Supplementary material
{R, Echidna and Wombat codes}
for manuscript
Evaluation of alternative methods for estimating the precision of REML-based estimates of variance components and heritability

Contents

1. NPBS Scheme-1 Animal Model

1.1 R codes

```
# Create the necessary objects

fixed<-master$fi  # Read fixed vector for animals fi = yi - BLUP - ei
animal<-master$ai # Read vector of BLUP values
error<-master$ei  # Read vector of error values
dam<-master$Damid # Read family ids
common<-master[,c(-9:-12)] # Save the independent variables in the data frame separately
samp<- animal+error # Create the scheme-1 sampling vector for NPBS
sampdam<-as.data.frame(cbind(samp,dam)) # To be used on STEP: 1 below
b=as.data.frame(table(dam)) # To be used on STEP: 1 below

# Load the following libraries

library(sampling)
library(LaF)
library(progress)

### STEP:1 ACCESS THE FIRST ELEMENT OF EACH LEVEL OF FAMILY

first.element1 <- sampdam[!duplicated(sampdam$dam),]
first.element2 <- as.numeric(row.names(first.element1))
dam.unique <- unique(dam)
first.element3 <- data.frame(first.element2,dam.unique)
first.element4 <- merge(first.element3,b,by.x = "dam.unique",by.y = "dam",sort = F)

# Initiate the NPBS Scheme-1 algorithm for Animal Model

zz<-NULL
pb <- progress_bar$new(total = 10000)
system.time(for (i in 1:10000)
{
  pb$tick()

  ### STEP:2 GENERATE RANDOM LIST OF INDEX NUMBERS OF RANDOM EFFECTS
  ### Do it for BLUP

  n<-0
  z0 <- NULL
  for (i in 1:78)
  {
    n <- n + 1
    y <- first.element4$first.element2[n]
    x <- first.element4$Freq[n]
    z1 <- trunc(runif(x)*x+y)
    z0<-c(z0,z1)
  }

  ### STEP:3 RESAMPLE RANDOM EFFECTS BASED ON THE NUMBERS OBTAINED FROM STEP:2

  a <- 0
```

```
  z2 <- NULL
  for (i in 1:1413)
  {
    a <- a + 1
    z3 <- animal[z0[a]]
    z2 <- c(z2,z3)
  }

  ### STEP:2 GENERATE RANDOM LIST OF INDEX NUMBERS OF RANDOM EFFECTS
  ### Do it for Errors

  n<-0
  z0 <- NULL
  for (i in 1:78)
  {
    n <- n + 1
    y <- first.element4$first.element2[n]
    x <- first.element4$Freq[n]
    z1 <- trunc(runif(x)*x+y)
    z0<-c(z0,z1)
  }

  ### STEP:3 RESAMPLE RANDOM EFFECTS BASED ON THE NUMBERS OBTAINED FROM STEP:2

  a <- 0
  z4 <- NULL
  for (i in 1:1413)
  {
    a <- a + 1
    z3 <- error[z0[a]]
    z4 <- c(z4,z3)
  }

  ### STEP:4 GENERATE BOOTSTRAP DATA COMBINING RESAMPLED VALUES AND FIXED EFFECTS

  newy<-fixed + z2 + z4
  boot_dat<-cbind(common,newy)
  write.table(boot_dat,"boot_dat.dat",row.names=FALSE,col.names = FALSE)
  system("Echidna.exe magur.as",show.output.on.console = FALSE)
  z<-get_lines("magur.evp", c(4, 5, 9, 11))
  zz<-rbind(zz,z)
})
write.table(zz,"zzout.txt")
```

1.2 Echidna codes

```
Animal !P # !P specifies presence of pedigree as numerator relationship matrix
Sire 76 # No. of sires 76
Dam 78  # No. of dams 78
Stock 2 # No. of stocks 2
Batch 2 # No. of batch 2
Pond 11 # No. of ponds 11
Sex 2   # NO. of sex 2
```

```
 BW1     # Initial body weight as linear covariate
 newY    # Respose variable
magur.ped     # Pedigree file
boot_dat.dat  # Data file
newY ~ mu BW1 Stock Batch Pond Sex !r Animal # Model line
residual units # Residual variance
VPREDICT
F Total Residual + Animal    # Phenotypic variance
F Genetic Animal             # Additive variance
H Heritability Genetic Total # Heritability
```

1.3 Wombat codes

```
# specify the pedigree file to be used
PED E:\boot\boot_dat.d

# specify the data file and its layout
DAT E:\boot\boot_dat.d
  animal
  sire 76
  dam 78
  stock 2
  batch 2
  pond 11
  sex 2
  bw1
  newy
end

# Choose the type of analysis
ANAL UNI

# Specify the model to be fitted
MODEL
 RAN animal NRM
 FIX stock
 FIX batch
 FIX pond
 FIX sex
 COV bw1(1)
 TR newy
END

# Give starting values for variance components
VAR animal 1
255.72
VAR error 1
324.86
```

2. NPBS Scheme-1 Family Model

2.1 R codes

```
# Create the necessary objects

fixed<-master$fi  # Read fixed vector for animals fi = yi - BLUP - ei
animal<-master$ai # Read vector of BLUP values
error<-master$ei  # Read vector of error values
dam<-master$Damid # Read family ids
common<-master[,c(-9:-12)] # Save the independent variables in the data frame separately
sampanimal<-as.data.frame(cbind(animal,dam)) # Create 2 column data frame to be used in-
                                             #-Step:1 as an argument while re-sampling-
                                             #-BLUPS
samperror<-as.data.frame(cbind(error,dam)) # Create 2 column data frame to be used in-
                                           #-Step:2 as an argument while re-sampling-
                                           #-errors
b=as.data.frame(table(dam))

# Load the following libraries

library(sampling)
library(LaF)
library(progress)


# Initiate the NPBS Scheme-1 algorithm for Family Model

zz<-NULL
pb <- progress_bar$new(total = 10000)
system.time(for (i in 1:10000)
{
  pb$tick()

  ### STEP:1 Re-sample with replacement to create the vector of family effects

  a=sample(uni1que(dam))
  x=0
  z0<-NULL
  for (i in 1:78)
  {
    x=x+1
    z1<-sample(sampanimal$animal[sampanimal$dam==a[x]],
               size = b$Freq[b$dam==unique(sampanimal$dam)[x]],replace = TRUE)
    z0<-c(z0,z1)
  }

  ### STEP:2 Re-sample with replacement to create the vector of error effects

  c=sample(unique(dam))
  y=0
  z01<-NULL
  for (i in 1:78)
  {
```

```
    y=y+1
    z2<-sample(samperror$error[samperror$dam==c[y]],
                size = b$Freq[b$dam==unique(samperror$dam)[y]],replace = TRUE)
    z01<-c(z01,z2)
  }

  ### STEP:3 GENERATE BOOTSTRAP DATA COMBINING RESAMPLED VALUES AND FIXED EFFECTS

  newy<-fixed + z0 + z01
  boot_dat<-cbind(common,newy)
  write.table(boot_dat,"boot_dat.dat",row.names=FALSE,col.names = FALSE)
  system("Echidna.exe magur.as",show.output.on.console = FALSE)
  z<-get_lines("magur.evp", c(4, 5, 9, 11))
  zz<-rbind(zz,z)
})
write.table(zz,"zzout.txt")
```

2.2 Echidna codes

```
 Animal !I # !I species that no pedigree is available
 Sire 76 # No. of sires 76
 Dam 78   # No. of dams 78
 Stock 2 # No. of stocks 2
 Batch 2 # No. of batch 2
 Pond 11 # No. of ponds 11
 Sex 2   # NO. of sex 2
 BW1     # Initial body weight as linear covariate
 newY    # Respose variable
boot_dat.dat   # Data file
newY ~ mu BW1 Stock Batch Pond Sex !r Dam # Model line
residual units # Residual variance
VPREDICT
F Total Residual + Dam       # Phenotypic variance
F Genetic 2*Dam              # Additive variance
H Heritability Genetic Total # Heritability
```

2.2 Wombat codes

```
# specify the data file and its layout
DAT E:\boot\boot_dat.d
  animal
  sire 76
  dam 78
  stock 2
  batch 2
  pond 11
  sex 2
  bw1
  newy
end

# Choose the type of analysis
```

```
ANAL UNI

# Specify the model to be fitted
MODEL
 RAN dam
 FIX stock
 FIX batch
 FIX pond
 FIX sex
 COV bw1(1)
 TR newy
END

# Give starting values for variance components
VAR dam 1
255.72
VAR error 1
324.86
```

3. NPBS Scheme-2 Animal Model

3.1 R codes

```
# Create the necessary objects to initiate the algorithm

fixed<-master$fi  # Read fixed vector for animals fi = yi - BLUP - ei
animal<-master$ai # Read vector of BLUP values
error<-master$ei  # Read vector of error values
dam<-master$Damid # Read family ids
common<-master[,c(-9:-12)] # Save the independent variables in the data frame separately
samp<- animal+error # Create the scheme-1 sampling vector for NPBS
sampdam<-as.data.frame(cbind(samp,dam)) # To be used in STEP: 1 below
b=as.data.frame(table(dam)) # To be used on STEP: 1 below


#Load the following Libraries

library(sampling)
library(LaF)
library(progress)


### STEP:1 ACCESS THE FIRST ELEMENT OF EACH LEVEL OF FAMILY

first.element1 <- sampdam[!duplicated(sampdam$dam),]
first.element2 <- as.numeric(row.names(first.element1))
dam.unique <- unique(dam)
first.element3 <- data.frame(first.element2,dam.unique)
first.element4 <- merge(first.element3,b,by.x = "dam.unique",by.y = "dam",sort = F)


# Initiate the NPBS Scheme-2 algorithm for Animal Model

zz<-NULL
pb <- progress_bar$new(total = 10000)
system.time(for (i in 1:10000)
{
  pb$tick()

### STEP:2 GENERATE RANDOM LIST OF INDEX NUMBERS OF RANDOM EFFECTS

  n<-0
  z0 <- NULL
  for (i in 1:78)
  {
    n <- n + 1
    y <- first.element4$first.element2[n]
    x <- first.element4$Freq[n]
    z1 <- trunc(runif(x)*x+y)
    z0<-c(z0,z1)
  }

### STEP:3 RESAMPLE RANDOM EFFECTS BASED ON THE NUMBERS OBTAINED FROM STEP:2
```

```
  a <- 0
  z2 <- NULL
  for (i in 1:1413)
  {
    a <- a + 1
    z3 <- sampdam$samp[z0[a]]
    z2 <- c(z2,z3)
  }

### STEP:4 GENERATE BOOTSTRAP DATA COMBINING RESAMPLED VALUES AND FIXED EFFECTS

  newy<-fixed + z2
  boot_dat<-cbind(common,newy)
  write.table(boot_dat,"boot_dat.dat",row.names=FALSE,col.names = FALSE)
  system("Echidna.exe magur.as",show.output.on.console = FALSE)
  z<-get_lines("magur.evp", c(4, 5, 9, 11))
  zz<-rbind(zz,z)
})
write.table(zz,"zzout.txt")
```

3.2 Echidna codes

```
  Animal !P # !P specifies presence of pedigree as numerator relationship matrix
 Sire 76 # No. of sires 76
 Dam 78   # No. of dams 78
 Stock 2 # No. of stocks 2
 Batch 2 # No. of batch 2
 Pond 11 # No. of ponds 11
 Sex 2   # NO. of sex 2
 BW1     # Initial body weight as linear covariate
 newY    # Respose variable
magur.ped     # Pedigree file
boot_dat.dat  # Data file
newY ~ mu BW1 Stock Batch Pond Sex !r Animal # Model line
residual units # Residual variance
VPREDICT
F Total Residual + Animal    # Phenotypic variance
F Genetic Animal             # Additive variance
H Heritability Genetic Total # Heritability
```

3.3 Wombat codes

```
# specify the pedigree file to be used
PED E:\boot\boot_dat.d

# specify the data file and its layout
DAT E:\boot\boot_dat.d
  animal
  sire 76
  dam 78
  stock 2
  batch 2
```

```
  pond 11
  sex 2
  bw1
  newy
end

# Choose the type of analysis
ANAL UNI

# Specify the model to be fitted
MODEL
 RAN animal NRM
 FIX stock
 FIX batch
 FIX pond
 FIX sex
 COV bw1(1)
 TR newy
END

# Give starting values for variance components
VAR animal 1
255.72
VAR error 1
324.86
```

4. NPBS Scheme-2 Family Model

4.1 R codes

```r
# Create the necessary objects

fixed<-master$fi  # Read fixed vector for animals fi = yi - BLUP - ei
animal<-master$ai # Read vector of BLUP values
error<-master$ei  # Read vector of error values
dam<-master$Damid # Read family ids
common<-master[,c(-9:-12)] # Save the independent variables in the data frame separately
samp<- animal+error # Create the sampling vector combining BLUP and errors to be used-
                    #-in Step:3
sampdam<-as.data.frame(cbind(samp,dam)) # Create the two column data frame to be used-
                                        #-in Step:1
b=as.data.frame(table(dam)) # Create dam frequency table to be used as an argument in-
                            #-merge() in Step:1

# Load the following libraries

library(sampling)
library(LaF)
library(progress)

### STEP:1 ACCESS THE FIRST ELEMENT OF EACH LEVEL OF FAMILY

first.element1 <- sampdam[!duplicated(sampdam$dam),]
first.element2 <- as.numeric(row.names(first.element1))
dam.unique <- unique(dam)
first.element3 <- data.frame(first.element2,dam.unique)
first.element4 <- merge(first.element3,b,by.x = "dam.unique",by.y = "dam",sort = F)


# Initiate the NPBS Scheme-2 algorithm for Family Model

zz<-NULL
pb <- progress_bar$new(total = 10000)
system.time(for (i in 1:10000)
{
  pb$tick()

### STEP:2 GENERATE RANDOM LIST OF INDEX NUMBERS OF RANDOM EFFECTS

  n<-0
  z0 <- NULL
  for (i in 1:78)
  {
    n <- n + 1
    y <- first.element4$first.element2[n]
    x <- first.element4$Freq[n]
    z1 <- trunc(runif(x)*x+y)
    z0<-c(z0,z1)
  }
```

```
### STEP:3 RESAMPLE RANDOM EFFECTS BASED ON THE NUMBERS OBTAINED FROM STEP:2

  a <- 0
  z2 <- NULL
  for (i in 1:1413)
  {
    a <- a + 1
    z3 <- sampdam$samp[z0[a]]
    z2 <- c(z2,z3)
  }

### STEP:4 GENERATE BOOTSTRAP DATA COMBINING RESAMPLED VALUES AND FIXED EFFECTS

  newy<-fixed + z2
  boot_dat<-cbind(common,newy)
  write.table(boot_dat,"boot_dat.dat",row.names=FALSE,col.names = FALSE)
  system("Echidna.exe magur.as",show.output.on.console = FALSE)
  z<-get_lines("magur.evp", c(4, 5, 9, 11))
  zz<-rbind(zz,z)
})
write.table(zz,"zzout.txt")
```

4.2 Echidna codes

```
 Animal !I # !I species that no pedigree is available
 Sire 76 # No. of sires 76
 Dam 78  # No. of dams 78
 Stock 2 # No. of stocks 2
 Batch 2 # No. of batch 2
 Pond 11 # No. of ponds 11
 Sex 2   # NO. of sex 2
 BW1     # Initial body weight as linear covariate
 newY    # Respose variable
boot_dat.dat  # Data file
newY ~ mu BW1 Stock Batch Pond Sex !r Dam # Model line
residual units # Residual variance
VPREDICT
F Total Residual + Dam       # Phenotypic variance
F Genetic 2*Dam              # Additive variance
H Heritability Genetic Total # Heritability
```

4.3 Wombat codes

```
# specify the data file and its layout
DAT E:\boot\boot_dat.d
  animal
  sire 76
  dam 78
  stock 2
  batch 2
  pond 11
  sex 2
```

```
  bw1
  newy
end

# Choose the type of analysis
ANAL UNI

# Specify the model to be fitted
MODEL
 RAN dam
 FIX stock
 FIX batch
 FIX pond
 FIX sex
 COV bw1(1)
 TR newy
END

# Give starting values for variance components
VAR dam 1
255.72
VAR error 1
324.86
```

5. NPBS Scheme-3 Animal Model

5.1 R codes

```r
# Create the necessary objects to initiate the algorithm

fixed<-master$fi  # Read fixed vector for animals fi = yi - BLUP - ei
animal<-master$ai # Read vector of BLUP values
error<-master$ei  # Read vector of error values
dam<-master$Damid # Read family ids
common<-master[,c(-9:-12)] # save the independent variables in the data frame separately
samp<- animal+error # create the scheme-1 sampling vector for NPBS
sampdam<-as.data.frame(cbind(samp,dam)) # To be used on STEP: 1 below
b=as.data.frame(table(dam)) # To be used on STEP: 1 below

# Load the following libraries

library(sampling)
library(LaF)
library(progress)

### STEP:1 ACCESS THE FIRST ELEMENT OF EACH LEVEL OF FAMILY

first.element1 <- sampdam[!duplicated(sampdam$dam),]
first.element2 <- as.numeric(row.names(first.element1))
dam.unique <- unique(dam)
first.element3 <- data.frame(first.element2,dam.unique)
first.element4 <- merge(first.element3,b,by.x = "dam.unique",by.y = "dam",sort = F)

# Initiate the NPBS Scheme-3 algorithm for Animal Model

zz<-NULL
n<-1:78
pb <- progress_bar$new(total = 10000)
system.time(for (i in 1:10000)
{
  pb$tick()

  ### STEP:2 GENERATE RANDOM LIST OF INDEX NUMBERS OF RANDOM EFFECTS

  z01<-NULL
  q <- 0
  n <- sample(n)
  for (i in 1:78){
    q <- q + 1
    r <- n[q]
    x <- first.element4$Freq[r]
    y <- first.element4$first.element2[r]
    p <- first.element4$Freq[q]
    z <- sample(y:(y+(x-1)), size = p, replace = T)
    z01<-c(z01,z)
  }

  ### STEP:3 RESAMPLE RANDOM EFFECTS BASED ON THE NUMBERS OBTAINED FROM STEP:2
```

```
  a <- 0
  z4 <- NULL
  for (i in 1:1413)
  {
    a <- a + 1
    z3 <- sampdam$samp[z01[a]]
    z4 <- c(z4,z3)
  }

  ### STEP:4 GENERATE BOOTSTRAP DATA COMBINING RESAMPLED VALUES AND FIXED EFFECTS

  newy<-fixed + z4
  boot_dat<-cbind(common,newy)
  write.table(boot_dat,"boot_dat.dat",row.names=FALSE,col.names = FALSE)
  system("Echidna.exe magur.as",show.output.on.console = FALSE)
  z<-get_lines("magur.evp", c(4, 5, 9, 11))
  zz<-rbind(zz,z)
})
write.table(zz,"zzout.txt")
```

5.2 Echidna codes

```
 Animal !P # !P specifies presence of pedigree as numerator relationship matrix
 Sire 76 # No. of sires 76
 Dam 78   # No. of dams 78
 Stock 2 # No. of stocks 2
 Batch 2 # No. of batch 2
 Pond 11 # No. of ponds 11
 Sex 2   # NO. of sex 2
 BW1     # Initial body weight as linear covariate
 newY    # Respose variable
magur.ped     # Pedigree file
boot_dat.dat  # Data file
newY ~ mu BW1 Stock Batch Pond Sex !r Animal # Model line
residual units # Residual variance
VPREDICT
F Total Residual + Animal    # Phenotypic variance
F Genetic Animal             # Additive variance
H Heritability Genetic Total # Heritability
```

5.3 Wombat codes

```
# specify the pedigree file to be used
PED E:\boot\boot_dat.d

# specify the data file and its layout
DAT E:\boot\boot_dat.d
  animal
  sire 76
  dam 78
  stock 2
  batch 2
```

```
  pond 11
  sex 2
  bw1
  newy
end

# Choose the type of analysis
ANAL UNI

# Specify the model to be fitted
MODEL
 RAN animal NRM
 FIX stock
 FIX batch
 FIX pond
 FIX sex
 COV bw1(1)
 TR newy
END

# Give starting values for variance components
VAR animal 1
255.72
VAR error 1
324.86
```

6. NPBS Scheme-3 Family Model

6.1 R codes

```r
# Create the necessary objects

fixed<-master$fi  # Read fixed vector for animals fi = yi - BLUP - ei
animal<-master$ai # Read vector of BLUP values
error<-master$ei  # Read vector of error values
dam<-master$Damid # Read family ids
common<-master[,c(-9:-12)] # Save the independent variables in the data frame separately
samp<- animal+error # Create the sampling vector combining BLUP and errors to be used-
                    #-in Step:3
sampdam<-as.data.frame(cbind(samp,dam)) # Create the 2 column data frame to be used-
                                        #-in Step:1
b=as.data.frame(table(dam))# Create dam frequency table to be used as an argument in-
                           #-merge() in Step:1

# Load the following libraries

library(sampling)
library(LaF)
library(progress)

### STEP:1 ACCESS THE FIRST ELEMENT OF EACH LEVEL OF FAMILY

first.element1 <- sampdam[!duplicated(sampdam$dam),]
first.element2 <- as.numeric(row.names(first.element1))
dam.unique <- unique(dam)
first.element3 <- data.frame(first.element2,dam.unique)
first.element4 <- merge(first.element3,b,by.x = "dam.unique",by.y = "dam",sort = F)


#Initiate the NPBS Scheme-2 algorithm for Family Model

zz<-NULL
n<-1:78
pb <- progress_bar$new(total = 10000)
system.time(for (i in 1:10000)
{
  pb$tick()

  ### STEP:2 GENERATE RANDOM LIST OF INDEX NUMBERS OF RANDOM EFFECTS

  z01<-NULL
  q <- 0
  n <- sample(n)
  for (i in 1:78){
    q <- q + 1
    r <- n[q]
    x <- first.element4$Freq[r]
    y <- first.element4$first.element2[r]
    p <- first.element4$Freq[q]
    z <- sample(y:(y+(x-1)), size = p, replace = T)
```

```
    z01<-c(z01,z)
  }

  ### STEP:3 RESAMPLE RANDOM EFFECTS BASED ON THE NUMBERS OBTAINED FROM STEP:2

  a <- 0
  z4 <- NULL
  for (i in 1:1413)
  {
    a <- a + 1
    z3 <- sampdam$samp[z01[a]]
    z4 <- c(z4,z3)
  }

  ### STEP:4 GENERATE BOOTSTRAP DATA COMBINING RESAMPLED VALUES AND FIXED EFFECTS

  newy<-fixed + z4
  boot_dat<-cbind(common,newy)
  write.table(boot_dat,"boot_dat.dat",row.names=FALSE,col.names = FALSE)
  system("Echidna.exe magur.as",show.output.on.console = FALSE)
  z<-get_lines("magur.evp", c(4, 5, 9, 11))
  zz<-rbind(zz,z)
})
write.table(zz,"zzout.txt")
```

6.2 Echidna codes

```
 Animal !I # !I species that no pedigree is available
 Sire 76 # No. of sires 76
 Dam 78  # No. of dams 78
 Stock 2 # No. of stocks 2
 Batch 2 # No. of batch 2
 Pond 11 # No. of ponds 11
 Sex 2   # NO. of sex 2
 BW1     # Initial body weight as linear covariate
 newY    # Respose variable
boot_dat.dat  # Data file
newY ~ mu BW1 Stock Batch Pond Sex !r Dam # Model line
residual units # Residual variance
VPREDICT
F Total Residual + Dam       # Phenotypic variance
F Genetic 2*Dam              # Additive variance
H Heritability Genetic Total # Heritability
```

6.3 Wombat codes

```
# specify the data file and its layout
DAT E:\boot\boot_dat.d
  animal
  sire 76
  dam 78
  stock 2
```

```
  batch 2
  pond 11
  sex 2
  bw1
  newy
end

# Choose the type of analysis
ANAL UNI

# Specify the model to be fitted
MODEL
 RAN dam
 FIX stock
 FIX batch
 FIX pond
 FIX sex
 COV bw1(1)
 TR newy
END

# Give starting values for variance components
VAR dam 1
255.72
VAR error 1
324.86
```

7. parametric bootstrap

7.1 R codes

```
# Load the following libraries

library(LaF)
library(progress)

#Initiate the Parametric Bootstrap algorithm

zz<-NULL
pb <- progress_bar$new(total = 10000)
system.time(for(i in 1:10000)
{
  pb$tick()
  system("wombat.exe wombat.par",show.output.on.console = FALSE)
  system("wombat.exe wombat1.par",show.output.on.console = FALSE)
  z<-get_lines("SumEstimates.out", c(44, 55))
  zz<-rbind(zz,z)
})
write.table(zz,"zzout.txt")
```

7.2 Wombat codes - 1

```
# Parameter file-1
# This file will generate one simulated data set in each run
# Loop this file within R code to generate as many simulated data as possible

RUNOP --simul

# Specify the pedigree file to be used
PEDS E:/BootPara/BW5/boot.d

# Specify the data file and its layout
DATA E:/BootPara/BW5/boot.d
  animal
  sire 76
  dam 78
  stock 2
  batch 2
  pond 11
  sex 2
  nbw1
  BW5
end

# Choose the type of analysis
ANAL UNI

# Specify the model to be fitted
MODEL
 RAN animal NRM
```

```
 FIX stock
 FIX batch
 FIX pond
 FIX sex
 COV nbw1(1)
 TR BW5
END

# Give starting values for variance compoents
VAR animal 1
255.72
VAR error 1
324.86
```

7.3 Wombat codes - 2

```
# Parameter file-2
# This file is used for REML analysis of data produced by file-1
# Loop this file within R code

# Specify the pedigree file to be used
PEDS E:/BootPara/BW5/boot.d

# Specify the data file and its layout
DATA E:/BootPara/BW5/SimData001.dat
  animal
  sire 76
  dam 78
  stock 2
  batch 2
  pond 11
  sex 2
  nbw1
  BW5
end

# Choose the type of analysis
ANAL UNI

# Specify the model to be fitted
MODEL
 RAN animal NRM
 FIX stock
 FIX batch
 FIX pond
 FIX sex
 COV nbw1(1)
 TR BW5
END

# Give starting values for variance components
VAR animal 1
255.72
```

```
VAR error 1
324.86
```

8. Bayesian analysis

8.1 R codes

```
# For Animal model

 # Create the pedigree object

pedigree <- read.table("harvped.txt", header = T) # Import pedigree
data <- read.table("harv.txt", header = T) # Import data

# Load the followig library
library(MCMCglmm)

# Generate the inverse gamma prior

prior <- list(R = list(V=1, nu=0.002), G = list(G1 = list(V=1, nu=0.002)))

# Fit the Bayesian Animal model

model <- MCMCglmm(BW5 ~ 1+bw1+factor(stock)+factor(batch)+factor(pond)
                  +factor(sex), random = ~animal,family = "gaussian",prior = prior,
                  pedigree = data.frame(pedigree),data = data.frame(data),
                  nitt = 1000000,burnin = 50000, thin = 100)

# Get the Model Summary

summary(model)

# Extract variance components to obtain heritabiity

herit <- modelharvK1$VCV[, "animal"]/(modelharvK1$VCV[, "animal"] +
                                        modelharvK1$VCV[, "units"])

# Find the mean heritability
mean(herit)

# Find the effective sample size

effectiveSize(herit)
```

9 Asymptotic sampling

9.1 R codes

```
# For Animal model

RUNOP --sample
COM Simple univariate analysis

# Specify the pedigree file to be used
PEDS ../harv.d

# Specify the data file and its layout
DATA ../harv.d
  animal
  sire 76
  dam 78
  stock 2
  batch 2
  pond 11
  sex 2
  nbw1
  BW
end

# Choose the type of analysis
ANAL UNI

# Specify the model to be fitted
MODEL
 RAN animal NRM
 FIX stock
 FIX batch
 FIX pond
 FIX sex
 COV nbw1(1)
 TR BW
END

# Give starting values for variance compoents
VAR animal 1
255.72
VAR error 1
324.86
```

10. Download links for Echidna and Wombat

```
The software Wombat can be downloaded from:

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

The Echidna Mixed Model software can be downloaded from:

https://www.echidnamms.org/

The data set generated using different NPBS schemes is analysed with Wombat or Echidna
```