

Tutorial: Introdução ao Software Brblup



DOCUMENTOS 168

Tutorial: Introdução ao Software Brblup

Roberto Iroshi Higa

Autor

Exemplares desta publicação podem ser adquiridos na:

Embrapa Informática Agropecuária

Av. Dr. André Tosello, 209 - Cidade Universitária
Campinas, SP, Brasil
CEP. 13083-886
Fone: (19) 3211-5700
www.embrapa.br
www.embrapa.br/fale-conosco/sac

Comitê Local de Publicações
da Unidade Responsável

Presidente

Stanley Robson de Medeiros Oliveira

Secretaria-Executiva

Maria Fernanda Moura

Membros

Adriana Farah Gonzalez, membro nato, Alexandre de Castro, membro indicado, Carla Cristiane Osawa, membro nato, Debora Pignatari Drucker, membro eleito, Ivan Mazoni, membro eleito, João Camargo Neto, membro indicado, Joao Francisco Goncalves Antunes, membro eleito, Magda Cruciol, membro nato.

Revisão de texto

Adriana Farah Gonzalez

Normalização bibliográfica

Carla Cristiane Osawa

Projeto gráfico da coleção

Carlos Eduardo Felice Barbeiro

Editoração eletrônica

Mariana Pilatti sob supervisão de Magda Cruciol

Foto da capa

Gisele Rosso, Pexels e Mariana Pilatti

1ª edição

Versão digital (2020)

Todos os direitos reservados.

A reprodução não autorizada desta publicação, no todo ou em parte,
constitui violação dos direitos autorais (Lei nº 9.610).

Dados Internacionais de Catalogação na Publicação (CIP)
Embrapa Informática Agropecuária

Higa, Roberto Hitoshi,

Tutorial : Introdução ao Software Brblup / Roberto Hitoshi Higa. -
Campinas : Embrapa Informática Agropecuária, 2020.

PDF (42 p.) : il. color. - (Documentos / Embrapa Informática
Agropecuária, ISSN 1677-9274 ; 168).

1.Tutorial. 2. Software. 3. Software brblup. I. Título. II. Embrapa
Informática Agropecuária. III. Série.

CDD (21. ed.) 005.1068

Autor

Roberto Iroshi Higa

Engenheiro Elétrico, doutor em Engenharia Elétrica, pesquisador da Embrapa Informática Agropecuária, Campinas, SP.

Apresentação

Programas de melhoramento genético animal atualmente utilizam softwares para realização de avaliações genéticas importados, via de regra desenvolvidos por Universidades estrangeiras, devido à ausência de produtos similares no mercado interno. Como resultado, observa-se uma dependência tecnológica do setor em relação a esses fornecedores, além de um custo anual com licenciamento de software que pode chegar ao patamar dos 30.000 dólares/ano.

Desde 2014, as Embrapas Informática Agropecuária e Gado de Corte e o programa Embrapa/Geneplus vêm desenvolvendo uma solução computacional para realizar avaliações genéticas em grandes populações de animais. Resultado dessa colaboração, o brbv e seu aplicativo brblup associados têm por objetivo tanto prover independência tecnológica aos programas de melhoramento genético animal nacionais quanto fomentar pesquisa e desenvolvimento nacionais no tema.

Este tutorial apresenta uma parte do resultado obtido até o presente momento no desenvolvimento do brbv e seus aplicativos. Trata-se de um passo a passo para iniciantes com a apresentação de diversos exemplos que ilustram tanto a forma de especificar uma gama de modelos utilizados na avaliação genética de animais como a utilização brblup para obtenção dos correspondentes valores genéticos.

Silvia Maria Fonseca Silveira Massruhá

Chefe-geral

Embrapa Informática Agropecuária

Sumário

| | |
|----------------------------------------------|----|
| Introdução | 9 |
| Pré-requisitos | 9 |
| Instalação | 11 |
| Aplicando o Brblup a Conjuntos de Dados..... | 13 |
| Considerações Finais | 40 |
| Referências | 41 |

Introdução

Programas de melhoramento genético animal têm por objetivo promover o aumento de alelos desejáveis em uma população, identificando os indivíduos superiores para características de interesse da cadeia produtiva, particularmente aquelas de importância econômica. Para isso, faz-se necessária a realização de avaliações genéticas que, a partir de dados fenotípicos e genealógicos, utilizam modelos genético-estatísticos para predição do valor genético de cada indivíduo para as diferentes características de interesse (Menezes et al., 2013).

Embora a literatura mundial seja rica em modelos genético-estatísticos, cada vez mais acurados e complexos, não se observa a sua adoção em rotinas de avaliação dos programas de melhoramento de gado de corte no Brasil (Menezes, 2013). Dentre as razões que explicam esse fato, está o acesso restrito que os programas de melhoramento genético nacionais têm a softwares com capacidade de realizar avaliações genéticas de grandes populações, utilizando modelos mais complexos. Atualmente, todos eles utilizam softwares importados, via de regra desenvolvidos por Universidades estrangeiras, devido à ausência de produtos similares no mercado interno. Como resultado, observa-se uma dependência tecnológica do setor em relação a esses fornecedores, além de um custo anual com licenciamento de software, que pode chegar ao patamar dos 30.000 dólares/ano.

O projeto MaxiDep (Menezes, 2013) teve como um de seus objetivos a redução desse ‘gap’ tecnológico, tanto desenvolvendo uma solução computacional capaz de realizar avaliações genéticas em grandes populações de animais quanto fomentando pesquisa e desenvolvimento no tema. Resultado desse projeto, o software brblup é uma solução totalmente desenvolvida pela Embrapa, no escopo do programa de melhoramento genético Embrapa/Geneplus. Ela é distribuída como um pacote Python e disponibiliza uma interface de ‘linha-de-comando’ para resolução de modelos genético-estatísticos em avaliação genética animal, utilizando funcionalidades implementadas em um segundo pacote Python denominado brbv, também desenvolvido pela Embrapa. Essas funcionalidades contemplam a especificação e resolução, utilizando Best Linear Unbiased Prediction (BLUP) (Mrode, 2014), de diferentes modelos de avaliação bem como o cálculo das correspondentes acurácia. Além disso, um terceiro pacote Python, que provê funcionalidades para preparação de dados, denominado brtools, também acompanha a distribuição do brblup.

O presente tutorial é um passo a passo para utilização do software brblup, incluindo a instalação dos pacotes brbv, brblup e brtools, e focando em um conjunto de exemplos ilustrativos das funcionalidades que estes softwares provêm. As demais seções do presente documento encontram-se organizadas da seguinte forma: a) a seção ‘Pré-requisitos’ apresenta os pré-requisitos necessários para instalação dos pacotes brbv, brblup e brtools; b) a seção ‘Instalação’ apresenta os procedimentos de instalação, propriamente ditos; enquanto que c) a seção ‘Aplicando o brblup a conjunto de dados’ apresenta o uso do brblup com dez diferentes conjuntos de dados; os comentários finais e bibliografias relevantes são apresentados nas seções; d) ‘Comentários finais’; e e) ‘Bibliografia’.

Pré-requisitos

Os pacotes brbv, brblup e brtools foram desenvolvidos para o sistema operacional Linux/Ubuntu (release 18.04), utilizando a linguagem de programação Python (versão 3.6.8) e os seguintes pacotes associados: Numpy (versão 1.17.2), Scipy (versão 1.3.1) e Pytables (versão 3.5.2). Caso

esses pacotes ainda não tenham sido instalados, é preciso executar os seguintes comandos em um terminal (*shell* do Unix):

```
> sudo apt-get upgrade  
> sudo apt install python3-pip  
> sudo apt-get install libhdf5-dev  
> sudo apt install python3-numpy python3-scipy
```

A utilização da ferramenta *Virtualenv* (versão 16.7.5) permite que os passos contidos neste tutorial sejam executados em um ambiente totalmente isolado da instalação default de Python. Também é possível criar diferentes ambientes virtuais, independentes entre si, tal que em caso de necessidade, o tutorial pode ser executado diversas vezes, de forma independente, bastando utilizar diferentes ambientes virtuais para cada caso. Por fim, ao longo deste tutorial, será assumido que todos os arquivos utilizados estejam organizados sob um mesmo subdiretório (ex: *demo-brblup-v3*). Para criar o ambiente virtual, executam-se os seguintes comandos no mesmo terminal:

```
> sudo apt install virtualenv  
> mkdir demo-brblup-v3  
> cd demo-brblup-v3  
> virtualenv -p /usr/bin/python3 venv
```

Running virtualenv with interpreter /usr/bin/python3

...

Installing setuptools, pip, wheel...

done.

Note que o comando *virtualenv* cria um subdiretório denominado *venv* sob o subdiretório *demo-brblup-v3*. É neste subdiretório que ficarão armazenados os arquivos correspondentes à instalação virtual de Python, e onde serão instalados os pacotes *brbv*, *brblup* e *brtools*. A utilização da opção *-p /user/bin/python3* garante que essa instalação seja baseada na versão 3 de Python. Para ativar o ambiente virtual criado, deve-se utilizar o seguinte comando:

```
> source venv/bin/activate  
(venv) >
```

O prompt (venv) indica que o ambiente virtual está ativo, ou seja, que todos os comandos Python executados nesse ambiente utilizam as versões que foram instaladas no ambiente virtual. Ao final de uma seção de trabalho, para desativar o ambiente virtual, deve-se executar o seguinte comando:

```
(venv) > deactivate
```

```
>
```

A partir da próxima seção, será assumido que todos os comandos serão executados com o ambiente virtual (venv) ativado e que o diretório de trabalho é o diretório ‘.... /demo-brblup-v3/’ .

Instalação

Para utilizar o software brblup é preciso instalar dois pacotes Python: brbv e brblup. Adicionalmente, para utilização de funcionalidade de apoio à preparação de dados, deve-se também instalar o pacote brtools. Os pacotes distribuídos para instalação são¹:

- brbv-2.0.0-cp36-cp36m-linux_x86_64.whl
- brtools-2.0.0.tar.gz
- brblup-3.0.0.tar.gz

Para instalação desses pacotes utiliza-se o processo padrão de Python (utilização do comando pip3), apenas assegurando que o pacote brbv preceda a instalação dos outros dois pacotes. Para instalação dos três pacotes, deve-se executar os seguintes comandos:

```
(venv) > pip3 install brbv-2.1.0-cp36-cp36m-linux_x86_64.whl
```

```
Processing ./brbv-2.0.0-cp36-cp36m-linux_x86_64.whl
```

```
...
```

```
Successfully built brbv
```

```
Installing collected packages: numpy, scipy, six, numexpr, mock, tables, brbv
```

```
Successfully installed brbv-1.0.0 mock-3.0.5 numexpr-2.7.0 numpy-1.17.2 scipy-1.3.1 six-1.12.0
tables-3.5.2
```

```
(venv) > pip3 install brtools-2.0.0.tar.gz
```

```
Processing ./brtools-2.0.0.tar.gz
```

```
...
```

```
Successfully built brtools
```

¹ Os nomes dos arquivos de distribuição podem variar de acordo com a versão do software e a modalidade da distribuição.

Installing collected packages: brtools

Successfully installed brtools-2.0.0

(venv) > pip3 install brblup-3.1.0.tar.gz

Processing ./brblup-3.0.0.tar.gz

...

Successfully built brblup

Installing collected packages: brblup

Successfully installed brblup-3.0.0

Se o processo de instalação tiver sido bem-sucedido, os seguintes comandos passam a ficar disponíveis para execução: brblup_app e brtools_app. Para verificar se os mesmos se encontram disponíveis no ambiente virtual, execute cada um deles com a opção –help, conforme exemplificado abaixo para o caso de brblup_app:

(venv) > brblup_app –help

usage: brblup [-h] [-i CONF] [-o OUT] [-p PROG] [-r PROC] [-b INBREED]

[-k BLOCK] [-t TOL] [-n NITER]

Mixed model solver (BLUP). Version 3.0.0.

optional arguments:

-h, --help show this help message and exit

-i CONF, --input CONF

input: path to model specification file (XML format).

-o OUT, --output OUT output: path to solution (CSV format).

-r PROC, --processing PROC

processing: storage mode (memory or disk).

-k BLOCK, --block BLOCK

algorithm: block preconditioning specification (0 to 9).

-p PEV, --pev PEV pev: calculate pev option (yes or no). Default: No

-t TOL, --tol TOL algorithm: solution error tolerance.

-n NITER, --niter NITER

algorithm: maximum number of iterations.

Aplicando o brblup a conjuntos de dados

Esta seção ilustra a utilização do software brblup (comando brblup_app) por meio de uma série de exemplos, sendo oito deles (exemplos 1 a 8) exemplos didáticos apresentados por Mrode (2014), um baseado em dados gerados artificialmente por meio de simulação computacional (Simulado_7) e um (senepol1) referente a um conjunto de dados reais de animais não identificados da raça Senepol. Para cada exemplo serão apresentados o arquivo de configuração, em formato xml, e que define: a) o modelo genético-estatístico e parâmetros utilizados; b) os comandos para execução do software brblup; e c) a saída resultante (solução do modelo). Considera-se que todos os conjuntos de dados estão organizados sob o diretório ‘....../demo-brblup-v3/tests/data’. A correção da maioria das soluções obtidas por meio de brblup pode ser verificada junto à sua referência de origem (Mrode, 2014), a menos de diferenças de precisão numérica. Para os demais exemplos, bem como para alguns dos casos apresentados por Mrode (2014), comentários a respeito de eventuais especificidades serão apresentados junto às apresentações dos próprios exemplos.

Exemplo 1. Modelo Animal

O exemplo 1 (Mrode, 2014) é um modelo animal univariado que avalia o ganho de peso dos animais antes da desmama como característica de interesse e considera o sexo como efeito fixo. A variância genética considerada é igual a 20 e a variância residual igual a 40. Os arquivos de dados podem ser encontrados no diretório ‘tests/data/mrode_3.1’ e a especificação do modelo no arquivo de configuração ‘tests/data/mrode_3.1/conf.xml’, reproduzido a seguir.

```
<?xml version="1.0"?>
<brblup>
  <dataset filename="tests/data/mrode_3.1/data.csv">
    <!-- traits -->
    <trait type="real" column="2" levels="1"/>
    <!-- effects -->
    <effect type="alpha" columns="0" levels="8"/>
    <effect type="alpha" columns="1" levels="2"/>
  </dataset>
  <model>
    <!-- one equation for each trait -->
```

```

<equation trait="0" effects="0,1"/>
<!-- residual covariances -->
<covariances>40</covariances>
<random effects="0" levels="8" type="animal_additive" source="pedigree" filename="tests/data/mrode_3.1/pedigree.csv" id="0" sire="1" dam="2">
<!-- genetic covariances. -->
<covariances>20</covariances>
</random>
</model>
</brblup>

```

O arquivo de configuração possui duas seções principais, delimitadas pelas tags XML <dataset> e <model>. Na primeira são especificadas as informações sobre efeitos e características, extraídas do arquivo de dados e utilizadas pelo modelo genético-estatístico; enquanto que a segunda especifica o modelo propriamente dito (equações, grupos de efeitos genéticos e covariâncias). No presente exemplo:

- a tag <data> especifica que os dados encontram-se no arquivo “tests/data/mrode_3.1/data.csv” e que o identificador do animal encontra-se na primeira coluna (“0”). As tags aninhadas <trait> e <effect> especificam colunas dos arquivos de dados relevantes para o modelo e o tipo de dado a que cada um corresponde. Especificamente, a coluna “2” corresponde à uma característica quantitativa (“real”) e as colunas “0” e “1” a efeitos qualitativos (“alpha”), o primeiro com 8 e o segundo com 2 níveis. Note que para características e efeitos quantitativos o número de níveis não é relevante, sendo sempre indicado como sendo igual a “1”.
- Aninhadas na tag <model> estão as tags que descrevem o modelo: <equation>, <covariances> e <random>.
- a tag <equation> especifica que a primeira característica (“0”) listada na seção <dataset> é função da contribuição do primeiro efeito (“0”) listado na seção <dataset> mais a contribuição do segundo efeito (“1”) listado na seção <dataset>.
- a tag <covariances> nesse nível refere-se à covariância residual, que para o presente exemplo é igual a 40.
- a tag <random> especifica grupos de efeitos aleatórios e as correspondentes covariâncias. Para o presente exemplo, o grupo aleatório é composto apenas pelo primeiro efeito (“0”) listado na seção <dataset>, é do tipo “animal_additive” e possui covariância (genética) igual a 20. A matriz de parentesco, utilizada para obtenção das covariâncias entre níveis de efeitos (animais), é construída a partir de um arquivo de pedigree, “tests/data/mrode_3.1/data.csv”, onde a primeira coluna do arquivo (“0”) refere-se ao identificador do animal; e a segunda (“1”) e a terceira (“2”) aos correspondentes pai e mãe. Também é indicado que o número total de animais no pedigree é 8.

Para resolver o modelo especificado (avaliação genética) executa-se o comando brblup_app indicando como parâmetro o arquivo de configuração ‘tests/data/mrode_3.1/conf.xml’:

```
(venv) > brblup_app -i tests/data/mrode_3.1/conf.xml
```

Starting blup...

Files:

```
    Data: tests/data/mrode_3.1/data.csv
```

Pre-processing done: True

Effects:

```
    Effect: 0, Type: alpha, Column: 0, Number of Levels: 8
```

```
    Effect: 1, Type: alpha, Column: 1, Number of Levels: 2
```

Traits:

```
    Trait: 0, Type: real, Column: 2, Number of Levels: 1
```

Equations:

```
    Equation: 0, Trait: 0, Effects: 0, 1
```

Residual covariance matrix:

40.

Random groups:

```
    Random group: 0, Effects: [0], Levels: 8, Relation: animal_additive
```

Pedigree: tests/data/mrode_3.1/pedigree.csv, id column: 0, sire column: 1, dam column: 2

Covariance matrix:

20.

Setting up MME.

MME in memory. Using Block preconditioner.

running PCG algorithm step: 10 ...

Number of steps to convergence: 10.

Done!

Com exceção da especificação do arquivo de parâmetros (opção `-i`), o comando `brblup_app` considerou os valores default para as demais opções². Um total de 10 iterações ('brblup' imprime uma mensagem a cada 10 iterações executadas) do algoritmo Preconditioned Conjugate Gradient (PCG), default: preconditionador de jacobi) foram executadas e o resultado gravado no arquivo de saída '`./solution.csv`' (default: `-o solution.csv`). Para verificar seu conteúdo, utilize o comando `cat` do unix:

```
(venv) > cat solution.csv
```

| Trait | Effect | Level | Solution |
|-------|--------|----------|-----------------------|
| | | 0,0,0,0 | 0.09844457570387848 |
| | | 0,0,1,-0 | -0.018770099100872965 |
| | | 0,0,2,-0 | -0.04108420292708541 |
| | | 0,0,3,-0 | -0.00866312266194168 |
| | | 0,0,4,-0 | -0.18573209949465122 |
| | | 0,0,5,0 | 0.1768720876813018 |
| | | 0,0,6,-0 | -0.24945855483362878 |
| | | 0,0,7,0 | 0.18261468793069513 |
| | | 0,1,0,4 | 0.358502329854957 |
| | | 0,1,1,3 | 0.4044300059066748 |

No arquivo de saída (`./solution.csv`), a quarta coluna de linha apresenta a solução para um nível de um efeito para uma característica, respectivamente especificados pelas terceira, segunda e primeira colunas, sendo que a característica e o efeito são especificados pela correspondente coluna no arquivo de dados. Para este exemplo específico, a primeira linha do arquivo apresenta a solução, '0.09844457570387848', referente ao nível "0" do efeito "0" (efeito animal) da característica "0", conforme especificado no arquivo de configuração. Já a pernúltima linha apresenta a solução ('0,4.358502329854957') para o nível "0" do efeito "1" (efeito fixo sexo).

Neste ponto, vale a pena observar que os pacotes `brbv`, `brblup` e `brtools` adotam como padrão iniciar contagens a partir de '0' (zero). Assim, tanto a primeira coluna de um arquivo quanto o primeiro nível de um efeito são indicados por '0', o segundo por '1' e assim por diante. Níveis de efeitos no arquivo de dados também seguem o mesmo padrão, ou seja, são indicados por valores inteiros iniciando a partir de "0" (zero).

Para obter a acurácia, bem como outras medidas de precisão, da solução pode-se utilizar a opção '`-p yes`' ao executar o software '`brblup`'.

```
(venv) > brblup_app -i tests/data/mrode_3.1/conf.xml -p yes
```

```
Starting blup...
```

```
...
```

Number of steps to convergence: 10.

² Para verificar as opções disponíveis, execute o comando com a opção `--help`: `brblup_app --help`.

Done!

Em adição ao arquivo de saída “solution.csv” agora também será gerado o arquivo “accuracy.csv” contendo a acurácia da predição, obtida pela resolução das equações de Takahasi³(ทักษิ 1980) para obtenção de elementos da inversa da matriz de coeficientes do MME.

```
(venv) > cat accuracy.csv
```

| Trait | Effect | Level | PEV | RELI | ACC | ACC_B |
|--------------------------------------------------------------------------------------|--------|-------|-----|------|-----|-------|
| 0,0,0,18.84376845835795,0.057811577082102494,0.2404403815545602,0.02933609167853701 | | | | | | |
| 0,0,1,19.683828837697714,0.01580855811511428,0.1257320886453187,0.007935767258548054 | | | | | | |
| 0,0,2,18.25835138150555,0.08708243092472256,0.2950973244960424,0.044532800628259483 | | | | | | |
| 0,0,3,17.107206142941525,0.14463969285292377,0.3803152545624798,0.07514308828496274 | | | | | | |
| 0,0,4,17.124269869396873,0.14378650653015634,0.37919191253263346,0.0746819501004837 | | | | | | |
| 0,0,5,17.691310625451216,0.11543446872743923,0.3397564844523784,0.05948655975974448 | | | | | | |
| 0,0,6,17.67424689899587,0.11628765505020644,0.341009757998516,0.059940243947336325 | | | | | | |
| 0,0,7,16.894565859421153,0.15527170702894233,0.3940453108830789,0.08090898548018777 | | | | | | |

Exemplo 2. Modelo Repetibilidade

O exemplo 2 (Mrode, 2014) é um modelo animal aplicado a um conjunto de dados contendo mais de uma observação por animal, conhecido como modelo de repetibilidade. O exemplo considera o teor de gordura no leite em gado de leite como característica alvo e, além do efeito genético aditivo, também considera o efeito ambiental permanente (genético não aditivo) e dois efeitos fixos, o parto e o grupo de contemporâneos (do inglês herd+year+season, HYS). A variância genética considerada é igual a 20, a ambiental permanente é igual a 12 e a residual igual a 12. A especificação do modelo encontra-se no arquivo ‘tests/data/mrode_4.1/conf.xml’, reproduzido a seguir.

```
<?xml version="1.0"?>
<brblup>
<dataset filename="tests/data/mrode_4.1/data.csv">
    <!-- traits -->
    <trait type="real" column="4" levels="1"/>
    <!-- effects -->
```

³ O brblup também disponibiliza a opção ‘-enp’, que aproxima o cálculo da PEV como uma alternativa à opção ‘-pev’ para situações em que os recursos computacionais são insuficientes para a demanda de processamento.

```

<effect type="alpha" columns="0" levels="8"/>
<effect type="alpha" columns="1" levels="5"/>
<effect type="alpha" columns="2" levels="2"/>
<effect type="alpha" columns="3" levels="4"/>
</dataset>

<model>
  <!-- one equation for each trait -->
  <equation trait="0" effects="0,2,3,1"/>
  <!-- residual covariances -->
  <covariances>28</covariances>
  <random effects="1" levels="5" type="identity">
    <!-- permanent environmental covariances. -->
    <covariances>12</covariances>
  </random>
  <random effects="0" levels="8" type="animal_additive" source="pedigree" filename="tests/data/mrode_4.1/pedigree.csv" id="0" sire="1" dam="2">
    <!-- genetic covariances. -->
    <covariances>20</covariances>
  </random>
</model>
</brblup>

```

Observa-se que ele apresenta a mesma estrutura que o apresentado no exemplo 1, porém inclui um maior número de efeitos no modelo. O comando brblup_app é executado da mesma forma que no exemplo anterior, apenas especificando como parâmetros o arquivo ‘tests/data/mrode_4.1/conf.xml’:

```
(venv) > brblup_app -i tests/data/mrode_4.1/conf.xml
```

Starting blup...

...

Number of steps to convergence: 18.

Done!

E utilizando o comando cat para verificar a solução obtida:

```
(venv) > cat solution.csv
Trait,Effect,Level,Solution
0,0,0,10.147570481017123
0,0,1,-3.084152950787984
0,0,2,-7.063417530229073
0,0,3,13.580743021645915
0,0,4,-18.20697244782784
0,0,5,-18.38678173955278
0,0,6,9.328432422798661
0,0,7,24.193618432541083
0,1,0,8.416979301583458
0,1,1,-7.145576195094703
0,1,2,-17.22849690737691
0,1,3,-1.3896465758716323
0,1,4,17.346740376759847
0,2,0,96.54909603104043
0,2,1,120.94909603104
0,3,0,78.92291745988157
0,3,1,122.98836388777399
0,3,2,120.9438274612178
0,3,3,120.95699888577079
```

Conforme explicado por Mrode (2014), existe neste exemplo uma colinearidade no sistema de equações lineares que resulta desse modelo, isto é, o sistema admite diversas soluções, razão pela qual ele fixa o resultado para dois dos níveis do grupo de contemporâneos em zero. Como esse procedimento não foi executado neste exemplo, embora as soluções para os efeitos aleatórios coincidam com as apresentadas por Mrode (2014), a solução para os efeitos fixos é completamente diferente.

Exemplo 3. Modelo com Efeito Ambiental em Comum

O exemplo 3 (Mrode, 2014) é um modelo animal com efeitos ambientais em comum. O modelo animal considerado inclui o efeito genético aditivo, o sexo como efeito fixo e os grupos de irmãos completos como efeito ambiental em comum. A variância genética é igual a 20; a variância ambiental em comum é igual a 15 e a variância residual é igual a 65. A especificação do modelo utilizado encontra-se no arquivo ‘tests/data/mrode_4.2/conf.xml’, reproduzido a seguir.

```
<?xml version="1.0"?>
```

```

<brblup>

<dataset filename="tests/data/mrode_4.2/data.csv">

<!-- traits -->

<trait type="real" column="3" levels="1"/>

<!-- effects -->

<effect type="alpha" columns="0" levels="15"/>
<effect type="alpha" columns="1" levels="2"/>
<effect type="alpha" columns="2" levels="3"/>

</dataset>

<model>

<!-- one equation for each trait -->

<equation trait="0" effects="0,1,2"/>

<!-- residual covariances -->

<covariances>65</covariances>

<random effects="2" levels="3" type="identity">

<!-- permanent environmental covariances. -->

<covariances>15</covariances>

</random>

<random effects="0" levels="15" type="animal_additive" source="pedigree" filename="tests/data/
mrode_4.2/pedigree.csv" id="0" sire="1" dam="2">

<!-- genetic covariances. -->

<covariances>20</covariances>

</random>

</model>

</brblup>

```

Executando-se o comando brblup_app para resolver o modelo:

(venv) > brblup_app -i tests/data/mrode_4.2/conf.xml

Starting blup...

...

Number of steps to convergence: 13.

Done!

E utilizando o comando cat para verificar a solução:

(venv) > cat solution.csv

Trait,Effect,Level,Solution

0,0,0,-1.4407729468609805

0,0,1,-1.1748792270540613

0,0,2,1.4407729468589765

0,0,3,1.4407729468589765

0,0,4,-0.26589371980768445

0,0,5,-1.0975587761668333

0,0,6,-1.6670660225436438

0,0,7,-2.3337326892103083

0,0,8,3.9252560386479285

0,0,9,2.8947632850247413

0,0,10,-1.1414106280187368

0,0,11,1.5252560386479286

0,0,12,0.4478711755239906

0,0,13,0.5450305958138457

0,0,14,-3.818795491142676

0,2,0,-1.7623188405793995

0,2,1,2.161159420290212

0,2,2,-0.39884057970981973

0,1,0,91.49314009661892

0,1,1,75.76444444444502

Para obter e verificar o valor das acurárias, executa-se os comandos brblup_app e cat, conforme apresentado a seguir:

(venv) > brblup_app -i tests/data/mrode_4.2/conf.xml -p pev

Starting blup...

...

Number of steps to convergence: 13.

Done!

```
(venv) > cat accuracy.csv
```

Trait,Effect,Level,PEV,RELI,ACC,ACC_B

| Trait | Effect | Level | PEV | RELI | ACC | ACC_B |
|-------------------------------------------------------------------------------------|--------|-------|-----|------|-----|-------|
| 0,0,0,18.63768115942029,0.06811594202898552,0.260990310220486,0.034658579583878746 | | | | | | |
| 0,0,1,18.550724637681164,0.0724637681159418,0.2691909510290823,0.0369131753138463 | | | | | | |
| 0,0,2,18.637681159420286,0.06811594202898574,0.2609903102204864,0.03465857958387886 | | | | | | |
| 0,0,3,18.637681159420286,0.06811594202898574,0.2609903102204864,0.03465857958387886 | | | | | | |
| 0,0,4,18.63768115942029,0.06811594202898552,0.260990310220486,0.034658579583878746 | | | | | | |
| 0,0,5,17.43230273752013,0.12838486312399355,0.3583083352700486,0.06639669191031328 | | | | | | |
| 0,0,6,17.22244766505636,0.1388776167471819,0.3726628727780404,0.07203319927229179 | | | | | | |
| 0,0,7,17.22244766505636,0.1388776167471819,0.3726628727780404,0.07203319927229179 | | | | | | |
| 0,0,8,17.0063768115942,0.14968115942028992,0.38688649423350246,0.07787265490079409 | | | | | | |
| 0,0,9,16.91246376811594,0.1543768115942029,0.3929081465103554,0.08042227712618166 | | | | | | |
| 0,0,10,17.006376811594205,0.1496811594202898,0.3868864942335023,0.07787265490079398 | | | | | | |
| 0,0,11,17.0063768115942,0.14968115942028992,0.38688649423350246,0.07787265490079409 | | | | | | |
| 0,0,12,17.546602254428333,0.12266988727858341,0.35024261202569773,0.063340983750534 | | | | | | |
| 0,0,13,17.182544283413844,0.14087278582930784,0.37533023569825524,0.073108844485668 | | | | | | |
| 0,0,14,17.54660225442834,0.12266988727858297,0.35024261202569706,0.0633409837505342 | | | | | | |

Exemplo 4. Modelo Bivariado com Matrizes de Delineamento Idênticas

O exemplo 4 (Mrode, 2014) considera um modelo animal bivariado (ganho de peso antes da desmama e ganho de peso após a desmama), a mesma matriz de delineamento e nenhum valor faltante nos dados.

A covariância genética é dada pela matriz $\begin{pmatrix} 20 & 18 \\ 18 & 40 \end{pmatrix}$ e a covariância residual pela matriz $\begin{pmatrix} 40 & 11 \\ 11 & 30 \end{pmatrix}$

O sexo é considerado como efeito fixo. A especificação do modelo utilizado encontra-se no arquivo ‘tests/data/mrode_5.1/conf.xml’, reproduzido a seguir.

```
<?xml version="1.0"?>
<brblup>
<dataset filename="tests/data/mrode_5.1/data.csv">
<!-- traits -->
<trait type="real" column="2" levels="1"/>
<trait type="real" column="3" levels="1"/>
```

```

<!-- effects -->

<effect type="alpha" columns="0" levels="8"/>
<effect type="alpha" columns="1" levels="2"/>
</dataset>

<model>

  <!-- one equation for each trait -->
  <equation trait="0" effects="0,1"/>
  <equation trait="1" effects="0,1"/>
  <!-- residual covariances -->
  <covariances>
    40,11,
    11,30
  </covariances>

  <random effects="0" levels="8" type="animal_additive" source="pedigree" filename="tests/data/mrode_5.1/pedigree.csv" id="0" sire="1" dam="2">
    <!-- genetic covariances. -->
    <covariances>
      20,18,
      18,40
    </covariances>
  </random>
</model>
</brblup>

```

Para resolver o modelo e verificar a solução obtida, executa-se os comandos brblup_app e cat, conforme apresentado a seguir:

```
venv) > brblup_app -i tests/data/mrode_5.1/conf.xml
```

```
Starting blup...
```

```
...
```

```
Number of steps to convergence: 20.
```

```
Done!
```

```
(venv) > cat solution.csv
```

Trait,Effect,Level,Solution
 0,0,0,0.15091556731279213
 1,0,0,0.2795979727448686
 0,0,1,-0.015392509691481345
 1,0,1,-0.00761007081989948
 0,0,2,-0.07839189616418783
 1,0,2,-0.17034143859393955
 0,0,3,-0.010238958529291227
 1,0,3,-0.012670708624118362
 0,0,4,-0.27033144138923276
 1,0,4,-0.4778302616027221
 0,0,5,0.2758082575805722
 1,0,5,0.5172383870383891
 0,0,6,-0.3161175616394344
 1,0,6,-0.4789836950550802
 0,0,7,0.24375552300540235
 1,0,7,0.391961542524088
 0,1,0,4.360866999054443
 1,1,0,6.799897620385038
 0,1,1,3.3972615919043303
 1,1,1,5.880295937282168

E para obter e verificar o valor das acurárias, executa-se os comandos brblup_app e cat, conforme apresentado a seguir:

```
(venv) > brblup_app -i tests/data/mrode_5.1/conf.xml -p yes
```

Starting blup...

...

Number of steps to convergence: 20.

Done!

```
(venv) > cat solution.csv
```

Trait,Effect,Level,PEV,RELI,ACC,ACC_B

0,0,0,18.604734724528388,0.06976326377358055,0.26412736278844823,0.0355121896952665
 1,0,0,35.9017858552653,0.10245535361836744,0.32008647834353676,0.05261167075922213
 0,0,1,19.596375145064968,0.020181242746751593,0.14206070092306172,0.010142051982584
 1,0,1,38.76763091270867,0.030809227182283294,0.17552557415454675,0.0155251283970134
 0,0,2,17.89288621510458,0.10535568924477112,0.3245854113246175,0.05414360986710631
 1,0,2,33.79924377123426,0.15501890571914356,0.3937244032558098,0.08077146787055378
 0,0,3,16.504656253463335,0.17476718732683327,0.4180516562900251,0.09157674365240587
 1,0,3,29.724916676179664,0.2568770830955084,0.5068304283441439,0.1379542257487184
 0,0,4,16.541066394458973,0.1729466802770514,0.4158685853452403,0.09057528089294353
 1,0,4,29.864617768377705,0.2533845557905574,0.5033731774643514,0.1359308799584129
 0,0,5,17.15145004017563,0.14242749799121857,0.37739567828900555,0.07394789454978212
 1,0,5,31.50325489326588,0.21241862766835307,0.4608889537278509,0.1125421856044948
 0,0,6,17.115039899179994,0.14424800504100033,0.3797999539770909,0.0749313566231965
 1,0,6,31.36355380106786,0.2159111549733035,0.46466240968395917,0.11451208645928057
 0,0,7,16.284382165937338,0.18578089170313317,0.43102307560400194,0.0976590952988627
 1,0,7,29.159492251698026,0.2710126937075493,0.5205887952189803,0.14619246531056507

Exemplo 5. Modelo Bivariado com Dados Faltantes

O exemplo 5 (Mrode, 2014) ilustra a utilização de conjuntos de dados contendo valores faltantes, uma situação muito frequente em casos reais, utilizando o mesmo modelo descrito no exemplo 4. A especificação do modelo utilizado encontra-se no arquivo ‘tests/data/mrode_5.3/conf.xml’, sendo praticamente igual à especificação do exemplo 4 (‘tests/data/mrode_5.1/conf.xml’). Para resolver o modelo e verificar a solução obtida, executa-se os comandos brblup_app e cat, conforme apresentado a seguir:

```
(venv) > brblup_app -i tests/data/mrode_5.3/conf.xml
```

```
Starting blup...
```

```
...
```

```
Number of steps to convergence: 22.
```

```
Done!
```

```
(venv) > cat solution.csv
```

```
Trait,Effect,Level,Solution
```

0,0,0,0.15391435725408908
 1,0,0,0.28768817973478733
 0,0,1,-0.058867493200998995
 1,0,1,-0.05298074388089932
 0,0,2,-0.061486402953056324
 1,0,2,-0.16263757990575362
 0,0,3,0.02661648697699282
 1,0,3,0.035739305945193305
 0,0,4,-0.3065192596321517
 1,0,4,-0.5205479414571477
 0,0,5,0.23499825038066455
 1,0,5,0.47711175360986413
 0,0,6,-0.280116101361107
 1,0,6,-0.45228344320235997
 0,0,7,0.27161183231587066
 1,0,7,0.40733828651012205
 0,0,8,0.07696095546264846
 1,0,8,0.050619202640622606
 0,1,0,4.327295927356084
 1,1,0,6.792559621375322
 0,1,1,3.5981866845962784
 1,1,1,5.96638529341554

Exemplo 6. Modelo Bivariado com Matrizes de Delineamento Distintas

O exemplo 6 (Mrode, 2014) é construído a partir do exemplo 2 utilizando o parto para definir diferentes características e correspondentes efeitos fixos. A especificação do modelo utilizado encontra-se no arquivo ‘tests/data/mrode_5.5/conf.xml’, reproduzido a seguir.

```

<?xml version="1.0"?>
<brblup>
```

```

<dataset filename="tests/data/mrode_5.5/data.csv">
  <!-- traits -->
  <trait type="real" column="3" levels="1"/>
  <trait type="real" column="4" levels="1"/>
  <!-- effects -->
  <effect type="alpha" columns="0" levels="8"/>
  <effect type="alpha" columns="1" levels="2"/>
  <effect type="alpha" columns="2" levels="2"/>
</dataset>

<model>
  <!-- one equation for each trait -->
  <equation trait="0" effects="0,1"/>
  <equation trait="1" effects="0,2"/>
  <!-- residual covariances -->
  <covariances>
    65,27,
    27,70
  </covariances>
  <random effects="0" levels="8" type="animal_additive" source="pedigree" filename="tests/data/mrode_5.5/pedigree.csv" id="0" sire="1" dam="2">
    <!-- genetic covariances. -->
    <covariances>
      35,28,
      28,30
    </covariances>
  </random>
</model>
</brblup>

```

Para resolver o modelo e verificar a solução obtida, executa-se os comandos brblup_app e cat, conforme apresentado a seguir:

(venv) > brblup_app -i tests/data/mrode_5.5/conf.xml

Starting blup...

...

Number of steps to convergence: 20.

Done!

(venv) > cat solution.csv

Trait,Effect,Level,Solution

0,0,0,8.969159144235753

1,0,0,8.840288629084384

0,0,1,-2.9991427884765893

1,0,1,-2.7772802747189607

0,0,2,-5.9700163557589425

1,0,2,-6.0630083543652225

0,0,3,11.75424243135232

1,0,3,11.657587566163787

0,0,4,-16.25295661406713

1,0,4,-15.823507978242164

0,0,5,-17.31429689333369

1,0,5,-15.71912600307852

0,0,6,8.690473723986932

1,0,6,8.137644915234095

0,0,7,22.70213948329266

1,0,7,20.93068834076226

0,1,0,175.73126996363783

1,1,0,0.0

0,1,1,219.61329398892767

1,1,1,0.0

0,2,0,0.0

1,2,0,243.2390867421584

0,2,1,0.0

1,2,1,240.5497264663427

Exemplo 7. Modelo Bivariado sem Covariância Ambiental

O exemplo 7 (Mrode, 2014) ilustra o caso em que o modelo não contém covariância ambiental. Nele é pressuposto que os dados se referem a progêniés de 3 touros nascidos no mesmo rebanho e que a seleção considera simultaneamente mais de um objetivo de seleção, por exemplo corte e leite, de forma que machos e fêmeas são criados em regimes alimentares distintos. As características mensuradas são peso ao sobreano para machos e produção de gordura (no leite) para fêmeas. O grupo de contemporâneos (HYS) foi considerado como efeito fixo.

A covariância genética é dada pela matriz $\begin{pmatrix} 43 & 18 \\ 18 & 30 \end{pmatrix}$ e a covariância residual pela matriz $\begin{pmatrix} 77 & 0 \\ 0 & 70 \end{pmatrix}$

A especificação do modelo utilizado encontra-se no arquivo ‘tests/data/mrode_5.6/conf.xml’, reproduzido a seguir:

```
<?xml version="1.0"?>
<brblup>
  <dataset filename="tests/data/mrode_5.6/data.csv">
    <!-- traits -->
    <trait type="real" column="5" levels="1"/>
    <trait type="real" column="6" levels="1"/>
    <!-- effects -->
    <effect type="alpha" columns="0" levels="17"/>
    <effect type="alpha" columns="4" levels="3"/>
  </dataset>
  <model>
    <!-- one equation for each trait -->
    <equation trait="0" effects="0,1"/>
    <equation trait="1" effects="0,1"/>
    <!-- residual covariances -->
    <covariances>
      77,0,
      0,70
    </covariances>
    <random effects="0" levels="17" type="animal_additive" source="pedigree" filename="tests/data/mrode_5.6/pedigree.csv" id="0" sire="1" dam="2">
      <!-- genetic covariances. -->
      <covariances>
        43,18,
        18,30
      </covariances>
    </random>
  </model>
</brblup>
```

Para resolver o modelo e verificar a solução obtida, executa-se os comandos brblup_app e cat, conforme apresentado a seguir:

```
(venv) > brblup_app -i tests/data/mrode_5.6/conf.xml
```

```
Starting blup...
```

```
...
```

```
Number of steps to convergence: 38.
```

```
Done!
```

```
(venv) > cat solution.csv
```

```
Trait,Effect,Level,Solution
```

```
0,0,0,-3.364977689750998
```

```
1,0,0,1.2582392308949564
```

```
0,0,1,-1.4890900402853442
```

```
1,0,1,3.7737202271179457
```

```
0,0,2,4.23664590358582
```

```
1,0,2,-1.686971721623676
```

```
0,0,3,-6.939468136929375
```

```
1,0,3,-1.5714763672758116
```

```
0,0,4,-5.012200422105422
```

```
1,0,4,-2.0981304092480784
```

```
0,0,5,5.01220042210379
```

```
1,0,5,2.098130409260395
```

```
0,0,6,2.13678047224829
```

```
1,0,6,3.561300787083959
```

```
0,0,7,-4.273560944488825
```

```
1,0,7,-7.12260157417026
```

```
0,0,8,-12.161528636080673
```

```
1,0,8,-3.0907465451530403
```

```
0,0,9,-8.26284565330635
```

```
1,0,9,-1.2603355003192744
```

```
0,0,10,5.835811788286983
```

```
1,0,10,3.7763152293359243
```

0,0,11,12.632281535906166
 1,0,11,3.5577061046332603
 0,0,12,1.5226818634973933
 1,0,12,5.971070796071569
 0,0,13,-4.292018464946565
 1,0,13,-11.527388222061973
 0,0,14,-1.870220864154224
 1,0,14,0.010733706877908682
 0,0,15,4.290356856103344
 1,0,15,11.994997085756003
 0,0,16,2.6841136450570446
 1,0,16,1.6633828682002052
 0,1,0,412.2646235500865
 1,1,0,194.02892920392685
 0,1,1,276.2135169325098
 1,1,1,204.76619556815513
 0,1,2,0,0
 1,1,2,161.6629417124615

Diferente dos exemplos anteriores, os resultados obtidos neste exemplo diferem completamente daqueles apresentados por Mrode (2014). Para verificar a correção dos resultados obtidos, foram utilizadas duas estratégias distintas e que podem ser utilizadas pelo usuário para realizar suas próprias verificações. A primeira baseia-se na construção de um script Python para montagem ‘manual’ do sistema de equações, ou seja, primeiro construir as matrizes de incidência $X=[X_1 \ 0 \ 0 \ X_2]$ e $Z=[Z_1 \ 0 \ 0 \ Z_2]$, que são utilizadas para construir as matrizes $X'R-1X$, $X'R-1Z$, $Z'R-1XeZ'R-1Z+A-1G-1$, por multiplicação de matrizes, e os vetores $X'R-1ye$ $Z'R-1y$. A partir dessas matrizes, montar o MME e utilizar um solver padrão (ex: `scipy.sparse.linalg.cg`) para obter a solução. A segunda estratégia consiste em resolver o exemplo utilizando um software conhecido e confiável, tal como o BLUPF90 (Misztal et al., 2018). Em ambos os casos, para o presente exemplo, obteve-se o mesmo resultado que o apresentado acima, a menos de diferenças em função da precisão numérica.

Exemplo 8. Modelo Animal com Efeito Maternal

O modelo do exemplo 8 (Mrode, 2014) considera, como característica, o peso de bezerros ao nascimento, e como efeitos fixos o grupo de contemporâneos Heard-Year-Season (HYS) e o sexo do animal, para estimar os efeitos genéticos direto e maternal, considerando como efeitos ambientais permanentes as mães com mensurações da característica do modelo.

Considera-se a covariância genética igual a $\begin{pmatrix} 150 & -40 \\ -40 & 90 \end{pmatrix}$

a variância ambiental permanente igual a 40 e a variância residual igual a 350. A especificação do modelo encontra-se no arquivo ‘tests/data/mrode_6.1/conf.xml’, reproduzido a seguir.

```
<?xml version="1.0"?>
<brblup>
<dataset filename="tests/data/mrode_6.1/data.csv">
    <!-- traits -->
    <trait type="real" column="6" levels="1"/>
    <!-- effects -->
    <effect type="alpha" columns="0" levels="14"/>
    <effect type="alpha" columns="2" levels="14"/>
    <effect type="alpha" columns="3" levels="4"/>
    <effect type="alpha" columns="4" levels="3"/>
    <effect type="alpha" columns="5" levels="2"/>
</dataset>
<model>
    <!-- one equation for each trait -->
    <equation trait="0" effects="0,1,2,3,4"/>
    <!-- residual covariances -->
    <covariances>350</covariances>
    <random effects="2" levels="4" type="identity">
        <!-- permanent environmental covariances. -->
        <covariances>40</covariances>
    </random>
    <random effects="0,1" levels="14" type="animal_additive" source="pedigree" filename="tests/
data/mrode_6.1/pedigree.csv" id="0" sire="1" dam="2">
        <!-- genetic covariances: direct and maternal -->
        <covariances>
            150,-40,
            -40,90
        </covariances>
    </random>
</model>
```

```
</covariances>  
</random>  
</model>  
</brblup>
```

Para resolver o modelo e verificar a solução obtida, executa-se os comandos brblup_app e cat, conforme apresentado a seguir:

```
(venv) > brblup_app -i tests/data/mrode_6.1/conf.xml
```

```
Starting blup...
```

```
...
```

```
Number of steps to convergence: 36.
```

```
Done!
```

```
(venv) > cat solution.csv
```

```
Trait,Effect,Level,Solution
```

```
0,0,0,0.5638896776584965
```

```
0,1,0,0.26156519211801205
```

```
0,0,1,-1.2443893490784272
```

```
0,1,1,-1.5831613939253544
```

```
0,0,2,1.1649355595459936
```

```
0,1,2,0.735731974361402
```

```
0,0,3,-0.4844358881261305
```

```
0,1,3,0.5858642274459825
```

```
0,0,4,0.6295329689930905
```

```
0,1,4,-0.5074710759975832
```

```
0,0,5,-0.8586287129394125
```

```
0,1,5,0.8410405653044564
```

```
0,0,6,-1.1559681886594204
```

```
0,1,6,1.2993166238214362
```

```
0,0,7,1.9174040056638393
```

```
0,1,7,-0.15791481518988545
```

```
0,0,8,-0.5532626446847762
```

0,1,8,0.659541045923107
 0,0,9,-1.0550813684965323
 0,1,9,-0.1529535167873427
 0,0,10,0.3853576115616084
 0,1,10,0.9159579187932253
 0,0,11,0.8633170738777587
 0,1,11,0.44200779354890674
 0,0,12,-2.979573346194245
 0,1,12,0.0930557858155089
 0,0,13,1.7513016072403438
 0,1,13,0.3622134207830834
 0,2,0,-1.700831487435507
 0,2,1,0.4153974179768491
 0,2,2,0.8249150873284491
 0,2,3,0.4605189821303416
 0,3,0,14.492235259138104
 0,3,1,17.878690783939906
 0,3,2,15.926013538425746
 0,4,0,20.0477069859222
 0,4,1,13.198704343326792

O resultado obtido para os efeitos fixos é completamente diferente daquele reportado por Mrode (2014), o que é explicado pela existência de dependência entre eles⁴. Já os valores genéticos diretos e maternal, bem como o efeito ambiental permanente são idênticos aos apresentados por Mrode (2014), a menos de diferenças em função de precisão numérica.

Exemplo 9 Modelo com Covariáveis e Outros Efeitos Aleatórios

O exemplo 9 utiliza um conjunto de dados artificial para ilustrar a especificação de modelos que incluem efeitos fixos e covariáveis, além de outros efeitos aleatórios independentes dos efeitos genéticos aditivos direto e maternal. A especificação do modelo encontra-se no arquivo ‘tests/data/simulado_7/conf.xml’, reproduzido a seguir.

⁴ Existe um número infinito de soluções para os efeitos fixos. A solução apresentada neste documento e a apresentada em Mrode (2014), são apenas duas delas.

```
<?xml version="1.0"?>
<brblup>
<dataset filename="tests/data/simulado_7/data.csv">
    <!-- traits -->
    <trait type="real" column="9" levels="1"/>
    <trait type="real" column="10" levels="1"/>
    <!-- effects -->
    <effect type="alpha" columns="0" levels="35"/>
    <effect type="alpha" columns="2" levels="35"/>
    <effect type="alpha" columns="3" levels="3"/>
    <effect type="alpha" columns="4" levels="2"/>
    <effect type="alpha" columns="5" levels="4"/>
    <effect type="alpha" columns="6" levels="4"/>
    <effect type="real" columns="7" levels="1"/>
    <effect type="real" columns="8" levels="1"/>
</dataset>
<model>
    <!-- one equation for each trait -->
    <equation trait="0" effects="0,1,2,6,4"/>
    <equation trait="1" effects="0,1,3,7,5"/>
    <!-- residual covariances -->
    <covariances>
        6.3,2.8,
        2.8,9.4
    </covariances>
    <random effects="4" levels="4" type="identity">
        <!-- non genetic covariances. -->
        <covariances>3., 0.,
                    0., 0.
        </covariances>
    
```

```

</random>

<random effects="5" levels="4" type="identity">
    <!-- non genetic covariances. -->

    <covariances>0., 0.,
        0., 5.
    </covariances>

    </random>

    <random effects="0,1" levels="35" type="animal_additive" source="pedigree" filename="tests/
data/simulado_7/pedigree.csv" id="0" sire="2" dam="3">
        <!-- genetic covariances: direct and maternal -->

        <covariances>
            3. , 0.9,0.15,0.12,
            0.9 , 2.5,0.17,0.09,
            0.15,0.17, 1.5,0.08,
            0.12,0.09,0.08,1.75
        </covariances>
    </random>
</model>
</brblup>

```

Para resolver o modelo e verificar a solução obtida, executam-se os comandos brblup_app e cat, conforme apresentado a seguir:

```
(venv) > brblup_app -i tests/data/simulado_7/conf.xml
```

```
Starting blup...
```

```
....
```

```
Number of steps to convergence: 87.
```

```
Done!
```

```
(venv) > cat solution.csv
```

```
Trait,Effect,Level,Solution
```

```
0,0,0,0.27358312303707927
```

1,0,0,-0.23634707009834424

0,1,0,0.04829007052875544

1,1,0,-0.14002455250872436

0,0,1,-0.5924035066080465

1,0,1,0.34078400435385126

0,1,1,-0.42668915308259747

...

1,7,0,0.8047066858677184

Para verificar a correção dos resultados, pode-se utilizar as mesmas estratégias descritas na seção que apresenta o exemplo 7. No presente caso, utilizando-se a primeira delas, obtém-se o mesmo resultado apresentado acima, com diferenças em função da precisão numérica.

Exemplo 10 utilização de dados reais

Para demonstrar a utilização do software brblup em situações reais, neste exemplo ele é utilizado para avaliação genética utilizando um conjunto de dados real. Também é demonstrado a utilização do software brtools⁵, para auxílio na preparação de dados. O conjunto de dados utilizado, denominado Senepol1 possui 98.320 animais no pedigree e 16.842 observações de animais não identificados da raça Senepol. O modelo utilizado é bastante simplificado, animal univariado para característica peso ao sobreano com o grupo de contemporâneos como efeito fixo.

Inicialmente, o conjunto de dados é pré-processado utilizando o software auxiliar brtools. O brtools executa uma sequência de processamentos (tag <process>), comumente utilizada na preparação de dados para utilização pelo software brblup, previamente especificada em um arquivo de configuração (conf-brtools-0.xml), apresentado a seguir.

```
<?xml version="1.0"?>
<brtools>
  <!-- input: tests/data/senepol1/data.csv
      output: tests/data/senepol1/data.csv.sorted, pedigree.csv.sorted, map.csv
  -->
  <process name="sort" map="tests/data/senepol1/map.csv" suffix="sorted">
    <file name="tests/data/senepol1/pedigree.csv" type="pedigree" columns="0,1,2"/>
    <file name="tests/data/senepol1/data.csv" type="data" columns="0"/>
  </process>
```

⁵ Uma descrição mais detalhada das funcionalidades do software brtools encontra-se em documento à parte, em elaboração.

```

<!-- input: tests/data/senepol1/data.csv.sorted, pedigree.csv.sorted
      output: tests/data/senepol1/data.csv.sorted.proc0
-->

<process name="tasks" file="tests/data/senepol1/data.csv.sorted" suffix="proc0" ncols="7">
  <!-- dam column from pedigree inserted for maternal effect -->
  <task name="recode" columns="1"/><!--new column: recode gc_pd -->
</process>

```

Neste exemplo, é realizado um pré-processamento mínimo. Especificamente, uma sequência de dois processamentos é especificada:

1. Ordenamento do pedigree, visando gerar um sistema de equações mais compacto. Neste processo, os IDs dos animais nos arquivos de dados e pedigree são reatribuídos, tal que além dos novos arquivos correspondentes, um arquivo de mapeamento também é gerado.
2. Geração de uma nova coluna com os níveis do grupo contemporâneo (coluna 1) recodificados.

Para executar a sequência de pré-processamentos especificada, executa-se o comando br-tools_app, conforme apresentado a seguir:

```
(venv) > brtools_app -i tests/data/senepol1/conf-brtools-0.xml
```

Running sort...

Running tasks... preparing.

Preparing tasks...

Adding 1 column(s): Recoding.

... recoding column 1. Number of levels = 1061.

... running.

Done!

A especificação do modelo encontra-se no arquivo ‘conf-brblup-0.xml’, reproduzido a seguir.

```

<?xml version="1.0"?>
<brblup>
  <dataset filename="tests/data/data.csv.sorted.proc">
    <!-- traits -->
    <trait type="real" column="2" levels="1"/>
    <!-- effects -->

```

```

<effect type="alpha" columns="0" levels="3129"/>
<effect type="alpha" columns="3" levels="99"/>
</dataset>
<model>
  <!-- equation: pd = gc + ani -->
  <equation trait="0" effects="0,1"/>
  <!-- residual covariances -->
  <covariances>172</covariances>
  <random effects="0" levels="3129" type="animal_additive" source="pedigree" filename="tests/
  data/pedigree.csv.sorted" id="0" sire="1" dam="2">
    <!-- genetic covariances. -->
    <covariances>179</covariances>
  </random>
</model>
</brblup>

```

Para resolver o modelo, executa-se o comando brblup_app, conforme apresentado a seguir:

```
(venv) > brblup_app -i tests/data/senepol1/conf-brblup-1.xml
```

```
Starting blup...
```

```
...
```

```
Number of steps to convergence: 610 ...
```

```
Done!
```

Neste exemplo, a verificação da correção dos resultados é realizada por meio da segunda estratégia descrita na seção referente ao exemplo 7, ou seja, comparando os resultados com aqueles obtidos utilizando o software BLUPF90. Diferente dos exemplos anteriores, nesse caso a realização de uma comparação da solução por meio da comparação valor a valor não é possível, devido às dimensões do conjunto de dados. Dessa forma, optou-se por construir um gráfico do histograma das diferenças entre os resultados. Considerando que os resultados obtidos utilizando o software BLUPF90 estejam no arquivo ‘./solutions-blupf90.csv’ e os resultados obtidos utilizando o brlup estejam no arquivo ‘./solution.csv’, o script abaixo constrói um arquivo (‘./compara.csv’) com duas colunas, cada uma delas contendo os valores genéticos ordenados, obtidos utilizando cada um dos softwares. Os resultados obtidos para o efeito fixo foram ignorados na comparação.

```
(venv) > export LC_NUMERIC=en_US.utf8
(venv) > awk '{if ($2==2) print $4}' ./solutions-blupf90 | sort -g > sf1
(venv) > awk 'BEGIN{FNR > 1; FS=",,"} {if ($2==0) print $4}' ./solution.csv | sort -g > sf2
(venv) > paste -d"," sf1 sf2 > compara.csv
(venv) > rm sf1 sf2
```

O histograma das diferenças entre os valores das duas colunas do arquivo ‘./compara.csv’ é apresentado na Figura 1. Nele, observa-se que as diferenças concentram em torno de zero com a imensa maioria na terceira casa decimal (entre -0.005 e 0.005) e um pequeno número na segunda casa decimal (maior que 0.010), sugerindo que os resultados são idênticos a menos de diferenças de precisão numérica.

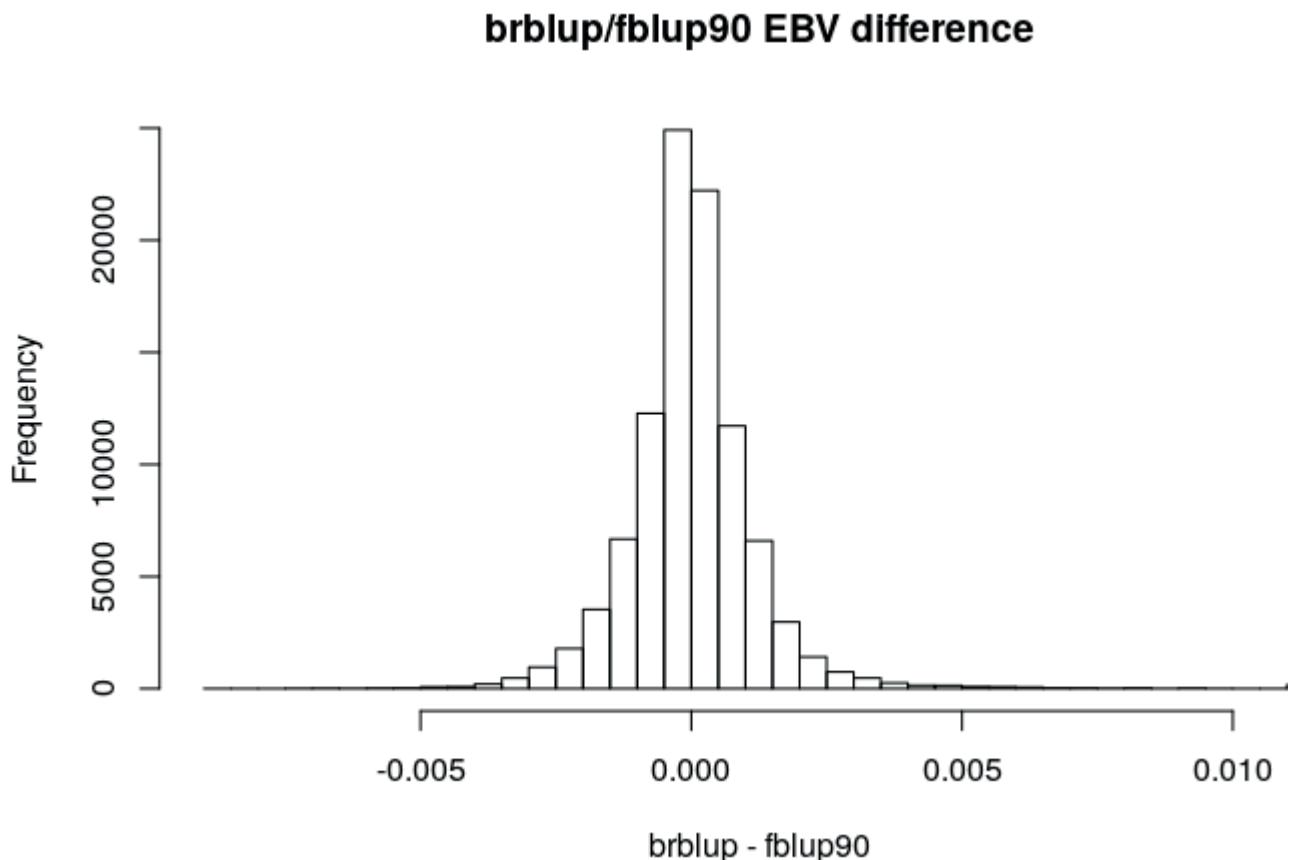


Figura 1. Distribuição da diferença valores genéticos preditos por brblup e BLUPF90

Especificamente para este exemplo que envolve uma quantidade de dados bem maior que os demais, a título de ilustração, também foi mensurado o tempo de execução da tarefa correspondente à leitura dos dados, montagem do MME em memória e execução das ~610 iterações para sua resolução, ou seja, excluindo a tarefa de preparação dos dados. Considerando um Desktop com arquitetura x86_64, 8 CPUs modelo Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz com 16GB de cache e sistema operacional GNI/Linux Ubuntu 18.04.3 LTS, a tarefa foi executada em 28 segundos.

Comentários Finais

O software brblup apresentado nas seções anteriores implementa funcionalidades para obtenção de

Atualmente, uma nova versão do brblup encontra-se em desenvolvimento no escopo do projeto BRBIFE (Silva, 2019) para inclusão de dados genômicos em seu procedimento de avaliação. Neste tutorial foram apresentados os procedimentos básicos de instalação do software brblup, bem como sua utilização, por meio de dez diferentes exemplos, em um ambiente de trabalho isolado do ambiente Python do sistema operacional. Para cada exemplo, para demonstrar a correção do resultado obtido, este foi comparado com uma solução alternativa. Indicações foram apresentadas para que o próprio usuário possa realizar suas verificações. Espera-se, com isso, que ao final do tutorial o usuário do brblup esteja habilitado a iniciar a sua utilização como ferramenta para avaliação genética de populações de animais.

Por fim, cabe observar que a versão do brblup a que se refere este tutorial foi desenvolvida para utilização no escopo do programa de melhoramento Embrapa/Geneplus e, portanto, é de uso restrito.

Uma versão para uso geral encontra-se atualmente em desenvolvimento.

Referências

MENEZES, G. R. de O. **Desenvolvimento e implementação de metodologias genético-estatísticas em avaliações genéticas de gado de corte**. Campo Grande, MS: Embrapa Gado de Corte, 2013. Projeto 02.13.14.004.00.00. Projeto concluído.

MENEZES, G. R. de O.; REGITANO, L. C. de A.; SILVA, M. V. G. B. da; CARDOSO, F. F.; SILVA, L. O. C. da; SIQUEIRA, F.; EGITO, A. A. do. Genômica aplicada ao melhoramento de gado de corte. In: ROSA, A. do N.; MARTINS. E. N.; MENEZES, G. R. de O.; SILVA, L. O. C. da (Ed.). **Melhoramento genético aplicado em gado de corte**: Programa Geneplus-Embrapa. Brasília, DF: Embrapa; Campo Grande, MS: Embrapa Gado de Corte, 2013. cap. 17, p. 213-225.

MISZTAL, I.; TSURUTA, S.; LOURENÇO, D.; MASUDA, Y.; AGUILAR, I.; LEGARRA, A.; VITEZI-CA, Z. **Manual for BLUPF90 family programs**. Athens: University of Georgia, 2018. Disponível em: <http://nce.ads.uga.edu/wiki/lib/exe/fetch.php?media=blupf90_all7.pdf>. Acesso em: 30 set. 2019.

MRODE, R. A. **Linear models for the prediction of animal breeding values**. 3. ed. Oxfordshire: CABI, 2014. 343 p.

SHEWCHUK, J. R. **An introduction to the conjugate gradient method without the agonizing pain**. Pittsburgh: Carnegie Mellon University, 1994. Technical report.

SILVA, L. O. C. **Otimização do fluxo genético em rebanhos sob seleção, de características econômicas relacionadas à eficiência alimentar, carcaça e reprodução, compondo índices de seleção e ferramental, visando ao melhoramento e uso da raça Nelore (BRBIFE)**. Campo Grande, MS: Embrapa Gado de Corte, 2019. Projeto: 20.18.01.013.00.00. 2019. Projeto em andamento.

บันสมประสังค์พ. Formation of a sparse bus impedance matrix and its application to short circuit study. **Engineering and Applied Science Research**, v. 7, n. 3, p. 16-29, 1980. Disponível em: <<https://www.tci-thaijo.org/index.php/easr/article/view/8330>>. Acesso em: 30 set. 2019.

