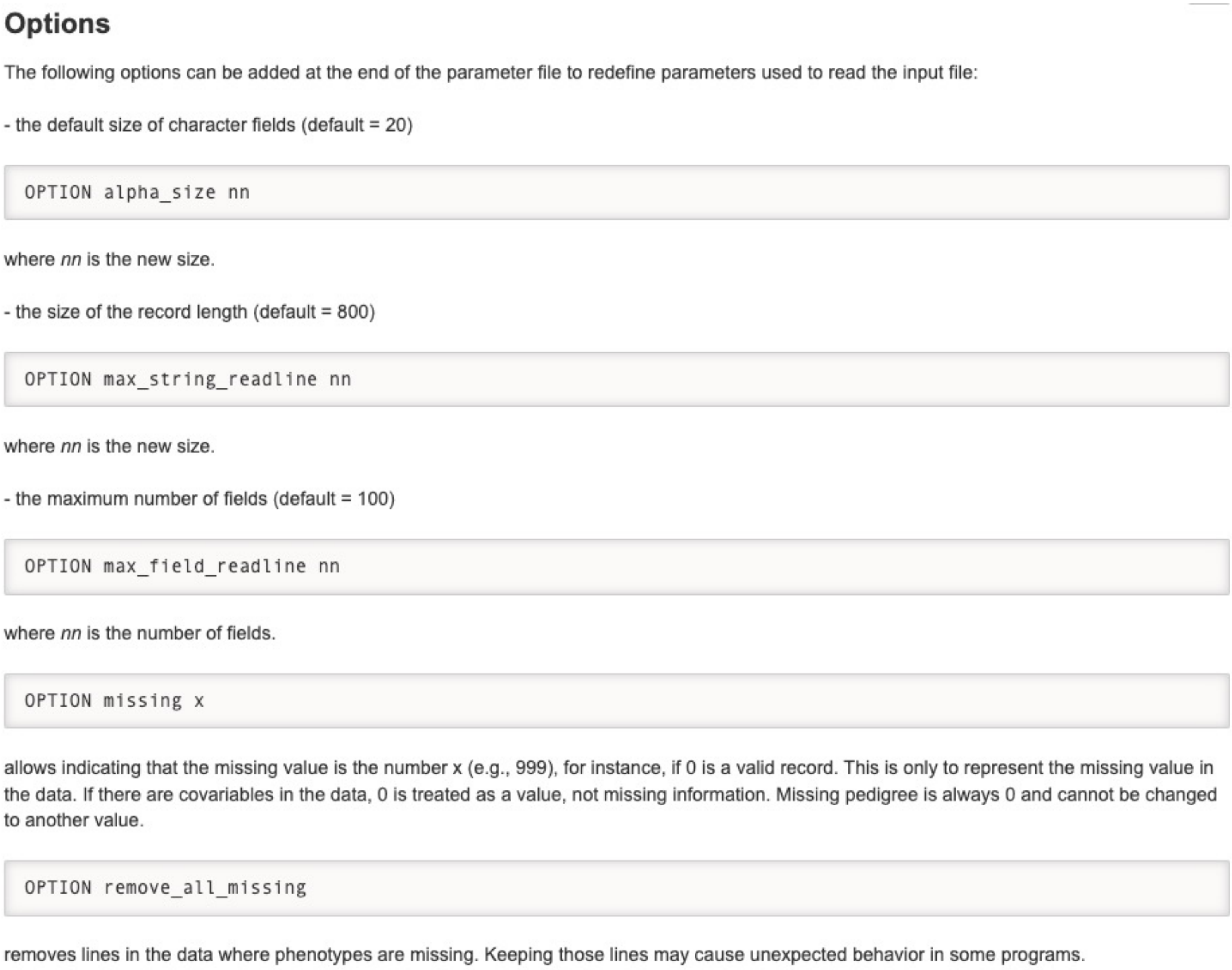
Keyword	optional	possible values
COMBINE	optional	definition of new field as a combination of existing fields
DATAFILE	mandatory	name of raw data file
TRAITS	mandatory	positions of observations in the raw data file
FIELDS_PASSED	mandatory	positions of items in the raw data file to be passed to renf90.dat
WEIGHT(S)	mandatory	positions of weights in the raw data file
RESIDUAL_VARIANCE	mandatory	residual covariance matrix
EFFECT	mandatory	effect description
NESTED	optional	positions of nested covariates
RANDOM	optional	declaration of random effect
OPTIONAL	optional	declaration of MAT, PE, MPE
FILE	optional	name of raw pedigree file
FILE_POS	optional	positions of animal ID, sire ID, and dam ID
SNP_FILE	optional	name of SNP marker file
PED_DEPTH	optional	the maximum generation back from animals with phenotype and/or genotype
GEN_INT	optional	generation interval to set unknown parent groups (UPG)
REC_SEX	optional	check if records are found in specific sex
UPG_TYPE	optional	UPG specification
INBREEDING	optional	create pedigree file with inbreeding code
RANDOM_REGRESSION	optional	put covariates for random regressions
RR_POSITION	optional	positions of covariates for random regressions
(CO)VARIANCES	optional	covariance components
(CO)VARIANCES_PE	optional	covariance components for animal PE effects
(CO)VARIANCES_MPE	optional	covariance components for maternal PE effects
OPTION	optional	option parameters

RENUMF90 parameter file

- If the data and pedigree files have header

```
#Parameter file for renumf90
DATAFILE
data
SKIP_HEADER
1
TRAITS
3
FIELDS_PASSED TO OUTPUT
1 #Line_ID
WEIGHT(S)

RESIDUAL_VARIANCE
1.0
EFFECT
2 cross alpha
EFFECT
1 cross alpha
RANDOM
animal
FILE
ped
SKIP_HEADER
1
(CO)VARIANCES
1.0
OPTION sol se
```

RENUMF90 options



Options

The following options can be added at the end of the parameter file to redefine parameters used to read the input file:

- the default size of character fields (default = 20)

```
OPTION alpha_size nn
```

where *nn* is the new size.

- the size of the record length (default = 800)

```
OPTION max_string_readline nn
```

where *nn* is the new size.

- the maximum number of fields (default = 100)

```
OPTION max_field_readline nn
```

where *nn* is the number of fields.

```
OPTION missing x
```

allows indicating that the missing value is the number x (e.g., 999), for instance, if 0 is a valid record. This is only to represent the missing value in the data. If there are covariables in the data, 0 is treated as a value, not missing information. Missing pedigree is always 0 and cannot be changed to another value.

```
OPTION remove_all_missing
```

removes lines in the data where phenotypes are missing. Keeping those lines may cause unexpected behavior in some programs.

RENUMF90 parameter file

Hints

- Keyword EFFECT is repeated as many times as effects in the model
- If (CO)VARIANCES for any effect are missing, default matrix with 1.0 in diagonal and 0.1 on off-diagonal will be used

RENUMF90 parameter file

Options passed to blupf90

- All lines that begin with the keyword OPTION are passed to parameter file `renf90.par`
  - Unless they are specific to renumf90
- This allows automation of process by using scripts
- For example:
  - `OPTION sol se`
  - `OPTION use_yams`

RENUMF90 parameter file

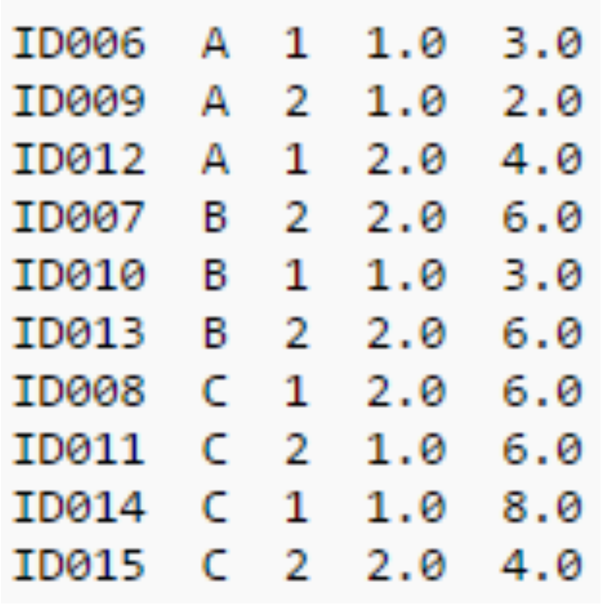
Model: $y = farm + sex + \beta\ age + e$

```
DATAFILE
data1.txt
TRAITS
5
FIELDS_PASSED TO OUTPUT
2
WEIGHT(S)

RESIDUAL_VARIANCE
1.0
EFFECT          #1st effect - farm
2 cross alpha
RANDOM
diagonal
(CO)VARIANCES
0.5
EFFECT         #2nd effect - sex
3 cross numer
EFFECT         #3rd effect - age
4 cov
```

What if we want to consider farm as random with variance = 0.5?

data1.txt

ID farm sex age phen



```
ID006  A  1  1.0  3.0
ID009  A  2  1.0  2.0
ID012  A  1  2.0  4.0
ID007  B  2  2.0  6.0
ID010  B  1  1.0  3.0
ID013  B  2  2.0  6.0
ID008  C  1  2.0  6.0
ID011  C  2  1.0  6.0
ID014  C  1  1.0  8.0
ID015  C  2  2.0  4.0
```

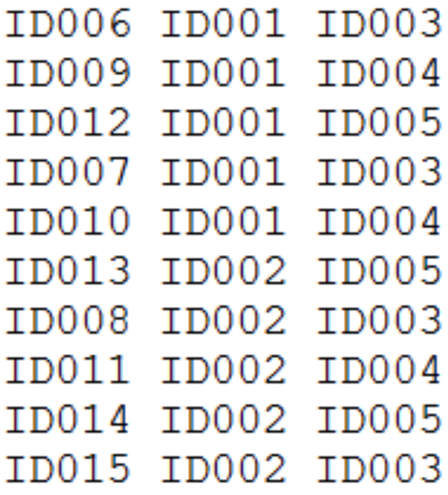
RENUMF90 parameter file

Model: $y = farm + sex + \beta\, age + animal + e$

```
DATAFILE
data1.txt
TRAITS
5
FIELDS_PASSED TO OUTPUT
2
WEIGHT(S)

RESIDUAL_VARIANCE
1.0
EFFECT        #1st effect - farm
2 cross alpha
EFFECT        #2nd effect - sex
3 cross numer
EFFECT        #3rd effect - age
4 cov
EFFECT        #4th effect - animal
1 cross alpha
RANDOM
animal
FILE
ped1.txt
FILE_POS
1 2 3 0 0
(CO)VARIANCES
0.2
```

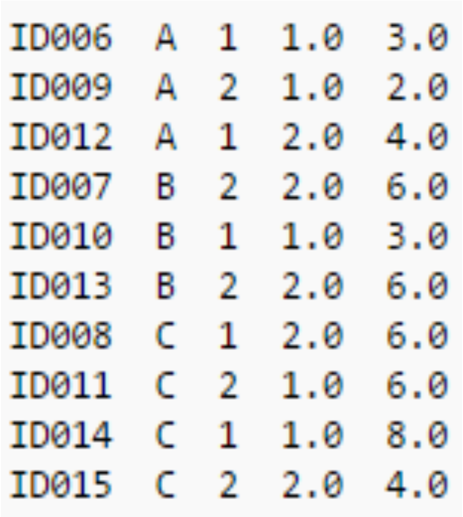
What if we want to consider animal effect as random with σ_u^2 = 0.2?

ped1.txt



ID	Sire	Dam
ID006	ID001	ID003
ID009	ID001	ID004
ID012	ID001	ID005
ID007	ID001	ID003
ID010	ID001	ID004
ID013	ID002	ID005
ID008	ID002	ID003
ID011	ID002	ID004
ID014	ID002	ID005
ID015	ID002	ID003

data1.txt



ID	farm	sex	age	phen
ID006	A	1	1.0	3.0
ID009	A	2	1.0	2.0
ID012	A	1	2.0	4.0
ID007	B	2	2.0	6.0
ID010	B	1	1.0	3.0
ID013	B	2	2.0	6.0
ID008	C	1	2.0	6.0
ID011	C	2	1.0	6.0
ID014	C	1	1.0	8.0
ID015	C	2	2.0	4.0

RENUMF90 output files

Pedigree file: `renaddxx.ped`

Data file: `renf90.dat`

Parameter file: `renf90.par`

Inbreeding file: `renf90.inb`

Renumbering table: `renf90.table`

Fields table: `renf90.fields`

RENUMF90 output files
Pedigree file: `renaddxx.ped`

1. Animal ID (from 1)
2. Parent 1 ID or UPG number for parent 1
3. Parent 2 ID or UPG number for parent 2
4. 3 minus number of known parents
5. Known or estimated year of birth (0 if not provided)
6. Number of known parents

   if genotyped: 10+number of known parents
7. Number of records
8. Number of progeny as parent 1
9. Number of progeny as parent 2
10. Original animal ID

RENUMF90 output files
Pedigree file: `renaddxx.ped`

- As inbreeding is default:

  Column 4:

  inb/upg code = 4000 / [(1+ms)(1-Fs) + (1+md)(1-Fd)]

  ms (md) is 0 if sire (dam) is known, and 1 otherwise
  Fs (Fd) is the coefficient of inbreeding of sire (dam)

  Ex: For an animal with both parents known and F=0
  inb/upg code = 2000

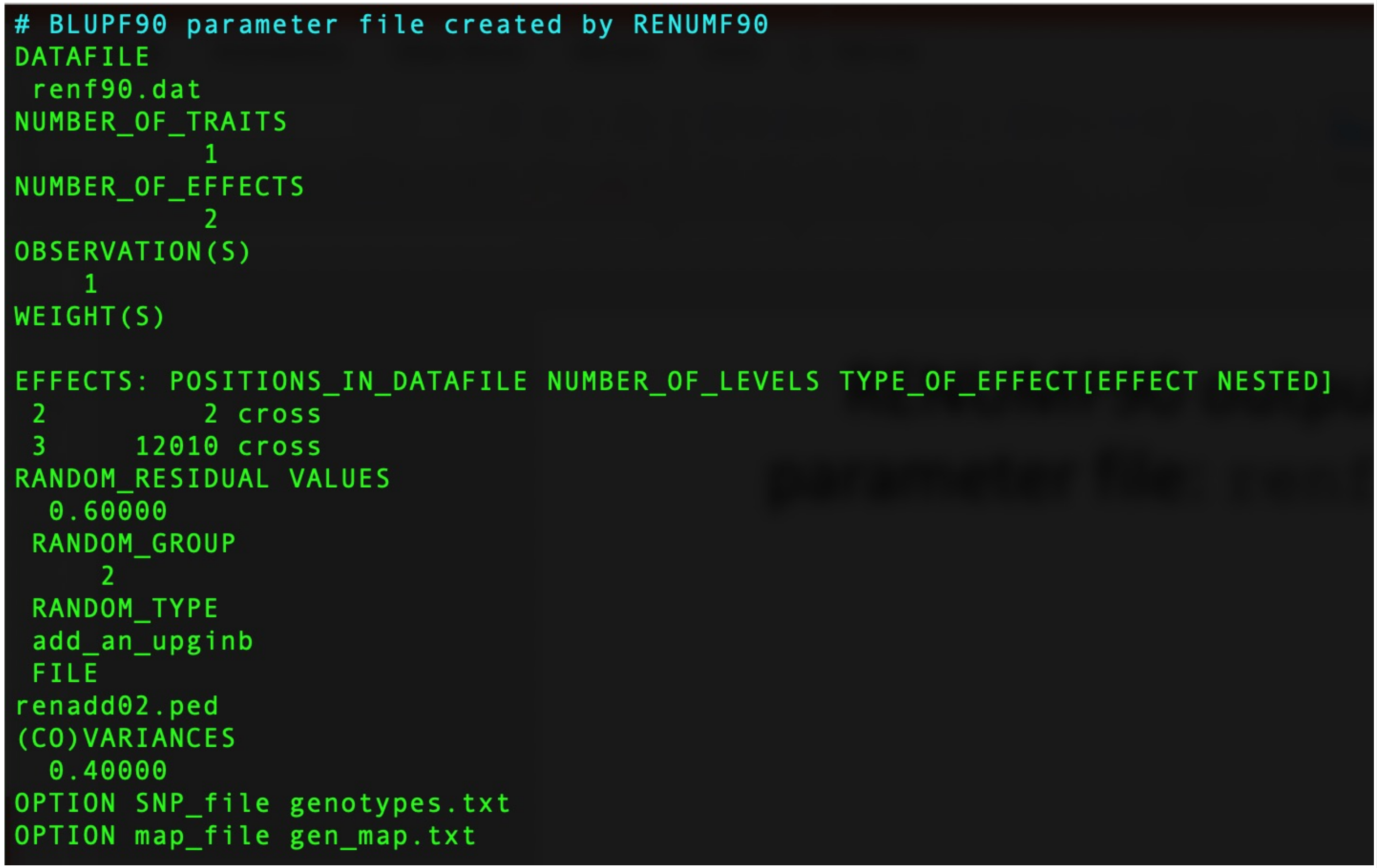
RENUMF90 output files
Data file: `renf90.dat`

1. Phenotype
2. Effect 1
3. Effect 2

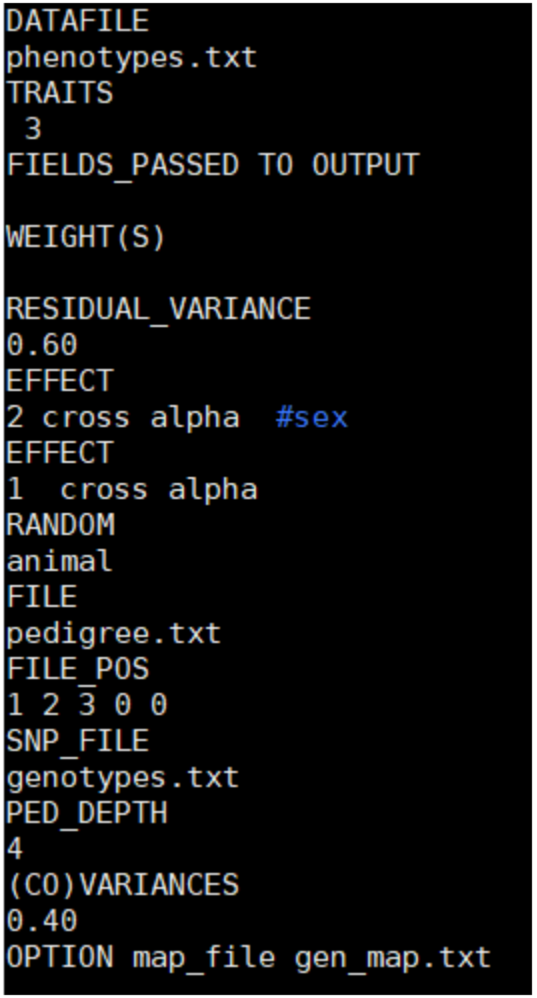
…. …

N. Effect n

RENUMF90 output files
parameter file: `renf90.par`



```
# BLUPF90 parameter file created by RENUMF90
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
           1
NUMBER_OF_EFFECTS
           2
OBSERVATION(S)
    1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
 2         2 cross
 3     12010 cross
RANDOM_RESIDUAL VALUES
  0.60000
 RANDOM_GROUP
     2
 RANDOM_TYPE
 add_an_upginb
 FILE
renadd02.ped
(CO)VARIANCES
  0.40000
OPTION SNP_file genotypes.txt
OPTION map_file gen_map.txt
```



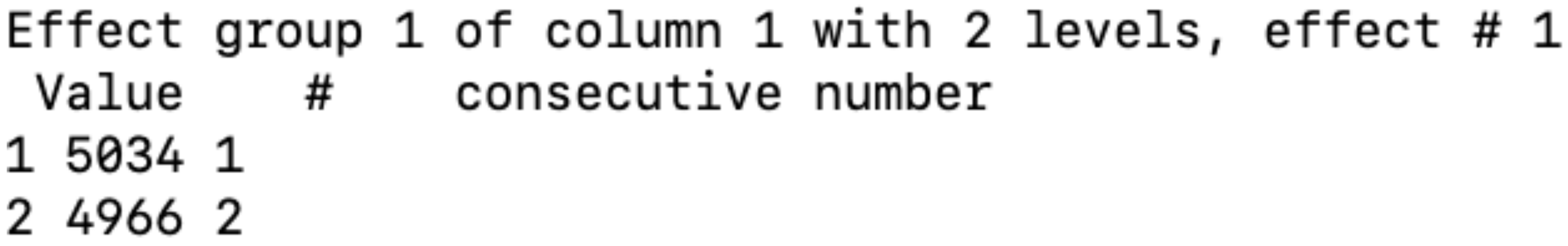
```
DATAFILE
phenotypes.txt
TRAITS
 3
FIELDS_PASSED TO OUTPUT

WEIGHT(S)

RESIDUAL_VARIANCE
0.60
EFFECT
2 cross alpha  #sex
EFFECT
1  cross alpha
RANDOM
animal
FILE
pedigree.txt
FILE_POS
1 2 3 0 0
SNP_FILE
genotypes.txt
PED_DEPTH
4
(CO)VARIANCES
0.40
OPTION map_file gen_map.txt
```

RENUMF90 output files
Inbreeding file: `renf90.inb`

- renf90.inb will have:

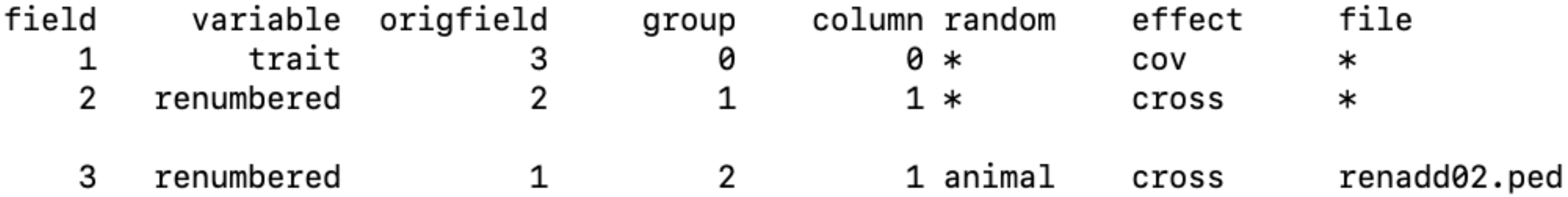
```
 origID      Inbreeding  newID
A71342462   0.059204   6927175
A17194772   0.032106        29
A13476873   0.002958   6550405
A1ZEP4813   0.000000        61
A14347077   0.019187   6550336
A64547711   0.026603        12
A71922414   0.000000   6942899
A17274771   0.019961        42
A53301967   0.000000   6550416
A4ZGF7566   0.000000       167
A3ZZS6645   0.000000        25
A07818367   0.000000   7117564
A17354770   0.050361        55
A53401908   0.000000        31
A13556872   0.063467   6550439
A14507075   0.071151   6550347
```

RENUMF90 output files
Renumbering table: `renf90.tables`



```
Effect group 1 of column 1 with 2 levels, effect # 1
 Value    #    consecutive number
1 5034 1
2 4966 2
```

RENUMF90 output files
Fields table: `renf90.fields`



```
field      variable  origfield     group    column random    effect     file
    1         trait          3         0         0 *         cov        *
    2    renumbered          2         1         1 *         cross      *

    3    renumbered          1         2         1 animal    cross      renadd02.ped
```

renumf90 FAQ

1) renumf90 cannot find the data file	Check for typos
2) How to include quadratic covariable?	Column in data file
3) Error when trying to use covariable	2 cov numer
4) Fixed effects in renf90.dat are different from original	renf90.tables
5) I want to have original IDs in renf90.dat	FIELDS_PASSED TO OUTPUT
6) I used renumf90 but want to have solutions with original IDs when using blupf90+	

renumf90 FAQ

6) When and how to run renumf90?

a) Objective to compare models

Run renumf90 ONCE with the most complete model

Remove effects from renf90.par

b) Objective to compare non-genomic vs genomic model

Run renumf90 ONCE with SNP file

For non-genomic: Remove option for SNP file from renf90.par

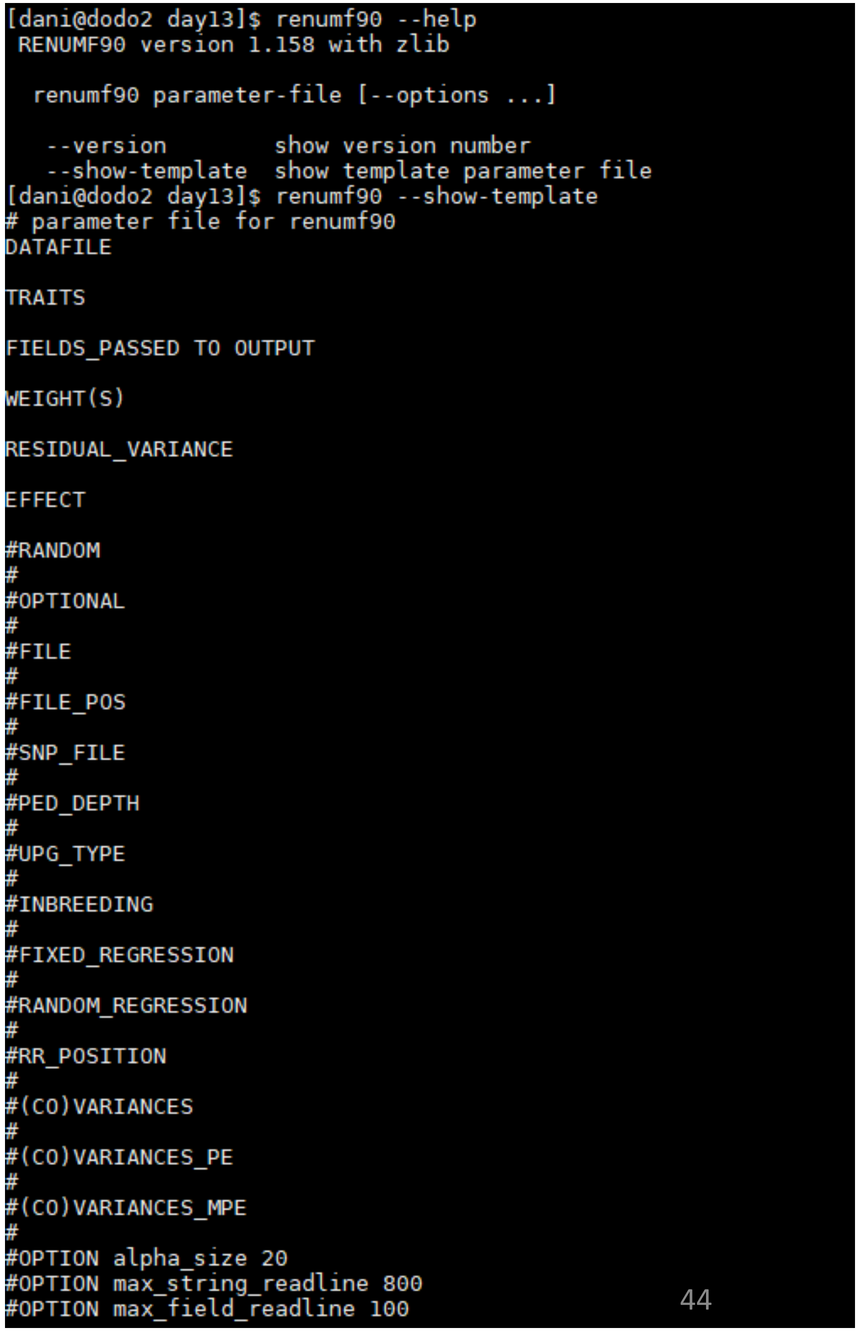
c) Objective to mask phenotypes for some animals for validation

Run renumf90 ONCE with the complete data

Remove animals from renf90.dat

renumf90 quick trick

- `renumf90 --help`
- `renumf90 --show-template`



```
[dani@dodo2 day13]$ renumf90 --help
 RENUMF90 version 1.158 with zlib

  renumf90 parameter-file [--options ...]

   --version        show version number
   --show-template  show template parameter file
[dani@dodo2 day13]$ renumf90 --show-template
# parameter file for renumf90
DATAFILE

TRAITS

FIELDS_PASSED TO OUTPUT

WEIGHT(S)

RESIDUAL_VARIANCE

EFFECT

#RANDOM
#
#OPTIONAL
#
#FILE
#
#FILE_POS
#
#SNP_FILE
#
#PED_DEPTH
#
#UPG_TYPE
#
#INBREEDING
#
#FIXED_REGRESSION
#
#RANDOM_REGRESSION
#
#RR_POSITION
#
#(CO)VARIANCES
#
#(CO)VARIANCES_PE
#
#(CO)VARIANCES_MPE
#
#OPTION alpha_size 20
#OPTION max_string_readline 800
#OPTION max_field_readline 100
```

General output from RENUMF90

- Output printed on the screen is not saved to any file!
- Should use redirection or pipes to store output or script

renumf90

```
renumf90 renum.par | tee renum.log
```

blupf90+

- `blupf90`: MME solver
- `airemlf90`: variance components using Average Information REML
- `remlf90`: variance components using Expectation Maximization REML

Mixed Model Equations Solver

Variance Components Estimation

$$\begin{bmatrix} \mathbf{X'R^{-1}X} & \mathbf{X'R^{-1}W} \\ \mathbf{W'R^{-1}X} & \mathbf{W'R^{-1}W + A^{-1} \otimes G_0^{-1}} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X'R^{-1}y} \\ \mathbf{W'R^{-1}y} \end{bmatrix}$$

blupf90+

MME Solver

Default

VC Estimation

- AI-REML:

```
OPTION method VCE
```

- EM-REML:

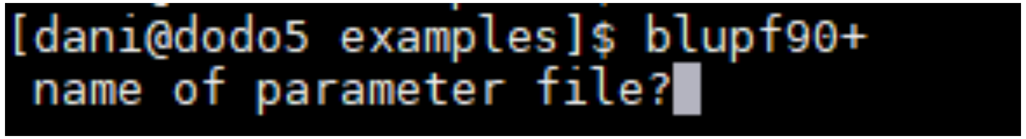
```
OPTION method VCE

OPTION EM-REML
```

blupf90+

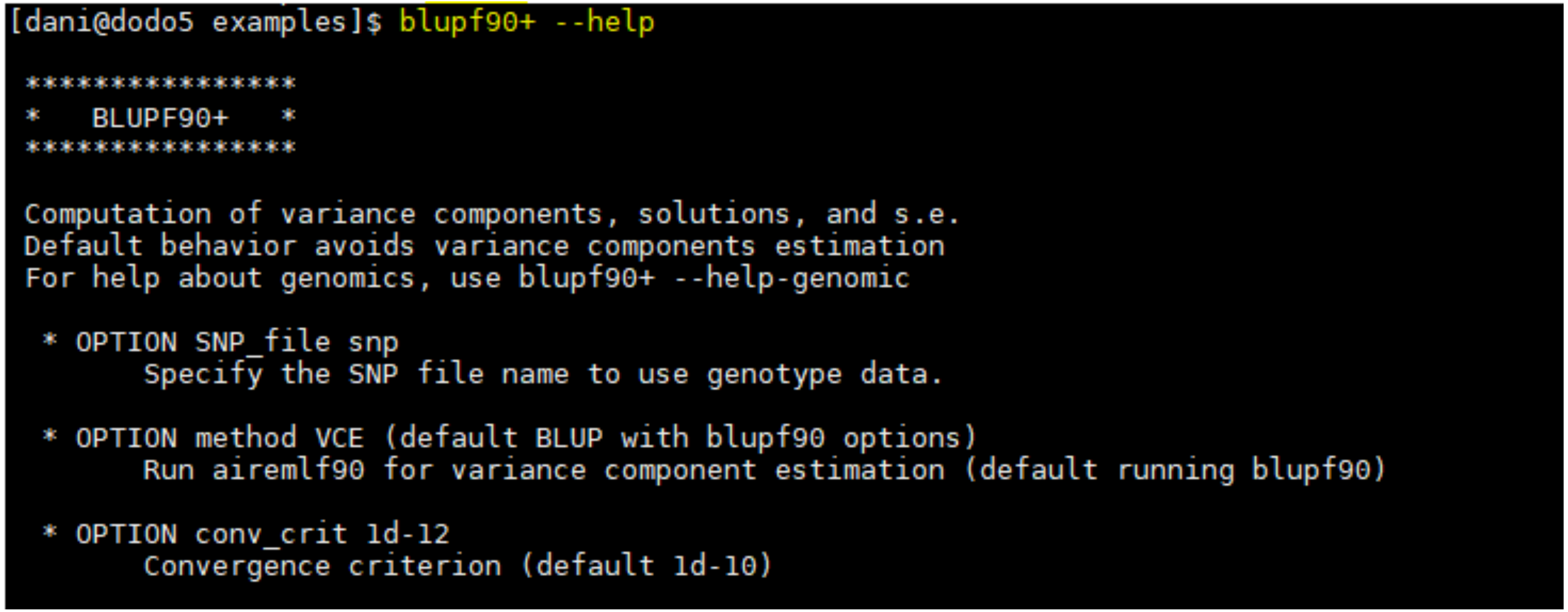
- Supports virtually any model used in AB&G:
  - animal model
  - models with maternal effect
  - MPE
  - PE
  - Random Regression
  - Social interaction
  - Multiple traits
    - up to 70 if no correlated effects
    - up to [70/number of correlated effects]

blupf90+

- How to use:



```
[dani@dodo5 examples]$ blupf90+
 name of parameter file?
```



```
[dani@dodo5 examples]$ blupf90+ --help

 ****************
 *   BLUPF90+   *
 ****************

 Computation of variance components, solutions, and s.e.
 Default behavior avoids variance components estimation
 For help about genomics, use blupf90+ --help-genomic

  * OPTION SNP_file snp
        Specify the SNP file name to use genotype data.

  * OPTION method VCE (default BLUP with blupf90 options)
        Run airemlf90 for variance component estimation (default running blupf90)

  * OPTION conv_crit 1d-12
        Convergence criterion (default 1d-10)
```

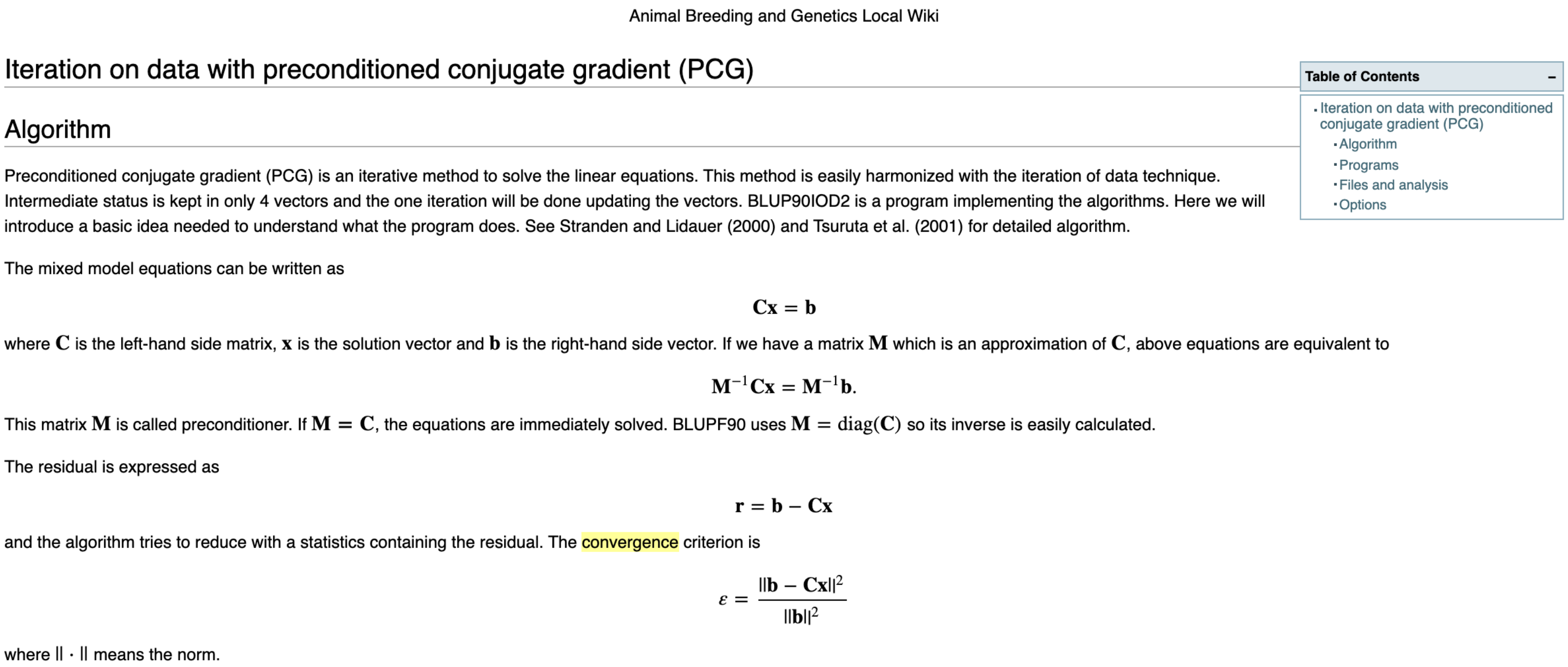
blupf90+

- Input files
  - Free format (minimum one space to separate columns)
  - TAB is not a valid separator
  - Only numbers: integer or real
  - Decimal separators “ . ” not “ , ”
  - One “ . ” is not a missing value as in SAS
  - All effects need to be renumbered from 1 (consecutively)

blupf90+

- Computes generalized solutions by several methods:
  - Preconditioner Conjugate Gradient (PCG)
    - Default Iterative method (fast)
  - Successive over-relaxation (SOR)
    - an iterative method based on Gauss-Seidel
  - Direct solution using sparse Cholesky factorization
    - FSPAK or YAMS (greater memory requirements)
- The solution values change among methods, but estimable functions are the same
- Prediction error variances can be obtained using sparse inverse (FSPAK or YAMS)

blupf90+ with PCG



Animal Breeding and Genetics Local Wiki

Iteration on data with preconditioned conjugate gradient (PCG)

Table of Contents –

Algorithm

Preconditioned conjugate gradient (PCG) is an iterative method to solve the linear equations. This method is easily harmonized with the iteration of data technique. Intermediate status is kept in only 4 vectors and the one iteration will be done updating the vectors. BLUP90IOD2 is a program implementing the algorithms. Here we will introduce a basic idea needed to understand what the program does. See Stranden and Lidauer (2000) and Tsuruta et al. (2001) for detailed algorithm.

The mixed model equations can be written as

$$\mathbf{Cx} = \mathbf{b}$$

where $\mathbf{C}$ is the left-hand side matrix, $\mathbf{x}$ is the solution vector and $\mathbf{b}$ is the right-hand side vector. If we have a matrix $\mathbf{M}$ which is an approximation of $\mathbf{C}$, above equations are equivalent to

$$\mathbf{M}^{-1}\mathbf{Cx} = \mathbf{M}^{-1}\mathbf{b}.$$

This matrix $\mathbf{M}$ is called preconditioner. If $\mathbf{M} = \mathbf{C}$, the equations are immediately solved. BLUPF90 uses $\mathbf{M} = \mathrm{diag}(\mathbf{C})$ so its inverse is easily calculated.

The residual is expressed as

$$\mathbf{r} = \mathbf{b} - \mathbf{Cx}$$

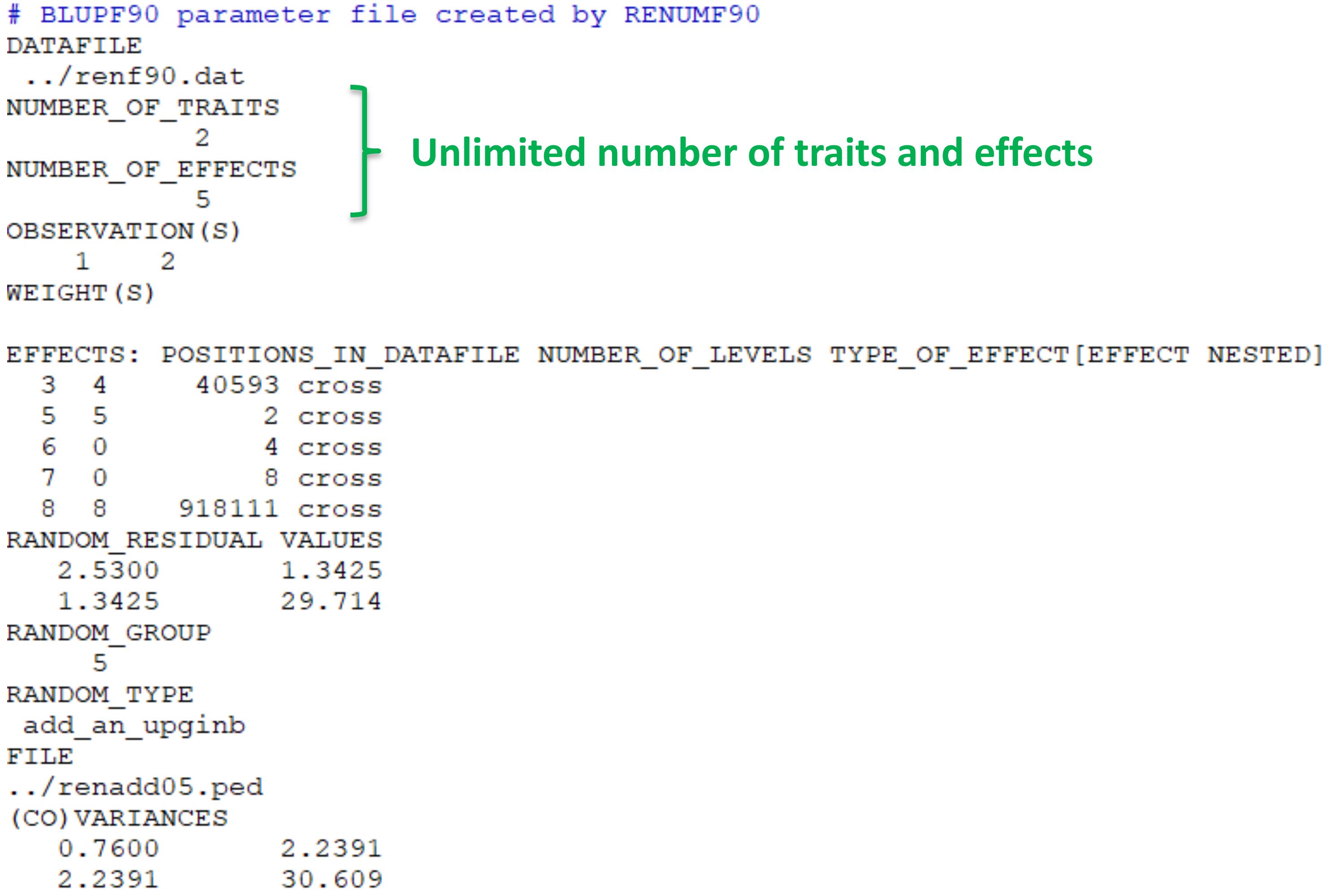
and the algorithm tries to reduce with a statistics containing the residual. The convergence criterion is

$$\varepsilon = \frac{||\mathbf{b} - \mathbf{Cx}||^2}{||\mathbf{b}||^2}$$

where $||\cdot||$ means the norm.

If $\mathbf{M}^{-1}\mathbf{C}$ has a better condition than **C**, the convergence is reached is faster

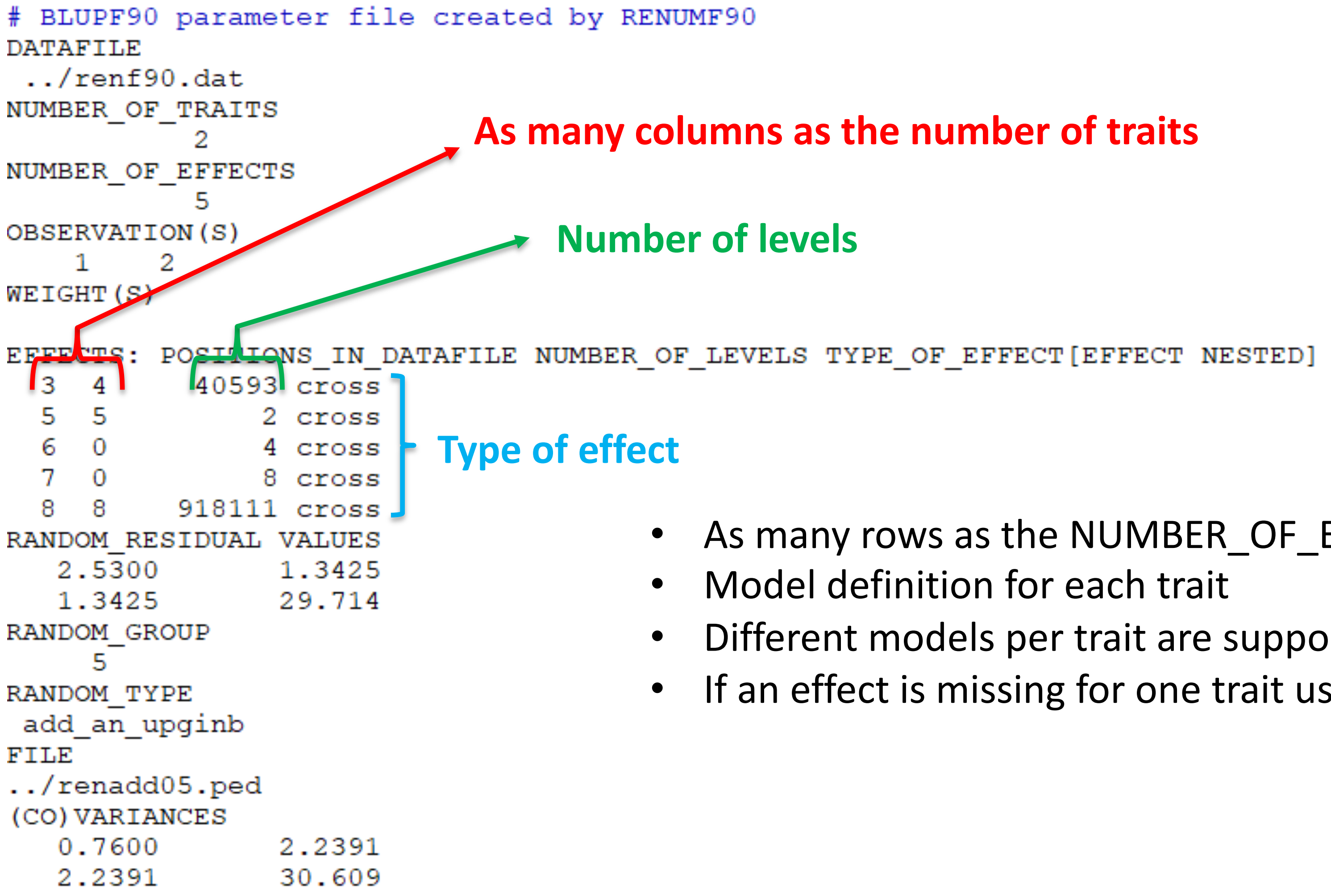
Parameter file for blupf90+



```
# BLUPF90 parameter file created by RENUMF90
DATAFILE
 ../renf90.dat
NUMBER_OF_TRAITS
           2
NUMBER_OF_EFFECTS
           5
OBSERVATION(S)
    1    2
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
  3  4      40593 cross
  5  5          2 cross
  6  0          4 cross
  7  0          8 cross
  8  8     918111 cross
RANDOM_RESIDUAL VALUES
   2.5300        1.3425
   1.3425        29.714
RANDOM_GROUP
     5
RANDOM_TYPE
 add_an_upginb
FILE
../renadd05.ped
(CO)VARIANCES
   0.7600        2.2391
   2.2391        30.609
```

Unlimited number of traits and effects

Parameter file for blupf90+



```
# BLUPF90 parameter file created by RENUMF90
DATAFILE
 ../renf90.dat
NUMBER_OF_TRAITS
           2
NUMBER_OF_EFFECTS
           5
OBSERVATION(S)
    1    2
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
  3  4      40593 cross
  5  5          2 cross
  6  0          4 cross
  7  0          8 cross
  8  8     918111 cross
RANDOM_RESIDUAL VALUES
   2.5300         1.3425
   1.3425         29.714
RANDOM_GROUP
     5
RANDOM_TYPE
 add_an_upginb
FILE
../renadd05.ped
(CO)VARIANCES
   0.7600         2.2391
   2.2391         30.609
```

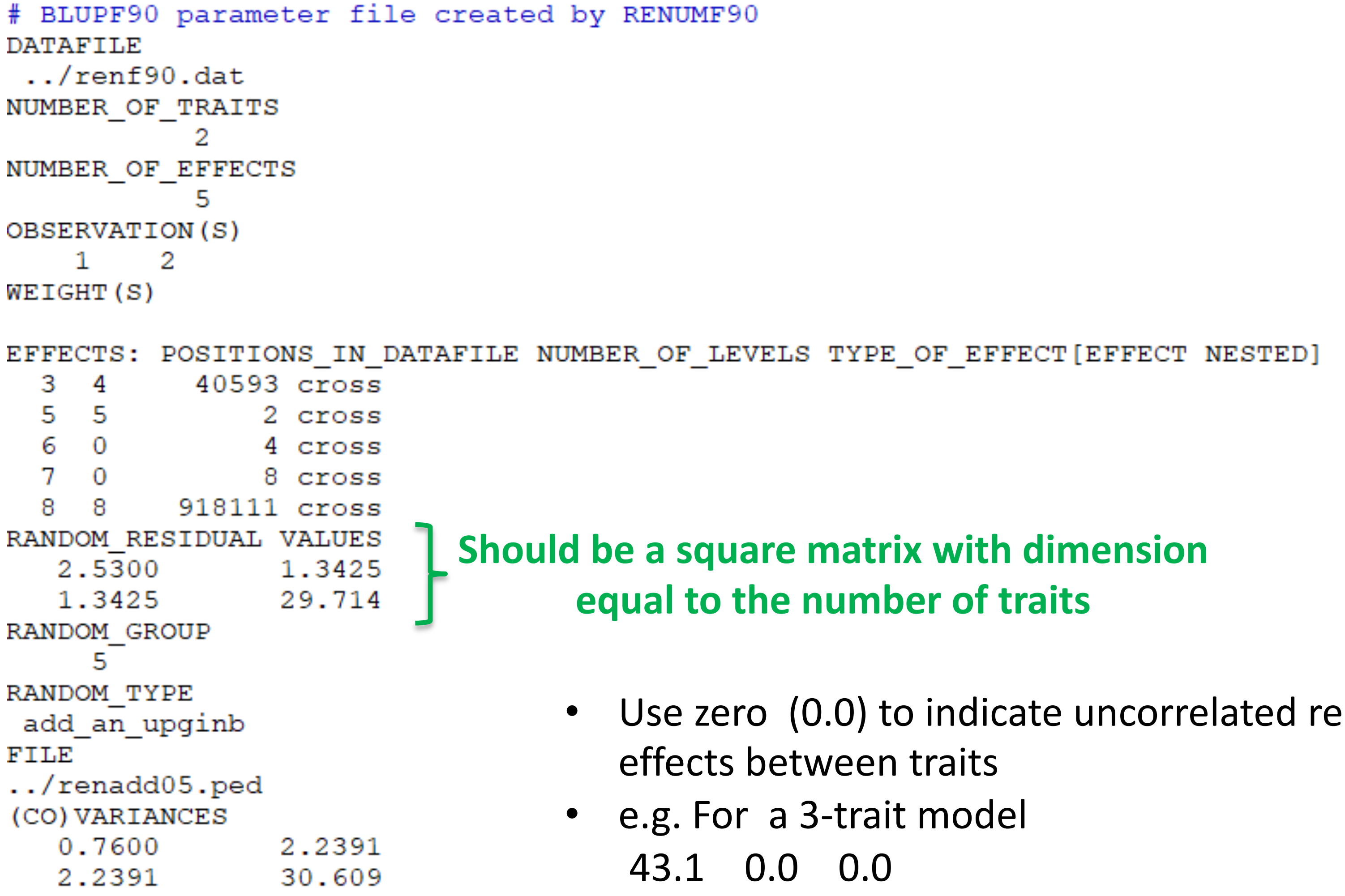
As many columns as the number of traits

Number of levels

Type of effect

- As many rows as the NUMBER_OF_EFFECTS
- Model definition for each trait
- Different models per trait are supported
- If an effect is missing for one trait use 0

Parameter file for blupf90+



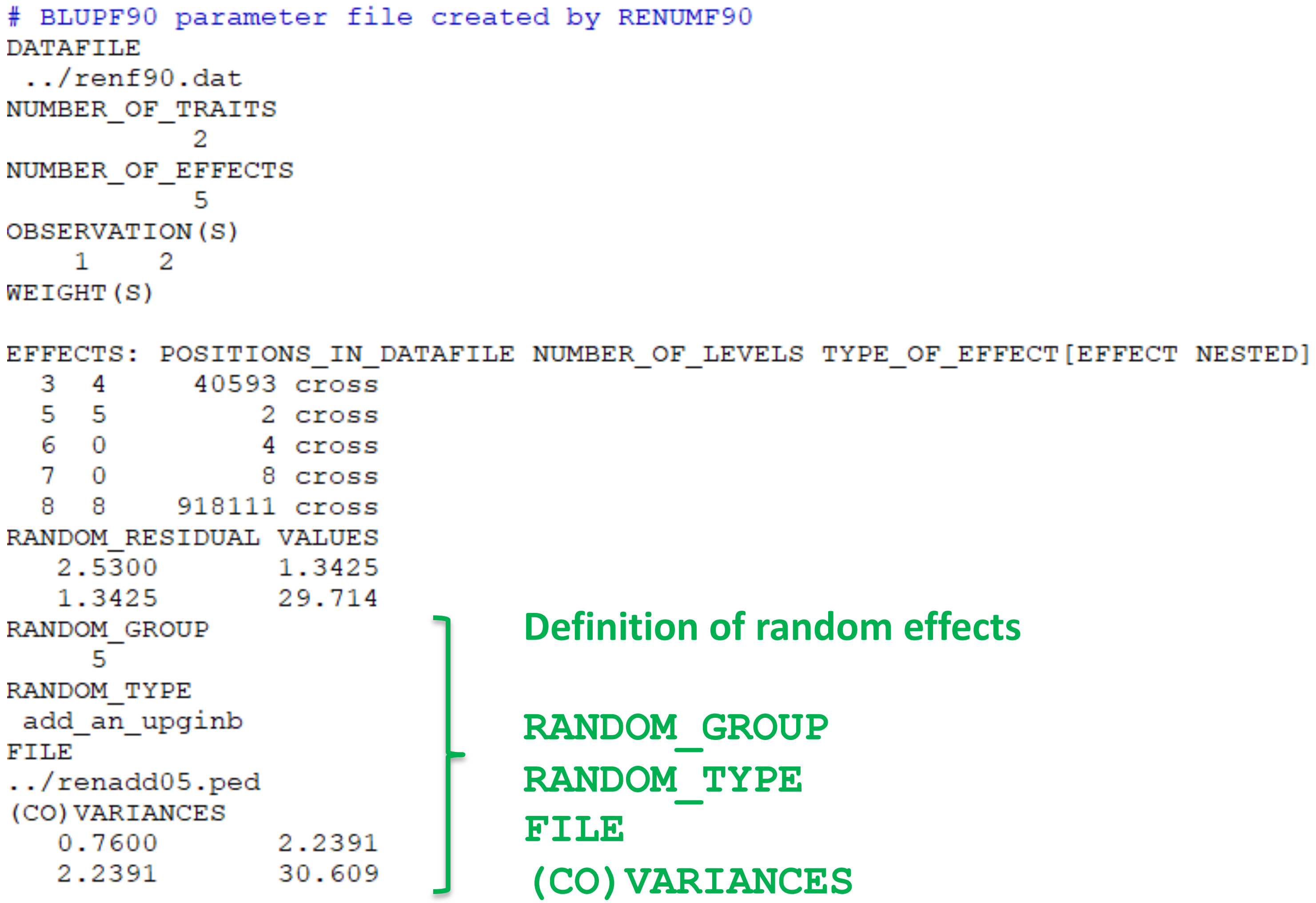
```
# BLUPF90 parameter file created by RENUMF90
DATAFILE
 ../renf90.dat
NUMBER_OF_TRAITS
           2
NUMBER_OF_EFFECTS
           5
OBSERVATION(S)
    1    2
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
  3  4      40593 cross
  5  5          2 cross
  6  0          4 cross
  7  0          8 cross
  8  8     918111 cross
RANDOM_RESIDUAL VALUES
   2.5300        1.3425
   1.3425        29.714
RANDOM_GROUP
     5
RANDOM_TYPE
 add_an_upginb
FILE
../renadd05.ped
(CO)VARIANCES
   0.7600        2.2391
   2.2391        30.609
```

Should be a square matrix with dimension equal to the number of traits

- Use zero (0.0) to indicate uncorrelated residual effects between traits
- e.g. For a 3-trait model

$$\begin{matrix} 43.1 & 0.0 & 0.0 \\ 0.0 & 5.1 & 3.2 \\ 0.0 & 3.2 & 10.3 \end{matrix}$$

Parameter file for blupf90+



```
# BLUPF90 parameter file created by RENUMF90
DATAFILE
 ../renf90.dat
NUMBER_OF_TRAITS
           2
NUMBER_OF_EFFECTS
           5
OBSERVATION(S)
    1    2
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
  3  4     40593 cross
  5  5         2 cross
  6  0         4 cross
  7  0         8 cross
  8  8    918111 cross
RANDOM_RESIDUAL VALUES
   2.5300        1.3425
   1.3425        29.714
RANDOM_GROUP
     5
RANDOM_TYPE
 add_an_upginb
FILE
../renadd05.ped
(CO)VARIANCES
   0.7600        2.2391
   2.2391        30.609
```

Definition of random effects

RANDOM_GROUP
RANDOM_TYPE
FILE
(CO)VARIANCES

Definition of random effects

- RANDOM_GROUP
  - Number of the effect(s) from list of effects
  - Correlated effects should be consecutive e.g., Maternal effects, Random Regression

- RANDOM_TYPE
  - diagonal, add_animal, add_sire, add_an_upg, add_an_upginb, add_an_meta, add_an_self, user_file, user_file_i, or par_domin

- FILE
  - Pedigree file, parental dominance, or user file

- (CO)VARIANCES
  - Square matrix with dimension equal to the number_of_traits*number_of_correlated_effects

(CO)VARIANCES

- Assuming 3 traits (T1-T3)

	T1	T2	T3
T1			
T2			
T3			

(CO)VARIANCES

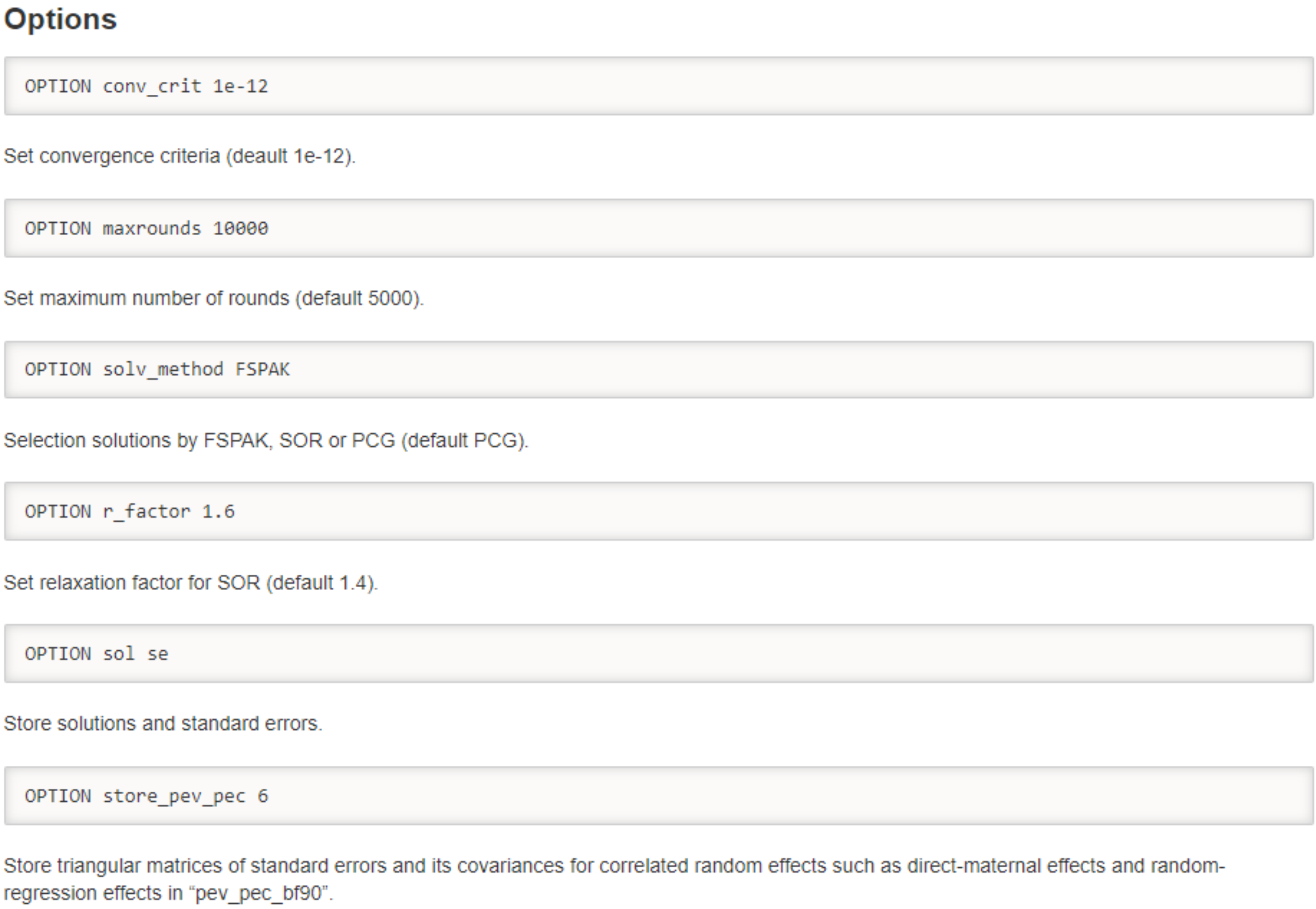
- Assuming 3 traits (T1-T3) and 2 correlated effects (E1-E2)

		Direct			Maternal		
		T1	T2	T3	T1	T2	T3
Direct	T1						
	T2						
	T3						
Maternal	T1						
	T2						
	T3						

OPTIONS for blupf90+

- Program behavior is modified by adding extra options at the end of the par file
- `OPTION option_name x1 x2 …`
- `option_name`: each program has its own options
- `x1 x2:` each option has its own parameters

Options for blupf90+



Options

```
OPTION conv_crit 1e-12
```

Set convergence criteria (deault 1e-12).

```
OPTION maxrounds 10000
```

Set maximum number of rounds (default 5000).

```
OPTION solv_method FSPAK
```

Selection solutions by FSPAK, SOR or PCG (default PCG).

```
OPTION r_factor 1.6
```

Set relaxation factor for SOR (default 1.4).

```
OPTION sol se
```

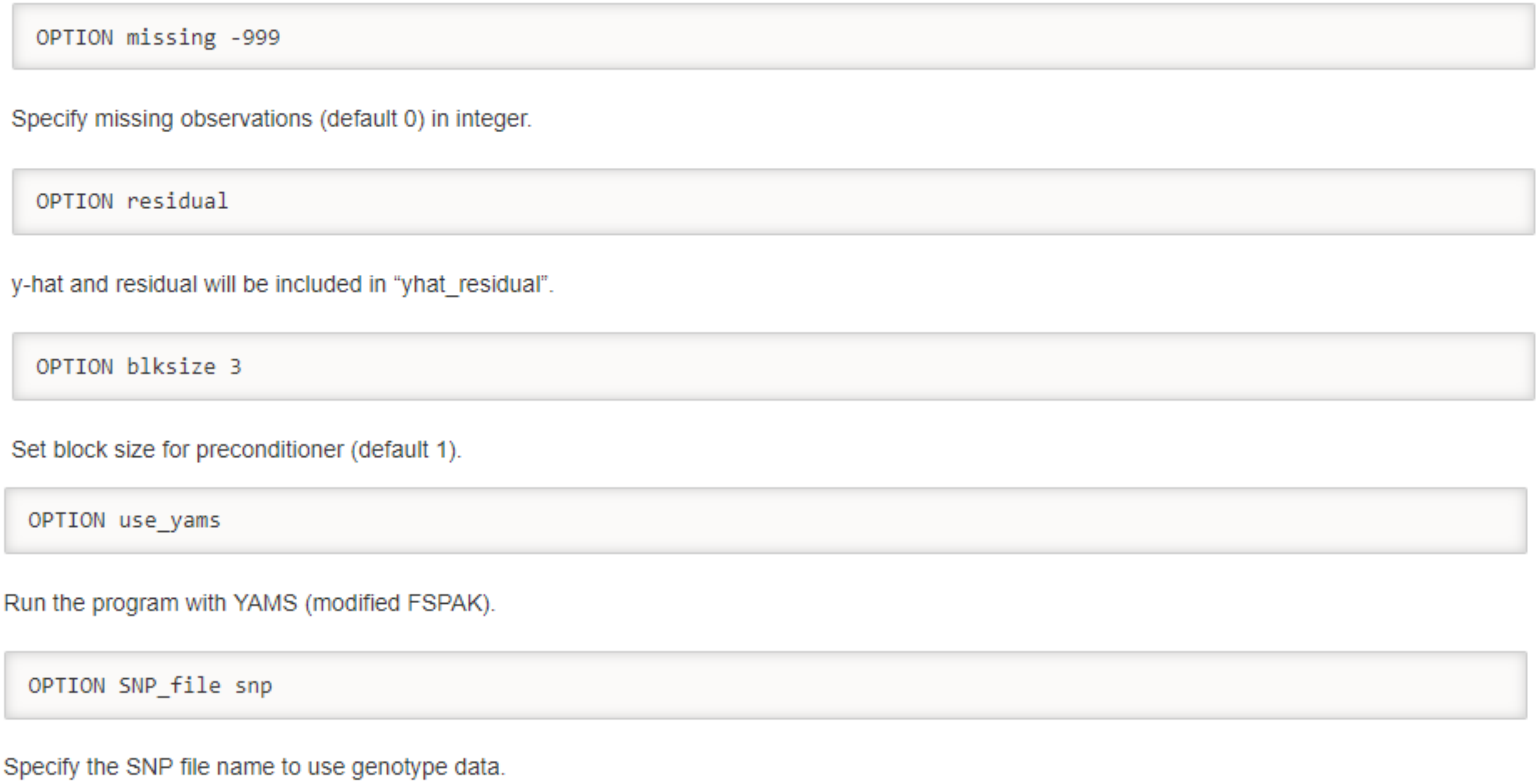
Store solutions and standard errors.

```
OPTION store_pev_pec 6
```

Store triangular matrices of standard errors and its covariances for correlated random effects such as direct-maternal effects and random-regression effects in "pev_pec_bf90".

Options for blupf90+

Missing data
Not pedigree!



```
OPTION missing -999
```

Specify missing observations (default 0) in integer.

```
OPTION residual
```

y-hat and residual will be included in “yhat_residual”.

```
OPTION blksize 3
```

Set block size for preconditioner (default 1).

```
OPTION use_yams
```

Run the program with YAMS (modified FSPAK).

```
OPTION SNP_file snp
```

Specify the SNP file name to use genotype data.

New options for blupf90+

- Storing reliabilities based on PEV

```
OPTION store_accuracy X
```

Number of animal effect

$$Rel = 1 - \frac{PEV}{\sigma_u^2(1+f)}$$

  - Adjusts for f (inbreeding) from **A**, **G**, or **H**
  - Turn inbreeding adjustment off
  - `OPTION correct_accuracy_by_inbreeding_direct 0`

- Storing solutions with original ID if renumf90 was used to renumber the data

```
OPTION origID
```

May not work with some programs and options

  - Only *solutions.original* is created

New options for blupf90+

- Storing reliabilities with original ID

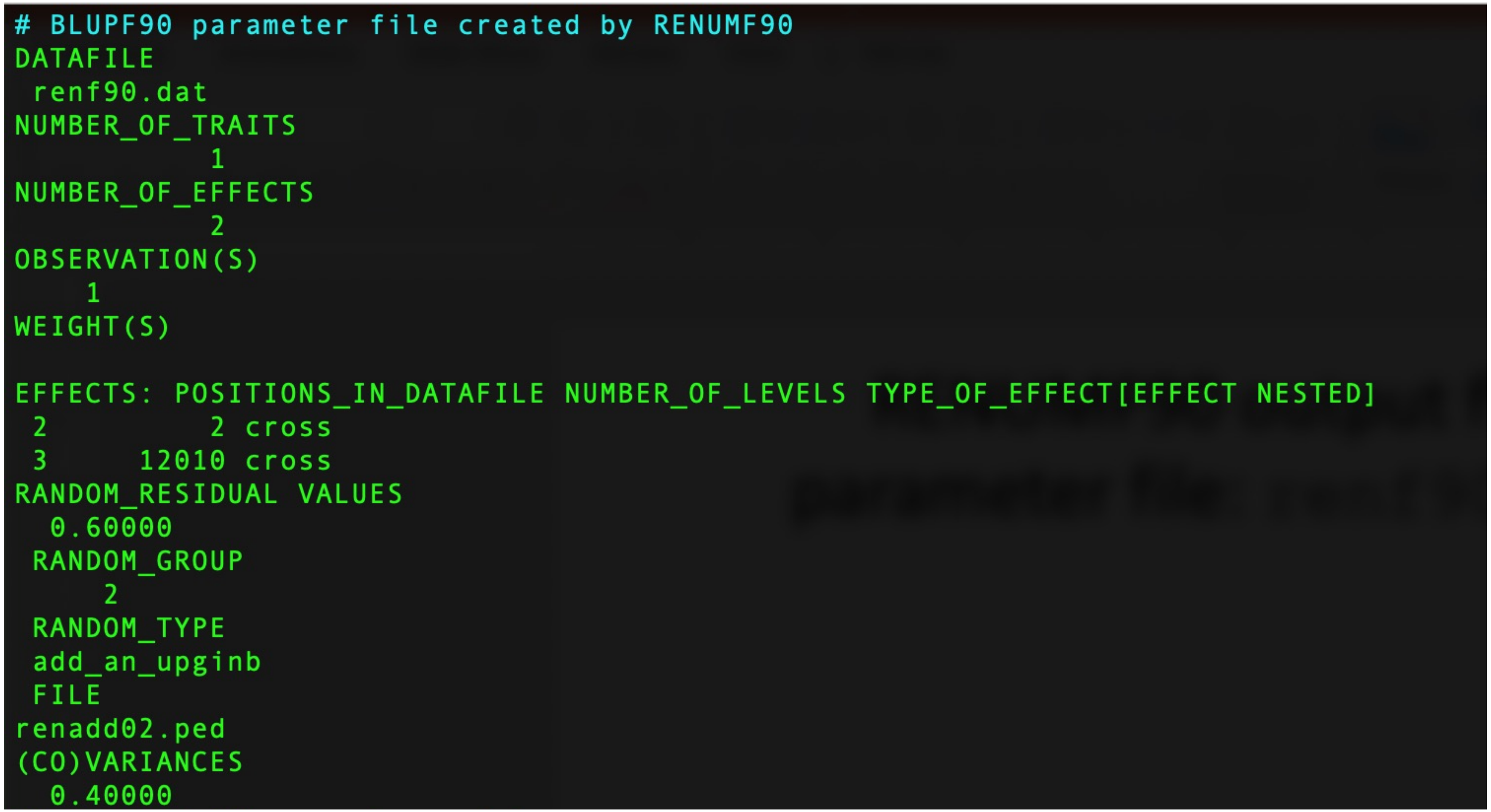
```
OPTION store_accuracy X orig
```

Number of animal effect

$$Rel = 1 - \frac{PEV}{\sigma_u^2(1+f)}$$

- Storing solutions and rel with original ID if renumf90 was used to renumber the data
  - The option will save *acc_bf90* with renumbered and original ID
  - If want to have *solutions.original* as well, combine with OPTION origID

Common parameter file for blupf90+



```
# BLUPF90 parameter file created by RENUMF90
DATAFILE
 renf90.dat
NUMBER_OF_TRAITS
           1
NUMBER_OF_EFFECTS
           2
OBSERVATION(S)
    1
WEIGHT(S)

EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
 2         2 cross
 3     12010 cross
RANDOM_RESIDUAL VALUES
  0.60000
 RANDOM_GROUP
     2
 RANDOM_TYPE
 add_an_upginb
 FILE
renadd02.ped
(CO)VARIANCES
  0.40000
```

Example

Model: $y = farm + sex + \beta\ age + animal + e$

```
DATAFILE
 data1.txt
NUMBER_OF_TRAITS
           1
NUMBER_OF_EFFECTS
           4
OBSERVATION(S)
    1
WEIGHT(S)


EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT[EFFECT NESTED]
 2         3 cross
 3         2 cross
 4 1 cov
 5        15 cross
RANDOM_RESIDUAL VALUES
   1.0000
 RANDOM_GROUP
     4
 RANDOM_TYPE
 add_an_upginb
 FILE
ped1.txt
(CO)VARIANCES
  0.20000
```

ped1.txt

Anim	Sire	Dam
1	15	14
14	0	0
2	15	14
11	0	0
3	15	11
4	15	11
5	12	13
12	0	0
6	12	11
7	15	13
13	0	0
8	12	13
9	12	14
15	0	0
10	12	11

data1.txt

phen	farm	sex	age	Anim
3	1	1	1	6
2	1	2	1	8
4	1	1	2	9
6	2	2	2	10
3	2	1	1	5
6	2	2	2	1
6	3	1	2	3
6	3	2	1	7
8	3	1	1	2
4	3	2	2	4

blupf90+

VC Estimation

EM-REML: expectation-maximization (EM) algorithm

AI-REML: average information (AI) algorithm

REML

- REML = restricted/residual maximum likelihood
  - Patterson and Thompson (1971)
- Most used method for VCE in AB&G

EM-REML

- This method requires iterations:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \mathbf{e}$$

$$\begin{bmatrix} \mathbf{X'X} & \mathbf{X'Z} \\ \mathbf{Z'X} & \mathbf{Z'Z} + \mathbf{A}^{-1}\dfrac{\sigma_e^2}{\sigma_a^2} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X'y} \\ \mathbf{Z'y} \end{bmatrix}$$

1) set initial variance components

2) compute $\hat{\boldsymbol{\beta}}$ and $\hat{\mathbf{u}}$ via mixed model equations

3) update variance components with the following equations

$$\hat{\sigma}_a^2 = \frac{\hat{\mathbf{u}}'\mathbf{A}^{-1}\hat{\mathbf{u}} + \mathrm{tr}\left(\mathbf{A}^{-1}\mathbf{C}^{uu}\right)}{N_a}$$

$$\hat{\sigma}_e^2 = \frac{\mathbf{y}'(\mathbf{y} - \mathbf{X}\hat{\beta} - \mathbf{Z}\hat{\mathbf{u}})}{N - \mathrm{rank}(\mathbf{X})}$$

Inverse of LHS for animal effect

\# animals (rank of A)

4) go to 1 or stop if the parameters do not change anymore

EM-REML

- Simpler equations
  - More complicated equations in multiple-trait models
- Easier to understand
- Very slow convergence (looks stable but may not converge)
- Computationally demanding, especially for $\mathbf{C^{uu}}$

$$\begin{bmatrix}\hat{\boldsymbol{\beta}}\\ \hat{\mathbf{u}}\end{bmatrix} = \begin{bmatrix}\mathbf{X'X} & \mathbf{X'Z}\\ \mathbf{Z'X} & \mathbf{Z'Z+A^{-1}}\frac{\boldsymbol{\sigma_e^2}}{\boldsymbol{\sigma_a^2}}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{X'}\mathrm{y}\\ \mathbf{Z'}\mathrm{y}\end{bmatrix}$$

AI-REML

Vector of variance components

$$\theta_{n+1} = \theta_n - \mathbf{H}^{-1}(\theta_n)\mathbf{d}(\theta_n)$$

Gradient (score vector)

Hessian Matrix

P = Projection
or hat matrix

Average-information algorithm uses this matrix as Hessian,

$$\mathbf{H}(\theta) = \mathcal{I}_A(\theta) = \begin{bmatrix} -\frac{1}{2}\mathbf{y}'\mathbf{PZAZ}'\mathbf{PZAZ}'\mathbf{Py} & -\frac{1}{2}\mathbf{y}'\mathbf{PZAZ}'\mathbf{PPy} \\ -\frac{1}{2}\mathbf{y}'\mathbf{PPZAZ}'\mathbf{Py} & -\frac{1}{2}\mathbf{y}'\mathbf{PPPy} \end{bmatrix}$$

expensive

Gradient

$$-2\mathrm{d}(\theta) = \begin{bmatrix} \mathrm{tr}(\mathbf{PZAZ}') - \mathbf{y}'\mathbf{PZAZ}'\mathbf{Py} \\ \mathrm{tr}(\mathbf{P}) - \mathbf{y}'\mathbf{PPy} \end{bmatrix} = \begin{bmatrix} \frac{N_a}{\sigma_a^2} - \frac{\mathrm{tr}(\mathbf{A}^{-1}\mathbf{C}^{uu})}{(\sigma_a^2)^2} - \frac{\hat{\mathbf{u}}'\mathbf{A}^{-1}\hat{\mathbf{u}}}{(\sigma_a^2)^2} \\ \frac{N - \mathrm{rank}(\mathbf{X})}{\sigma_e^2} - \frac{1}{\sigma_e^2}\left[N_a - \frac{\mathrm{tr}(\mathbf{A}^{-1}\mathbf{C}^{uu})}{\sigma_a^2}\right] - \frac{\hat{\mathbf{e}}'\hat{\mathbf{e}}}{(\sigma_e^2)^2} \end{bmatrix}$$

AI-REML

- Computationally demanding
- Much faster than EM-REML
  - Fewer iterations
- Provides estimation of standard errors
- BUT
  - For complex models and poor starting values
    - Slow convergence
    - Parameter estimates out of the parameter space
  - In some cases, initial rounds with EM-REML may help

blupf90+

VC Estimation

- AI-REML:

```
OPTION method VCE
```

- EM-REML:

```
OPTION method VCE

OPTION EM-REML
```

Original options for airemlf90 and remlf90 also work!

OPTION EM-REML

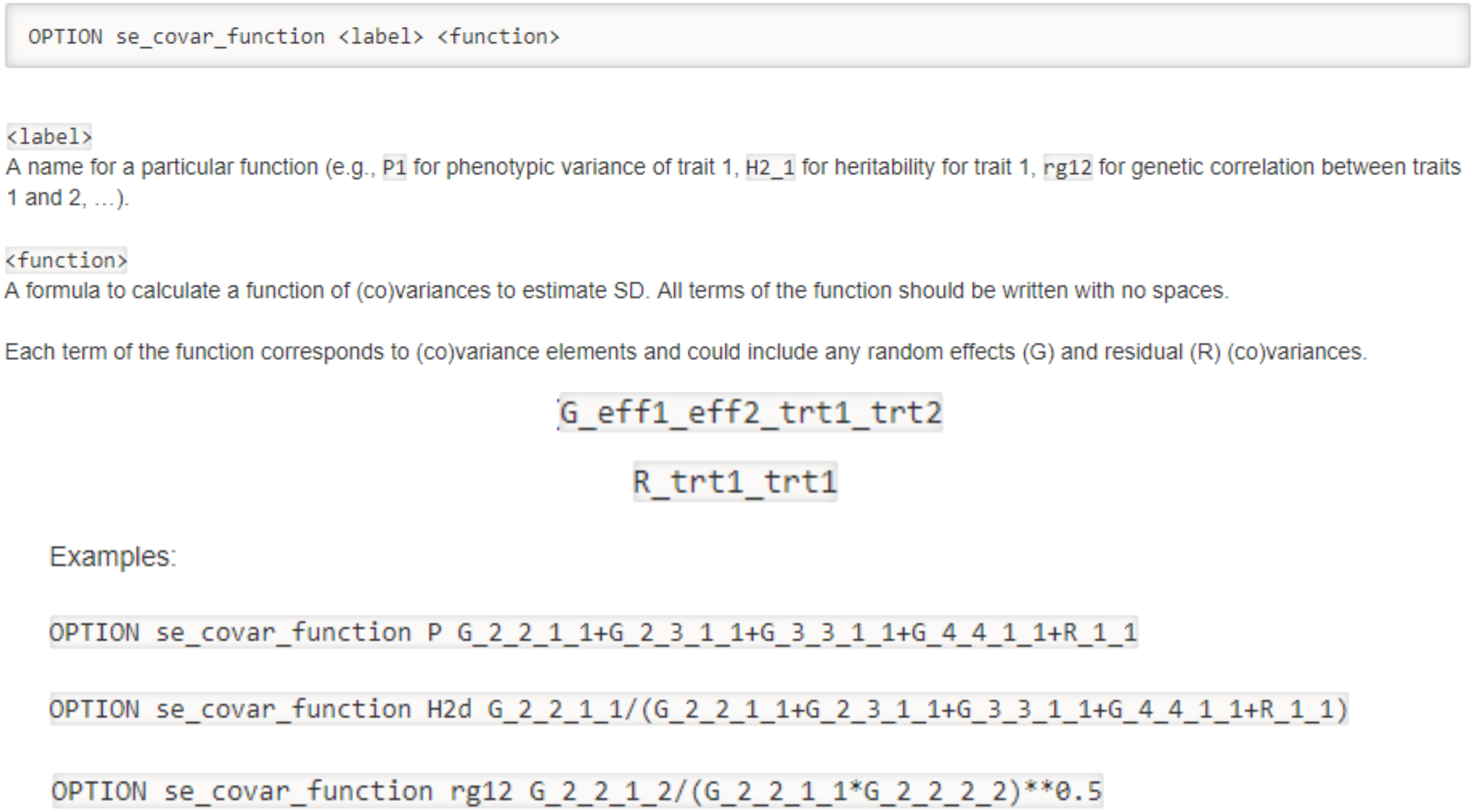
`OPTION method VCE`

`OPTION EM-REML`
`OPTION EM-REML pure` → Runs EM until convergence | shows EM output = remlf90

`OPTION EM-REML n` → Runs `n` EM rounds | switches to AI| shows AI output = airemlf90

`OPTION EM-REML ai` → Runs EM until convergence| switches to AI = airemlf90

Options for blupf90+



```
OPTION se_covar_function <label> <function>
```

`<label>`
A name for a particular function (e.g., `P1` for phenotypic variance of trait 1, `H2_1` for heritability for trait 1, `rg12` for genetic correlation between traits 1 and 2, …).

`<function>`
A formula to calculate a function of (co)variances to estimate SD. All terms of the function should be written with no spaces.

Each term of the function corresponds to (co)variance elements and could include any random effects (G) and residual (R) (co)variances.

`G_eff1_eff2_trt1_trt2`

`R_trt1_trt1`

Examples:

```
OPTION se_covar_function P G_2_2_1_1+G_2_3_1_1+G_3_3_1_1+G_4_4_1_1+R_1_1
```

```
OPTION se_covar_function H2d G_2_2_1_1/(G_2_2_1_1+G_2_3_1_1+G_3_3_1_1+G_4_4_1_1+R_1_1)
```

```
OPTION se_covar_function rg12 G_2_2_1_2/(G_2_2_1_1*G_2_2_2_2)**0.5
```

SE for genetic parameters

```
#genetic, permanent, residual
ahat=c(
  0.11478,
  0.13552,
  0.25290,
  )
```

with AI matrix:

```
# inverse of AI matrix (Sampling Variance)
AI=matrix(c(
  0.16799E-05, -0.96486E-06, -0.82566E-08,
 -0.96486E-06,  0.96167E-06, -0.37113E-07,
 -0.82566E-08, -0.37113E-07,  0.10864E-06)
,ncol=3)
```

```
require(MASS)
b=mvrnorm(10000,ahat,AI)
> head(b)
          [,1]      [,2]      [,3]
[1,] 0.1146738 0.1357640 0.2529399
[2,] 0.1163889 0.1342926 0.2528479
[3,] 0.1166155 0.1344342 0.2525161
[4,] 0.1142085 0.1358928 0.2534974
[5,] 0.1136835 0.1361108 0.2530133
[6,] 0.1140485 0.1365707 0.2530573
```

heritability and its standard deviation:

```
h2=b[,1]/(b[,1]+b[,2]+b[,3])
sd(h2)
> 0.002318198
```

Houle and Meyer (2015)

http://artadia.blogspot.com/2016/05/standard-error-of-variance-components.html

SE for genetic parameters

Houle and Meyer (2015):

*Large-sample theory shows that maximum-likelihood estimates (including restricted maximum likelihood, REML) asymptotically have a multivariate normal distribution, with covariance matrix derived from the inverse of the information matrix, and mean equal to the estimated **G**. This suggests that sampling estimates of **G** from this distribution can be used to assess the variability of estimates of **G**, and of functions of **G**.*

G = additive genetic variance–covariance matrices

Does blupf90+ for VCE always converge?

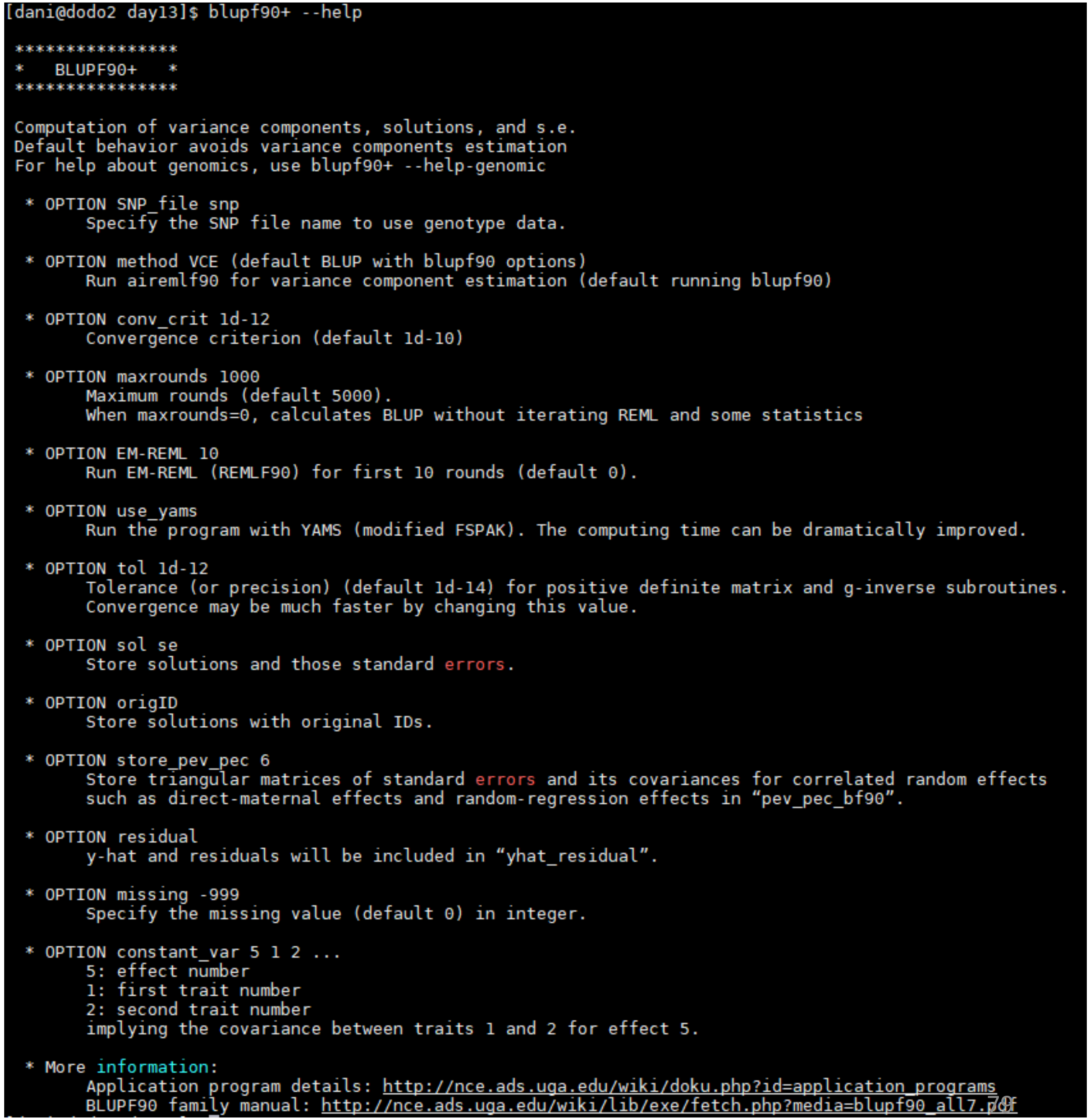
- When the expected variance is very small, or the covariance matrix is close to non-positive definite, try the following starting values:
  - much smaller = 0.00001
  - much bigger = 1000

- If blupf90+ does not converge with AI-REML but converges with EM-REML with the same data set and the same model:
  - run EM-REML again but with a smaller starting value to check the estimate because it could be an artifact

  - use OPTION EM-REML inside blupf90+ as an initial point for AI-REML:
    - `OPTION EM-REML xx`

blupf90+ quick trick

- `blupf90+ --help`



```
[dani@dodo2 day13]$ blupf90+ --help

 ****************
 *   BLUPF90+   *
 ****************

 Computation of variance components, solutions, and s.e.
 Default behavior avoids variance components estimation
 For help about genomics, use blupf90+ --help-genomic

  * OPTION SNP_file snp
       Specify the SNP file name to use genotype data.

  * OPTION method VCE (default BLUP with blupf90 options)
       Run airemlf90 for variance component estimation (default running blupf90)

  * OPTION conv_crit 1d-12
       Convergence criterion (default 1d-10)

  * OPTION maxrounds 1000
       Maximum rounds (default 5000).
       When maxrounds=0, calculates BLUP without iterating REML and some statistics

  * OPTION EM-REML 10
       Run EM-REML (REMLF90) for first 10 rounds (default 0).

  * OPTION use_yams
       Run the program with YAMS (modified FSPAK). The computing time can be dramatically improved.

  * OPTION tol 1d-12
       Tolerance (or precision) (default 1d-14) for positive definite matrix and g-inverse subroutines.
       Convergence may be much faster by changing this value.

  * OPTION sol se
       Store solutions and those standard errors.

  * OPTION origID
       Store solutions with original IDs.

  * OPTION store_pev_pec 6
       Store triangular matrices of standard errors and its covariances for correlated random effects
       such as direct-maternal effects and random-regression effects in “pev_pec_bf90”.

  * OPTION residual
       y-hat and residuals will be included in “yhat_residual”.

  * OPTION missing -999
       Specify the missing value (default 0) in integer.

  * OPTION constant_var 5 1 2 ...
       5: effect number
       1: first trait number
       2: second trait number
       implying the covariance between traits 1 and 2 for effect 5.

  * More information:
       Application program details: http://nce.ads.uga.edu/wiki/doku.php?id=application_programs
       BLUPF90 family manual: http://nce.ads.uga.edu/wiki/lib/exe/fetch.php?media=blupf90_all7.pdf
```

`gibbsf90+`

- `gibbs1f90`: stores single trait matrices once – fast for multi-trait models
- `gibbs2f90`: gibbs1f90 with joint sampling of correlated effects – Maternal effects and RRM
- `gibbs3f90`: gibbs2f90 with heterogeneous residual variance
- `thrgibbs1f90`: for linear-threshold models
- `thrgibbs3f90`: thrgibbs1f90 with heterogeneous residual variance

Variance Components Estimation

Mixed Model Equations Solver

$$\begin{bmatrix} \mathbf{X'R^{-1}X} & \mathbf{X'R^{-1}W} \\ \mathbf{W'R^{-1}X} & \mathbf{W'R^{-1}W+A^{-1}\otimes G_0^{-1}} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X'R^{-1}y} \\ \mathbf{W'R^{-1}y} \end{bmatrix}$$

`gibbsf90+`

Linear

Default

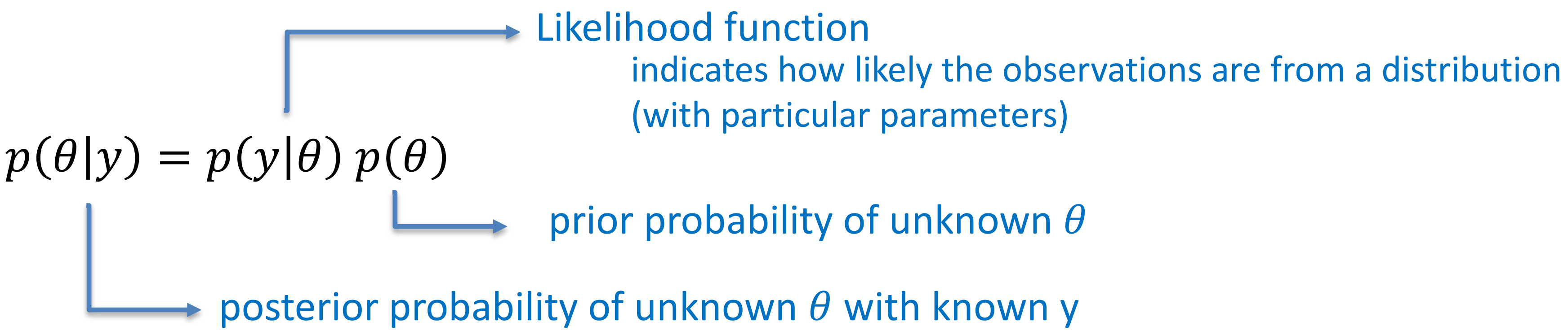
Threshold (-Linear)

```
OPTION cat 0 2 5
```

- Categories renumbered from **1**
- Missing records is only **0**

gibbsf90+

Bayes Theorem



Likelihood function

indicates how likely the observations are from a distribution (with particular parameters)

$$p(\theta|y) = p(y|\theta)\, p(\theta)$$

prior probability of unknown θ

posterior probability of unknown θ with known y

- Basic idea of Gibbs Sampling:
- Numerical method to draw samples from a posterior distribution (not always explicitly available)
- Draw samples = generate random numbers following a distribution
- The results are random numbers (not theoretical formulas)
- The posterior distribution will be drawn based on the numerical values (like a histogram)

gibbsf90+

Ingredients for Gibbs sampling

1) Theoretical derivation: conditional posterior distribution for each unknown parameter

2) Software: a random number generator for a particular distribution

```
# Basic Gibbs sampling for mu (normal) and sigma2 (inverted chi-square)
y <- c(14,16,18)
N <- length(y)
n.samples <- 100
mu <- rep(0,n.samples)
sigma2 <- rep(0,n.samples)

# initial value
mu[1] <- 0
sigma2[1] <- 10

# sampling
for(i in 2:n.samples){
    mu[i] <- rnorm(1, mean=mean(y), sd=sqrt(sigma2[i-1]/N)) # using the most recent sigma2
    df <- N-2
    S <- sum((y-mu[i])^2)
    sigma2[i] <- rinvchisq(1, df=df, scale=S) # using the most recent mu
}
```

gibbsf90+

- Name of parameter file?

  gibbs1.par
- Number of samples and length of burn-in?

  samples=10,000 to 100,000; burn-in=0
- Give n to store every n-th sample?

  10

- `gibbsf90+ parfile.par --samples i --burnin j --interval k`

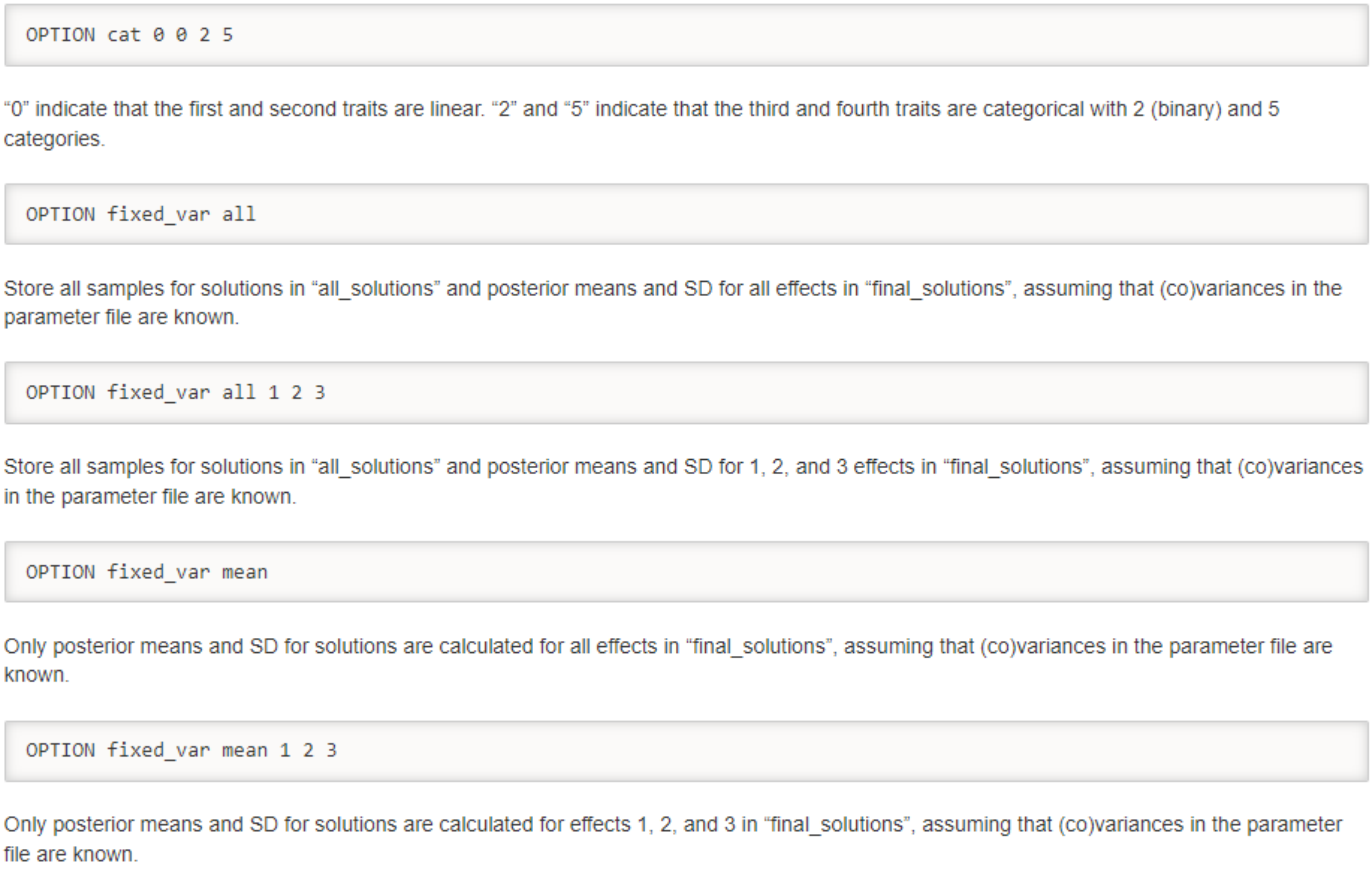
gibbsf90+

- Procedure
  - Run `gibbsf90+` to estimate variance components
  - Run `postgibbsf90` to process the samples and check convergence
  - Run `gibbsf90+` with new variance components to compute EBV (2k to 10k samples)

```
OPTION fixed_var mean X
```

Number of the animal effect

gibbsf90+



```
OPTION cat 0 0 2 5
```

“0” indicate that the first and second traits are linear. “2” and “5” indicate that the third and fourth traits are categorical with 2 (binary) and 5 categories.

```
OPTION fixed_var all
```

Store all samples for solutions in “all_solutions” and posterior means and SD for all effects in “final_solutions”, assuming that (co)variances in the parameter file are known.

```
OPTION fixed_var all 1 2 3
```

Store all samples for solutions in “all_solutions” and posterior means and SD for 1, 2, and 3 effects in “final_solutions”, assuming that (co)variances in the parameter file are known.

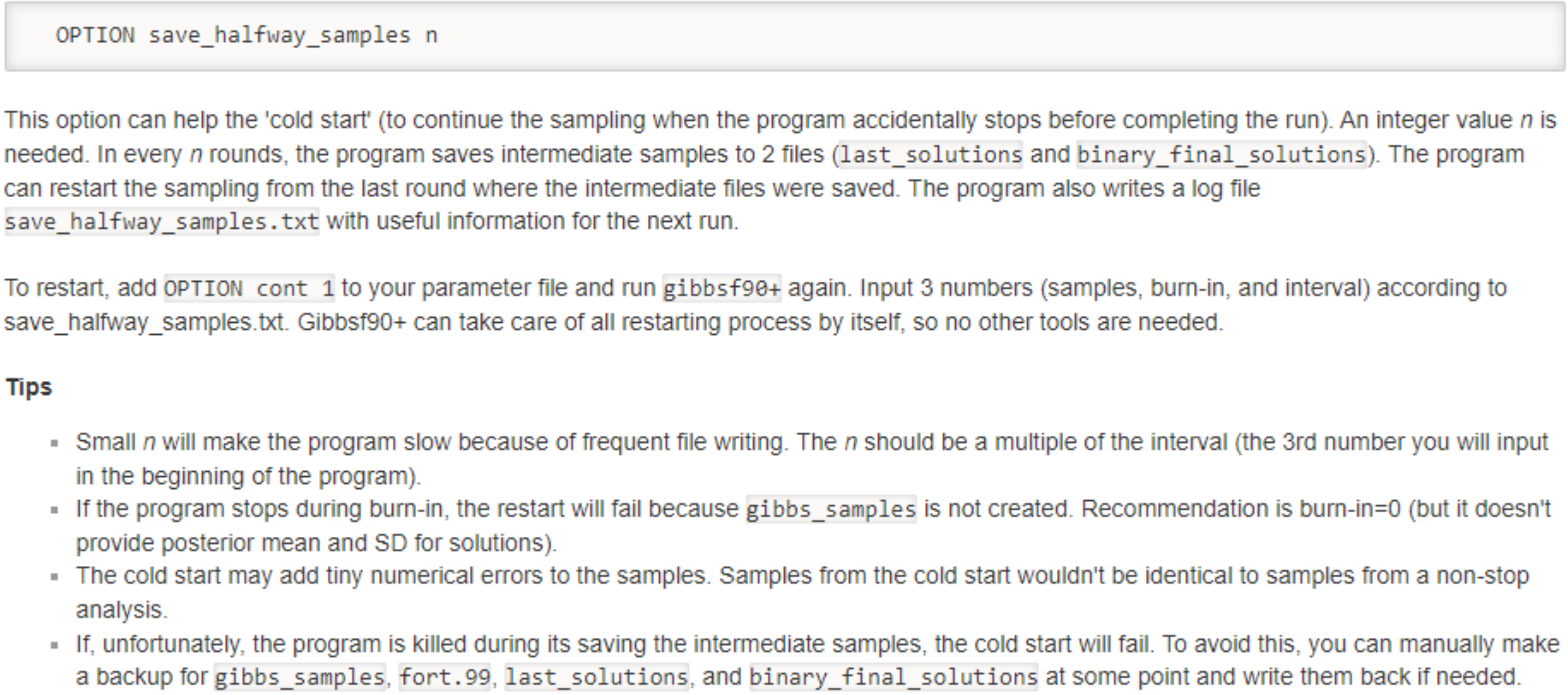
```
OPTION fixed_var mean
```

Only posterior means and SD for solutions are calculated for all effects in “final_solutions”, assuming that (co)variances in the parameter file are known.

```
OPTION fixed_var mean 1 2 3
```

Only posterior means and SD for solutions are calculated for effects 1, 2, and 3 in “final_solutions”, assuming that (co)variances in the parameter file are known.

gibbsf90+



```
OPTION save_halfway_samples n
```

This option can help the 'cold start' (to continue the sampling when the program accidentally stops before completing the run). An integer value *n* is needed. In every *n* rounds, the program saves intermediate samples to 2 files (`last_solutions` and `binary_final_solutions`). The program can restart the sampling from the last round where the intermediate files were saved. The program also writes a log file `save_halfway_samples.txt` with useful information for the next run.

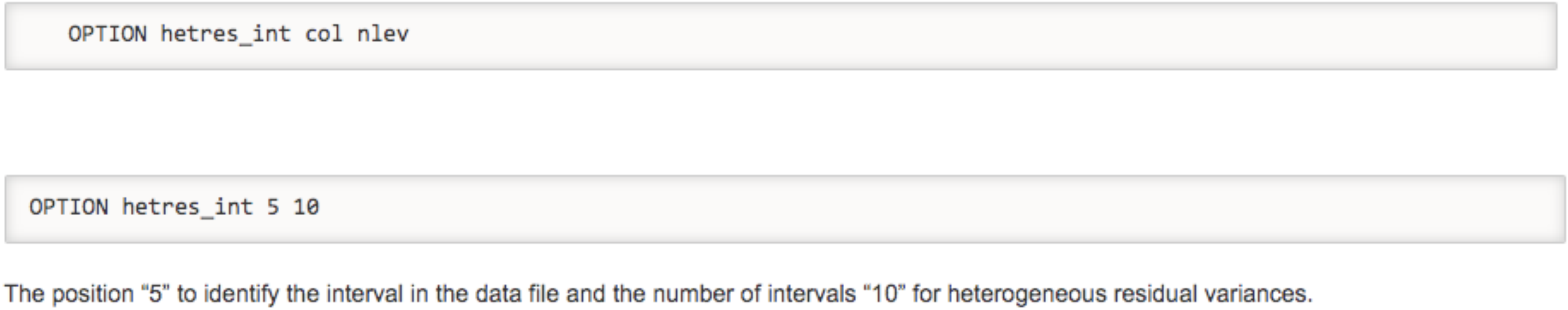
To restart, add `OPTION cont 1` to your parameter file and run `gibbsf90+` again. Input 3 numbers (samples, burn-in, and interval) according to save_halfway_samples.txt. Gibbsf90+ can take care of all restarting process by itself, so no other tools are needed.

Tips

- Small *n* will make the program slow because of frequent file writing. The *n* should be a multiple of the interval (the 3rd number you will input in the beginning of the program).
- If the program stops during burn-in, the restart will fail because `gibbs_samples` is not created. Recommendation is burn-in=0 (but it doesn't provide posterior mean and SD for solutions).
- The cold start may add tiny numerical errors to the samples. Samples from the cold start wouldn't be identical to samples from a non-stop analysis.
- If, unfortunately, the program is killed during its saving the intermediate samples, the cold start will fail. To avoid this, you can manually make a backup for `gibbs_samples`, `fort.99`, `last_solutions`, and `binary_final_solutions` at some point and write them back if needed.

```
OPTION se_covar_function <label> <function>
```

gibbsf90+

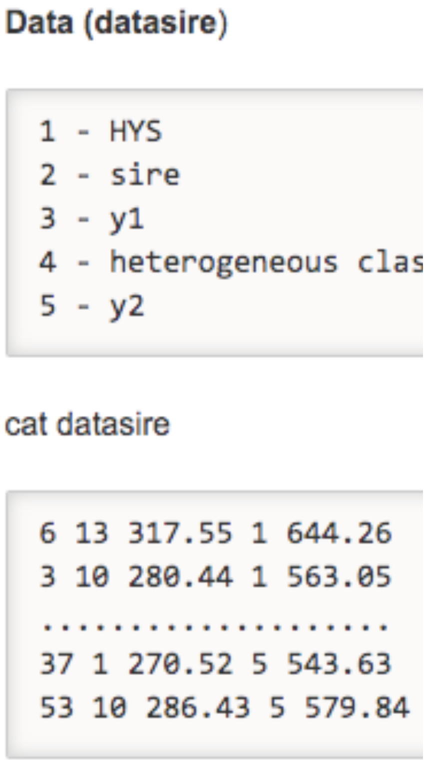


```
OPTION hetres_int col nlev
```

```
OPTION hetres_int 5 10
```

The position “5” to identify the interval in the data file and the number of intervals “10” for heterogeneous residual variances.

gibbsf90+

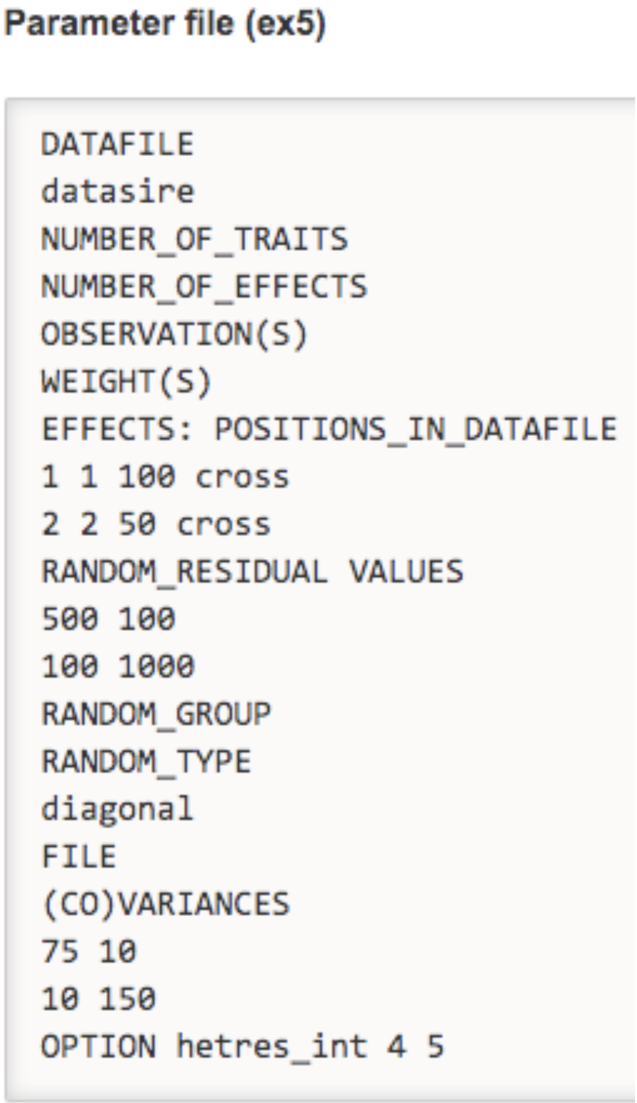


Data (datasire)

```
1 - HYS
2 - sire
3 - y1
4 - heterogeneous clas
5 - y2
```

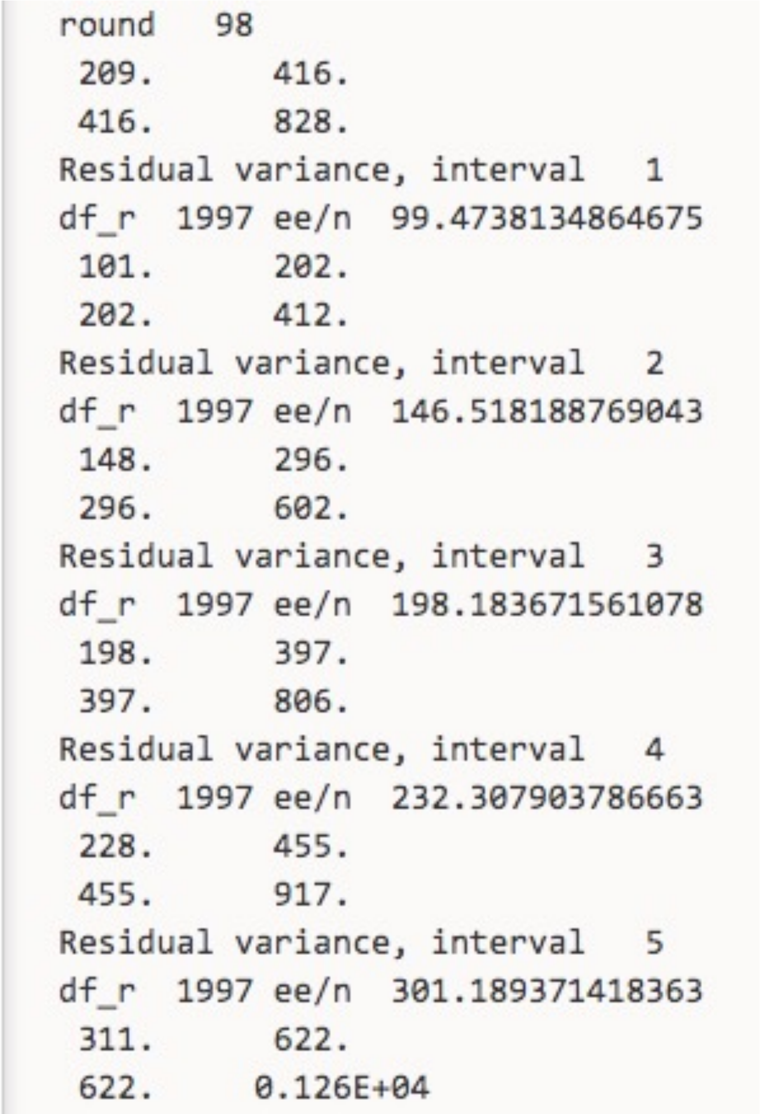
cat datasire

```
6 13 317.55 1 644.26
3 10 280.44 1 563.05
....................
37 1 270.52 5 543.63
53 10 286.43 5 579.84
```



Parameter file (ex5)

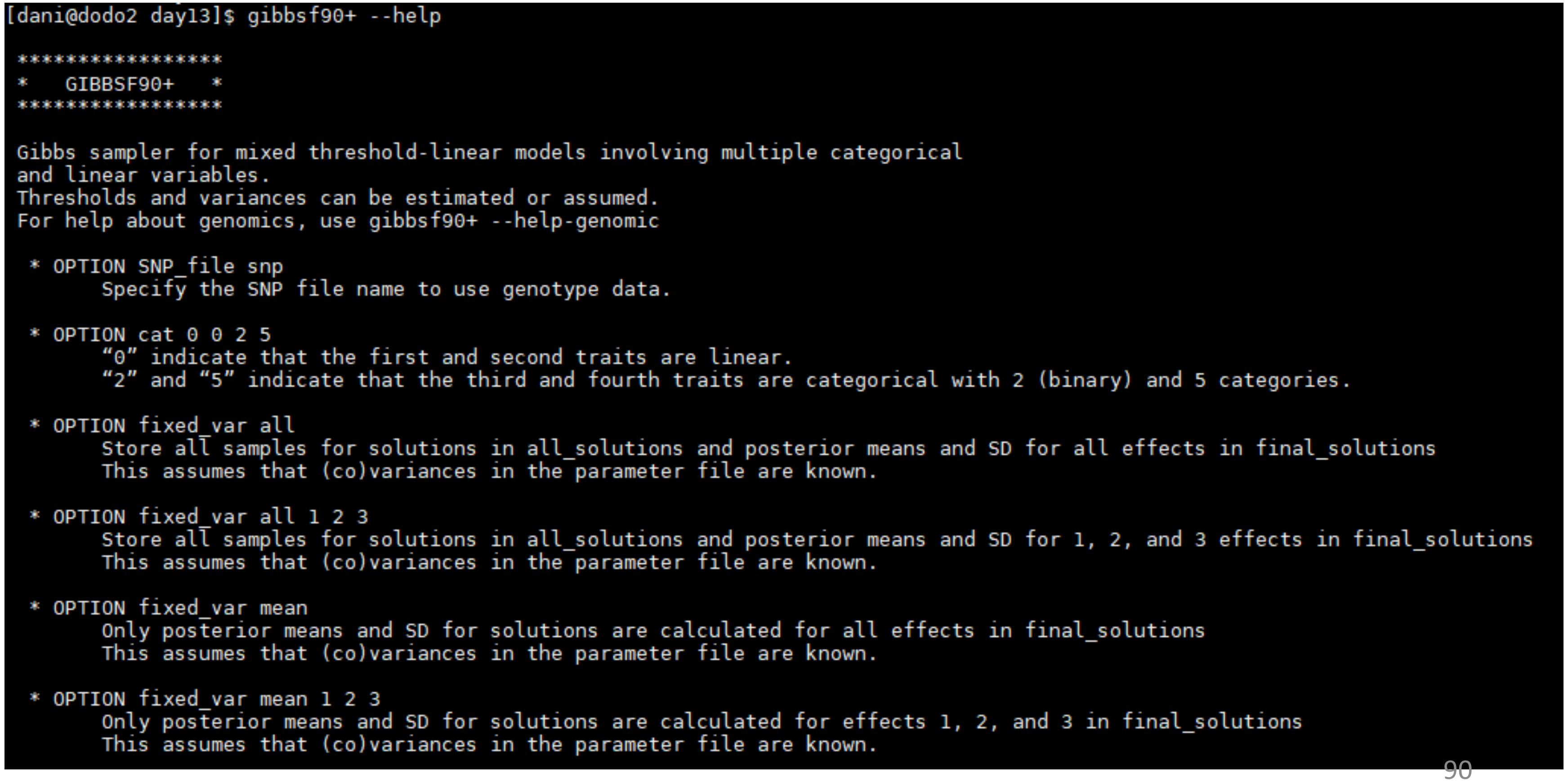
```
DATAFILE
datasire
NUMBER_OF_TRAITS
NUMBER_OF_EFFECTS
OBSERVATION(S)
WEIGHT(S)
EFFECTS: POSITIONS_IN_DATAFILE
1 1 100 cross
2 2 50 cross
RANDOM_RESIDUAL VALUES
500 100
100 1000
RANDOM_GROUP
RANDOM_TYPE
diagonal
FILE
(CO)VARIANCES
75 10
10 150
OPTION hetres_int 4 5
```



```
round    98
 209.       416.
 416.       828.
Residual variance, interval   1
df_r  1997 ee/n  99.4738134864675
 101.       202.
 202.       412.
Residual variance, interval   2
df_r  1997 ee/n  146.518188769043
 148.       296.
 296.       602.
Residual variance, interval   3
df_r  1997 ee/n  198.183671561078
 198.       397.
 397.       806.
Residual variance, interval   4
df_r  1997 ee/n  232.307903786663
 228.       455.
 455.       917.
Residual variance, interval   5
df_r  1997 ee/n  301.189371418363
 311.       622.
 622.     0.126E+04
```

gibbsf90+ quick trick

- `gibbsf90+ --help`



```
[dani@dodo2 day13]$ gibbsf90+ --help

 *****************
 *   GIBBSF90+   *
 *****************

 Gibbs sampler for mixed threshold-linear models involving multiple categorical
 and linear variables.
 Thresholds and variances can be estimated or assumed.
 For help about genomics, use gibbsf90+ --help-genomic

  * OPTION SNP_file snp
        Specify the SNP file name to use genotype data.

  * OPTION cat 0 0 2 5
        “0” indicate that the first and second traits are linear.
        “2” and “5” indicate that the third and fourth traits are categorical with 2 (binary) and 5 categories.

  * OPTION fixed_var all
        Store all samples for solutions in all_solutions and posterior means and SD for all effects in final_solutions
        This assumes that (co)variances in the parameter file are known.

  * OPTION fixed_var all 1 2 3
        Store all samples for solutions in all_solutions and posterior means and SD for 1, 2, and 3 effects in final_solutions
        This assumes that (co)variances in the parameter file are known.

  * OPTION fixed_var mean
        Only posterior means and SD for solutions are calculated for all effects in final_solutions
        This assumes that (co)variances in the parameter file are known.

  * OPTION fixed_var mean 1 2 3
        Only posterior means and SD for solutions are calculated for effects 1, 2, and 3 in final_solutions
        This assumes that (co)variances in the parameter file are known.
```

gibbsf90+ quick trick II

- Optimizing gibbsf90+ when using genomic data

Run renumf90 with the following option:

```
OPTION animal_order genotypes
```

Run gibbsf90+ with the following option:

```
OPTION separate_dense
```

postgibbsf90

- Basic idea of post-Gibbs analysis:

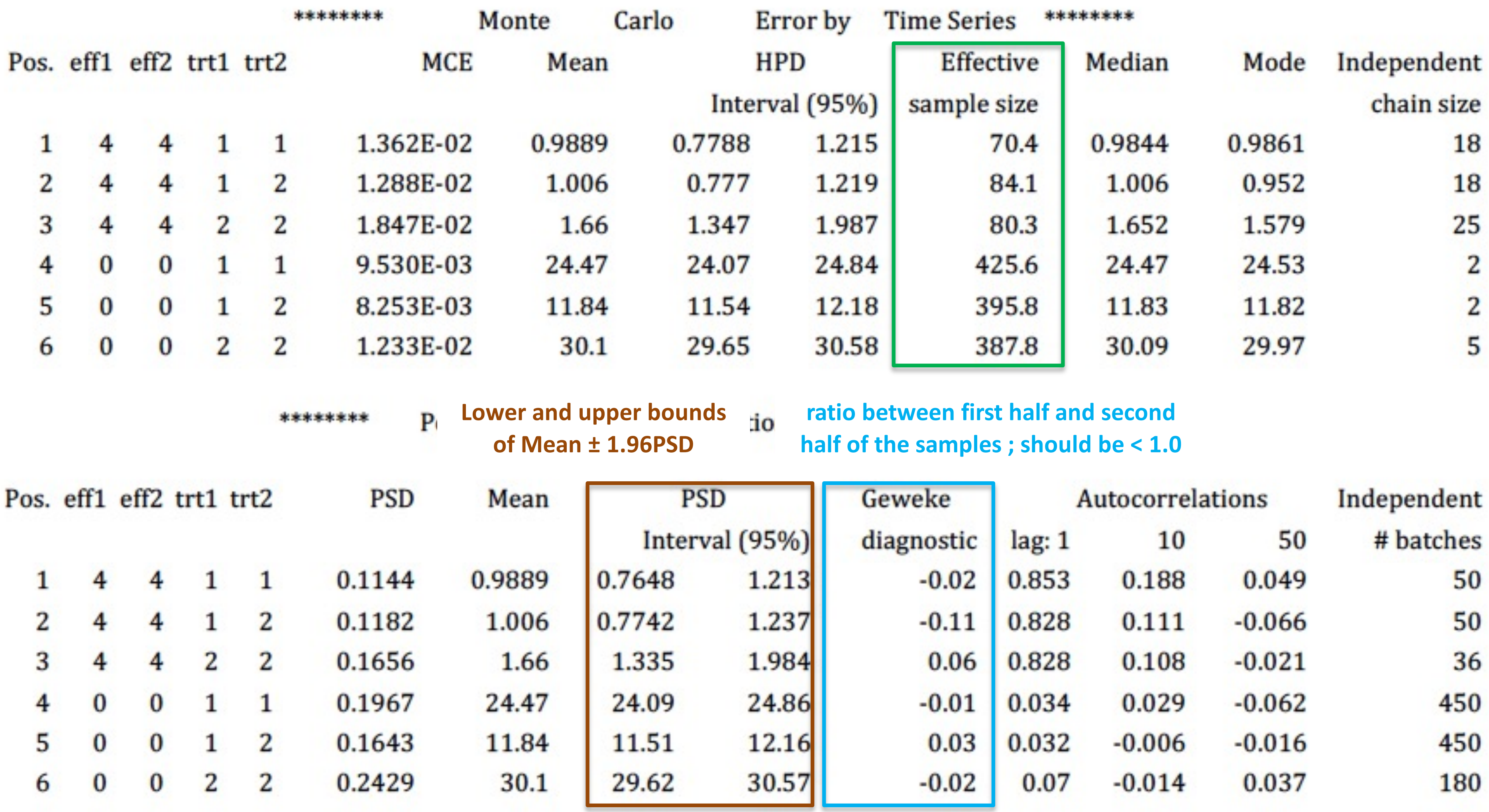
- Summarize and visualize the samples drawn by gibbsf90+
- Confirm if the chain converged
- Find the most probable value = posterior mode as a “point estimate”
- Find the reliability of the estimates = the highest posterior density as a “confidence interval”

postgibbsf90

- Name of parameter file?
  gibbs1.par
- Burn-in?
  0
- Give n to store every n-th sample? (1 means read all samples)
  10

- input files
  gibbs_samples, fort.99
- output files
  "postgibbs_samples"
    all Gibbs samples for additional post analyses
  "postmean"
    posterior means
  "postsd"
    posterior standard deviations
  "postout"

postgibbsf90



at least > 10 is recommended
> 30 may be better

******** Monte Carlo Error by Time Series ********

Pos.	eff1	eff2	trt1	trt2	MCE	Mean	HPD Interval (95%)		Effective sample size	Median	Mode	Independent chain size
1	4	4	1	1	1.362E-02	0.9889	0.7788	1.215	70.4	0.9844	0.9861	18
2	4	4	1	2	1.288E-02	1.006	0.777	1.219	84.1	1.006	0.952	18
3	4	4	2	2	1.847E-02	1.66	1.347	1.987	80.3	1.652	1.579	25
4	0	0	1	1	9.530E-03	24.47	24.07	24.84	425.6	24.47	24.53	2
5	0	0	1	2	8.253E-03	11.84	11.54	12.18	395.8	11.83	11.82	2
6	0	0	2	2	1.233E-02	30.1	29.65	30.58	387.8	30.09	29.97	5

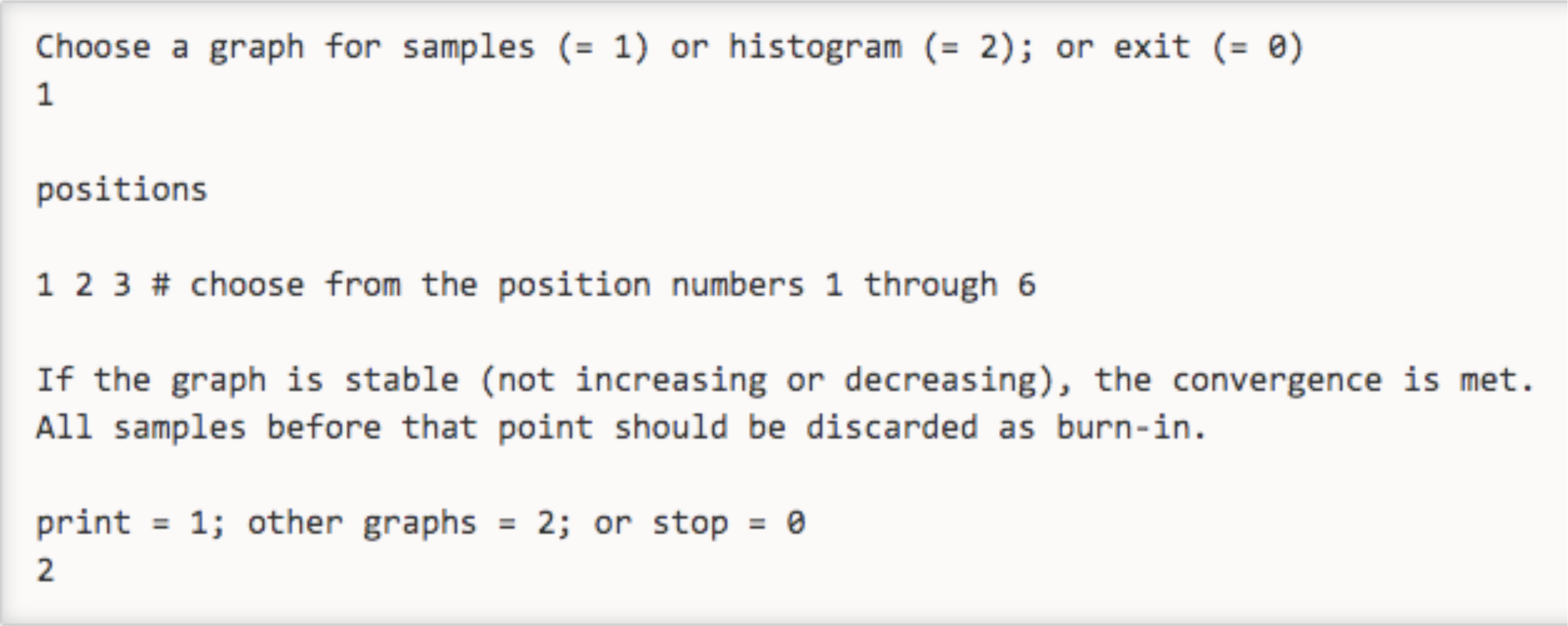
******** P tio

Lower and upper bounds of Mean ± 1.96PSD

ratio between first half and second half of the samples ; should be < 1.0

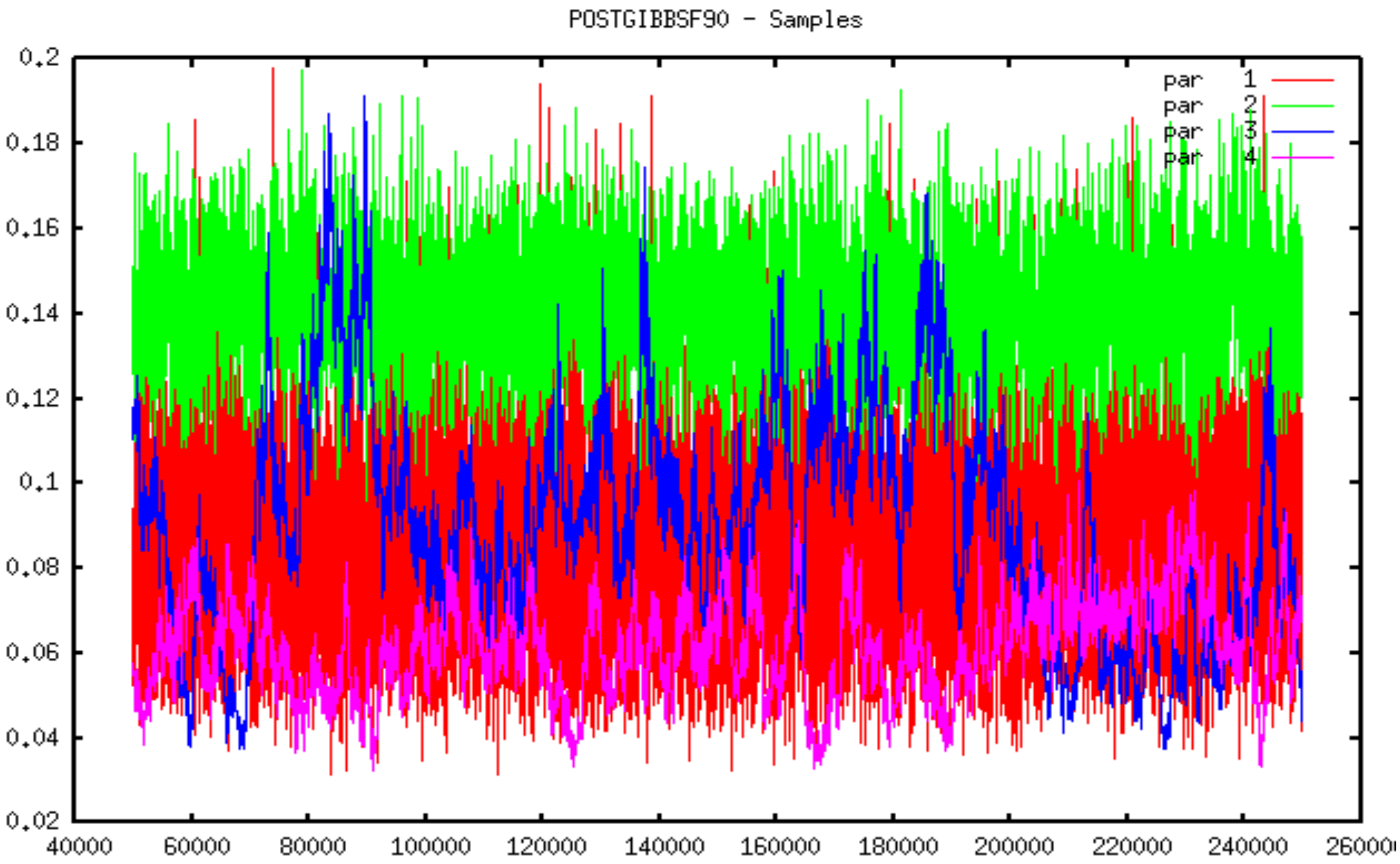
Pos.	eff1	eff2	trt1	trt2	PSD	Mean	PSD Interval (95%)		Geweke diagnostic	Autocorrelations lag: 1	10	50	Independent # batches
1	4	4	1	1	0.1144	0.9889	0.7648	1.213	-0.02	0.853	0.188	0.049	50
2	4	4	1	2	0.1182	1.006	0.7742	1.237	-0.11	0.828	0.111	-0.066	50
3	4	4	2	2	0.1656	1.66	1.335	1.984	0.06	0.828	0.108	-0.021	36
4	0	0	1	1	0.1967	24.47	24.09	24.86	-0.01	0.034	0.029	-0.062	450
5	0	0	1	2	0.1643	11.84	11.51	12.16	0.03	0.032	-0.006	-0.016	450
6	0	0	2	2	0.2429	30.1	29.62	30.57	-0.02	0.07	-0.014	0.037	180

postgibbsf90

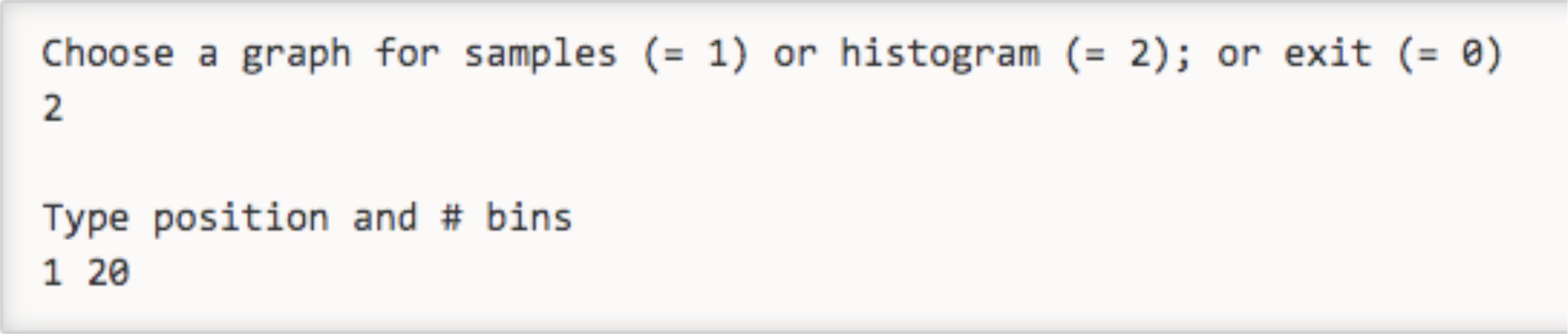


```
Choose a graph for samples (= 1) or histogram (= 2); or exit (= 0)
1

positions

1 2 3 # choose from the position numbers 1 through 6

If the graph is stable (not increasing or decreasing), the convergence is met.
All samples before that point should be discarded as burn-in.

print = 1; other graphs = 2; or stop = 0
2
```

postgibbsf90

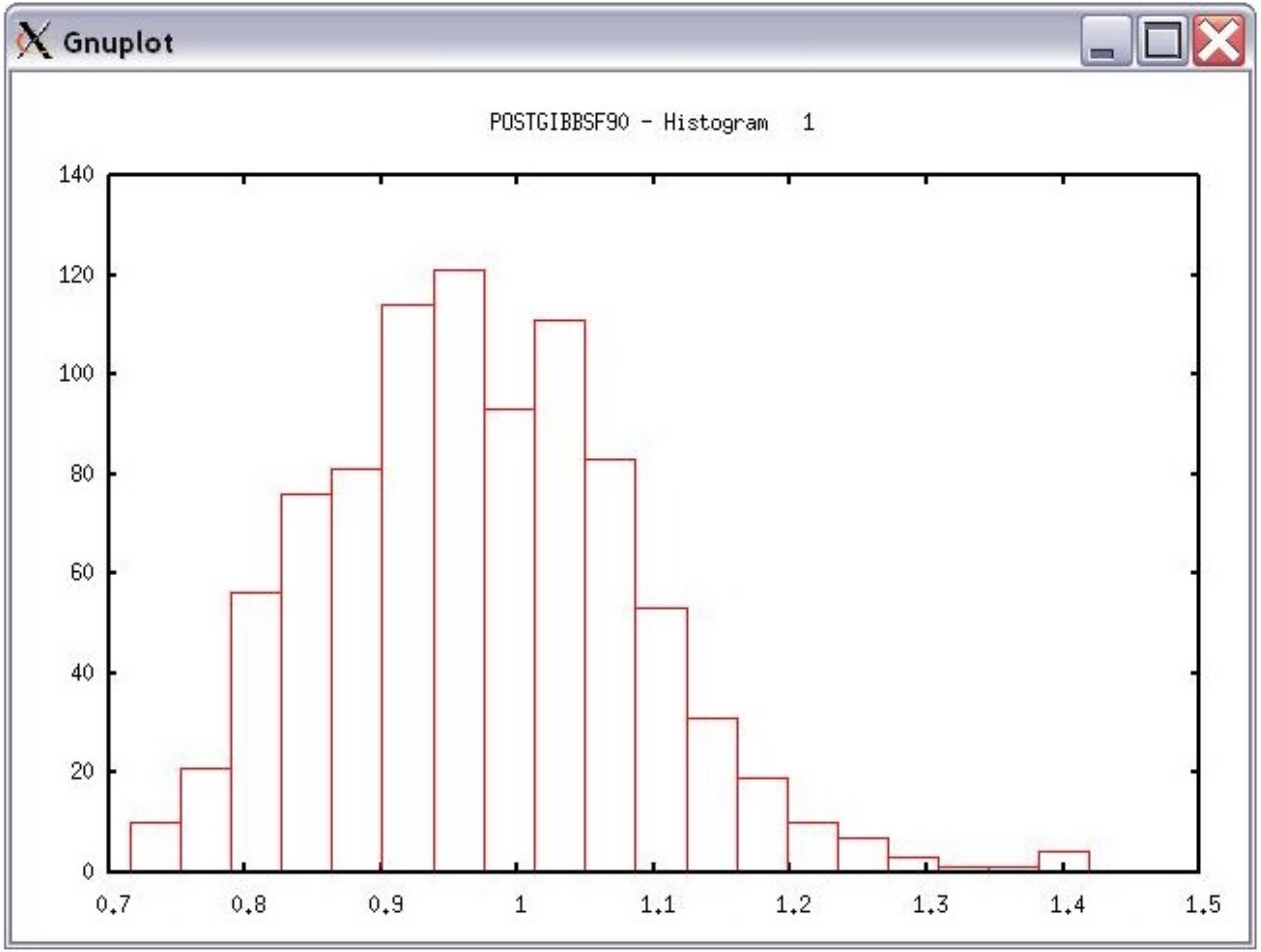


postgibbsf90



```
Choose a graph for samples (= 1) or histogram (= 2); or exit (= 0)
2

Type position and # bins
1 20
```

postgibbsf90



Common problems for BLUPF90 family

- Wrong position or formats for observation and effects
- Misspelling of Keywords
  - Program may stop
- (Co)variance matrices not symmetric, not positive definite
  - Program may not stop
- Large numbers (e.g., 305-day milk yield 10,000 kg)
  - Scale down i.e., 10,000 /1,000 = 10

General output from BLUPF90 family

- Output printed on the screen is not saved to any file!
- Should use redirection or pipes to store output

renumf90

```
renumf90 renum.par | tee renum.log
```

blupf90+

```
blupf90+ renf90.par | tee blup.log
```

gibbsf90+

```
gibbsf90+ exmr99s1 --samples 1000 --burnin 0 --interval 1 | tee gibbs.log
```