[8] studbookR: Quantitative genetics

Frank PG Princee

Draft 2016-12-29

Introduction

`studbookR` supports the following methods to estimate heritability (h^2) and repeatability (r_p):

- Parent-offspring regression
- Animal model - Restricted Maximum Likelihood
- Animal model - Markov chain Monte Carlo

Phenotypic traits

Data files that are created for natural history analyses by the *PML* software, include additional data that can be used in quantitative genetics i.e. estimation of heritability and repeatability (see vignette "[9] Data files").

The following phenotypic traits can be selected:

```
phenotype()

## 1: First age at reproduction
## 2: Fitness
## 3: Inter-birth interval
## 4: Lifespan
## 5: Litter day
## 6: Litter size
## 7: Litter sex-ratio
## 8: Litter survived

Selection: 6
```

These data can also be imported by calling specific functions e.g. `phenotype.firstAge()` or `phenotype.litter("SIZE")` (see help pages). The phenotypic data are stored in the global data frame `studbook.phenotype`.

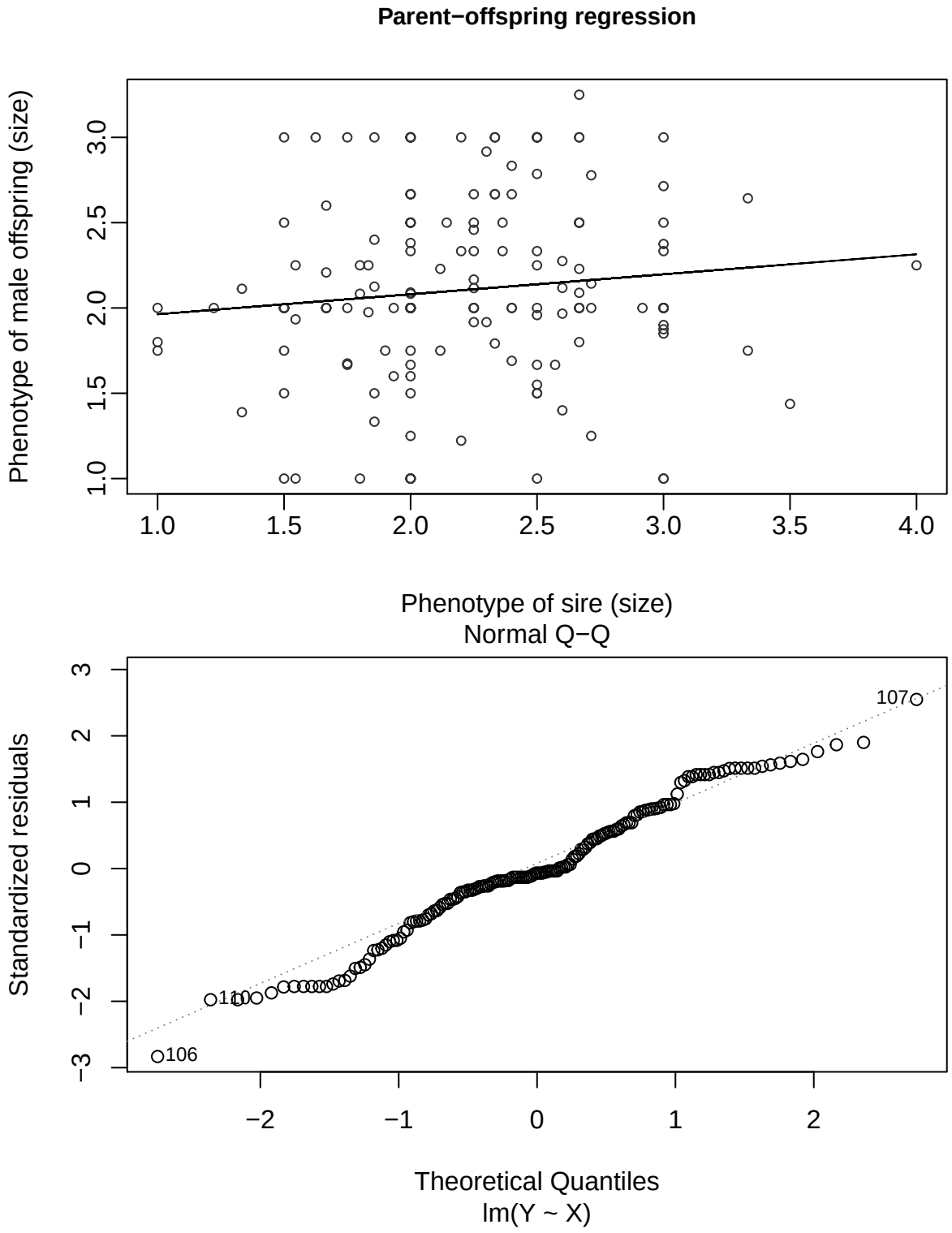
Parent-offspring regression

Parent-offspring (linear) regression can be applied as *midparent - offspring* or *single parent - offspring* regression. Offspring can be of single sex e.g. *sire - son* regression, as in the following example on litter size (phenotypic data already imported):

```
phenotype.regression(parent='sire',offspring='male') # Father-son regression
```

```
## ================================================
## Parent-offspring regression
## ================================================
##
## [ 2117 records imported ]
## Trait                 : SIZE
```

```
## Parent                : Phenotype of sire (size)
## Offspring             : Phenotype of male offspring (size)
## Offspring grouped     : TRUE
## Weighted family size  : TRUE
## Box-Cox test          : FALSE
## Lambda                : 1
## Phenotypic correlation: 0.8211847 (p=2.595743e-41) weighted
##
## Call:
## lm(formula = Y ~ X, data = data_set, weights = data_W)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -1.63700 -0.32150 -0.04123  0.41972  1.54378
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.84599    0.16429  11.236   <2e-16 ***
## X            0.11715    0.07116   1.646    0.102
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6094 on 162 degrees of freedom
## Multiple R-squared:  0.01645,    Adjusted R-squared:  0.01038
## F-statistic:  2.71 on 1 and 162 DF,  p-value: 0.1017
##
##                   2.5 %    97.5 %
## (Intercept)  1.52155593 2.1704230
## X           -0.02337456 0.2576674
## logLik: -122.0623    AIC: 250.1245 (df: 3 )
```




```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(data.fit)
## W = 0.97076, p-value = 0.001522
##
```

```
## [p < 0.05: non-normal distribution
##
## Durbin-Watson test for non-independence
##  lag Autocorrelation D-W Statistic p-value
##    1     0.004614682      1.967412   0.836
##  Alternative hypothesis: rho != 0
## [p < 0.05: no correlation between residuals]
##
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 3.698858    Df = 1     p = 0.05444971
## [p < 0.05: residuals are heteroscedastic]
##
##
## Bonferonni outlier test
##
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##      rstudent unadjusted p-value Bonferonni p
## 106 -2.896048          0.0043045      0.70593
## [p < 0.05: residual point is outlier]
##
## Potentially influential observations of
##   lm(formula = Y ~ X, data = data_set, weights = data_W) :
##
##     dfb.1_ dfb.X dffit   cov.r   cook.d hat
## 1   -0.38   0.34 -0.39_*  1.02    0.08   0.05_*
## 4   -0.04   0.04 -0.04    1.04_*  0.00   0.03
## 7    0.09  -0.08  0.09    1.08_*  0.00   0.06_*
## 10  -0.05   0.04 -0.07    1.05_*  0.00   0.04
## 14  -0.02   0.04  0.06    1.05_*  0.00   0.04_*
## 15   0.01  -0.01  0.01    1.04_*  0.00   0.03
## 40  -0.60   0.69  0.77_*  1.12_*  0.29   0.14_*
## 50   0.18  -0.21 -0.23    1.04    0.03   0.04_*
## 74  -0.09   0.11  0.13    1.06_*  0.01   0.05_*
## 106  0.78  -0.89 -0.97_*  1.02    0.45   0.10_*
## 107 -0.12   0.18  0.29    0.94_*  0.04   0.01
## 121  0.02  -0.02 -0.02    1.06_*  0.00   0.05_*
## 130  0.00   0.00  0.00    1.05_*  0.00   0.04
```

The Q-Q plot and various tests are used to determine whether the data are (more or less) normally distributed. Box-Cox transformation can be applied to "normalize" the data (see package help).

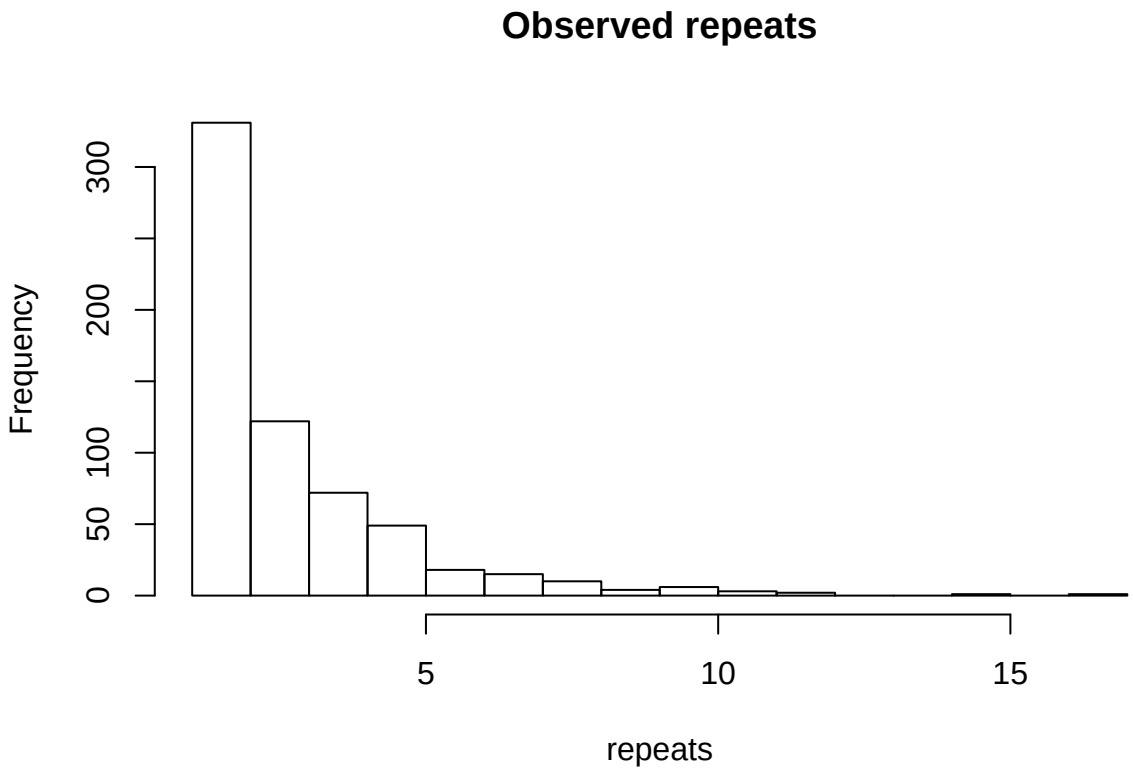
Repeatability

Repeatability in the selected phenotypic trait is estimated with an one-way ANOVA model, using the package `ICC`:

```
phenotype.repeatability()
```

```
## ================================================
## Repeatability in Litter
## ================================================
##
```

```
## Sex                     : all
## Trait                   : SIZE
## Minimum repeats         : 1
## Number of individuals   : 634
## Number of observations  : 1846
## Measurements per group  : 2.909
## Repeatability           : 0.09069 [0.04206, 0.1417]
```




```
## ------------------------------------------------
## [1] TRUE
```

Animal model

`studbookR` can estimate heritability, repeatibility and breeding values with the *animal model*. The packages `pedigreemm` and `MCMCglmm` are used to estimate these parameters following the Restrictive Maximum Likelihood (REML) and Markov chain Monte Carlo (MCMC) methods, respectively.

This model uses (full) pedigree data which needs to be provided in addition to the phenotypic data. The first three columns of the pedigree file must be conform the requirements of the `MCMCglmm` package i.e. `ID`, `DAM` and `SIRE`.

The pedigree data are imported with `pedigree.read()` and stored in the global data frame `studbook.pedigree`:

```
pedigree.read()
```

```
## [ 2117 records imported ]
```

```
tail(studbook.pedigree,3)
```

```
##        id  dam sire
## 2069 2069 1703 1739
## 2070 2070 1703 1739
## 2071 2071 1703 1739
```

The functions `phenotype()` and `pedigree.read` need to be called *before* the animal model is used. Alternatively, the function `animalmodel.init()` can be used to select and import phenotypic data, to import pedigree data.

REML

The package `pedigreemm` implements the *REML* method to estimate heritability. The function `animalmodel.reml()` serves as a wrapper around this package.

The `pedigreemm` functions require that numerical studbook IDs are used (`NA` for wild and unknown parents) in both phenotypic and pedigree data. Conversion of character based IDs, and restructuring of the pedigree data frame will be carried out automatically.

The basic model only includes individuals (*animal*) as a random effect:

```
animalmodel.reml()
```

```
## [ Reduced pedigree from 2117 to 639 records ]
## Model: SIZE ~ (1|animal)
## Time difference of 16.91943 secs
##
## Linear mixed model fit by REML ['lmerpedigreemm']
## Formula: formula(effects)
##    Data: studbook.phenotype
## Control:
## lme4::lmerControl(check.nobs.vs.nlev = "ignore", check.nobs.vs.nRE = "ignore")
##
## REML criterion at convergence: 4512.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.6875 -1.1138 -0.0827  1.0002  3.6803
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  animal   (Intercept) 0.03166  0.1779
##  Residual             0.64821  0.8051
## Number of obs: 1846, groups:  animal, 634
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  2.05751    0.03807   54.04
##
## VP               : 0.6798754
## heritability     : 0.04657088
## CV[A]            : 8.505333
##
## [Breeding values]
## Variance EBV     : 0.002896623
## Reliability EBV  : 0.09148463
```

```
## Prediction error : 0.02876577
## ---------------------------------------------------------------
## AIC              : 4518.129
## BIC              : 4534.692
## ---------------------------------------------------------------
```

The animal model can be extended with multiple fixed and random effects (which need to refer to column names in the phenotypic data). Random effects are entered as `(1|effect)`.

Multiple effect are entered as a text string in which the random effect of individuals `(1|animal)` need to be included. The following example, includes additonally sex as fixed effect and litter ID as random effect:

```
animalmodel.reml( "(1|animal) + SEX + (1|LITTER_ID)" )
```

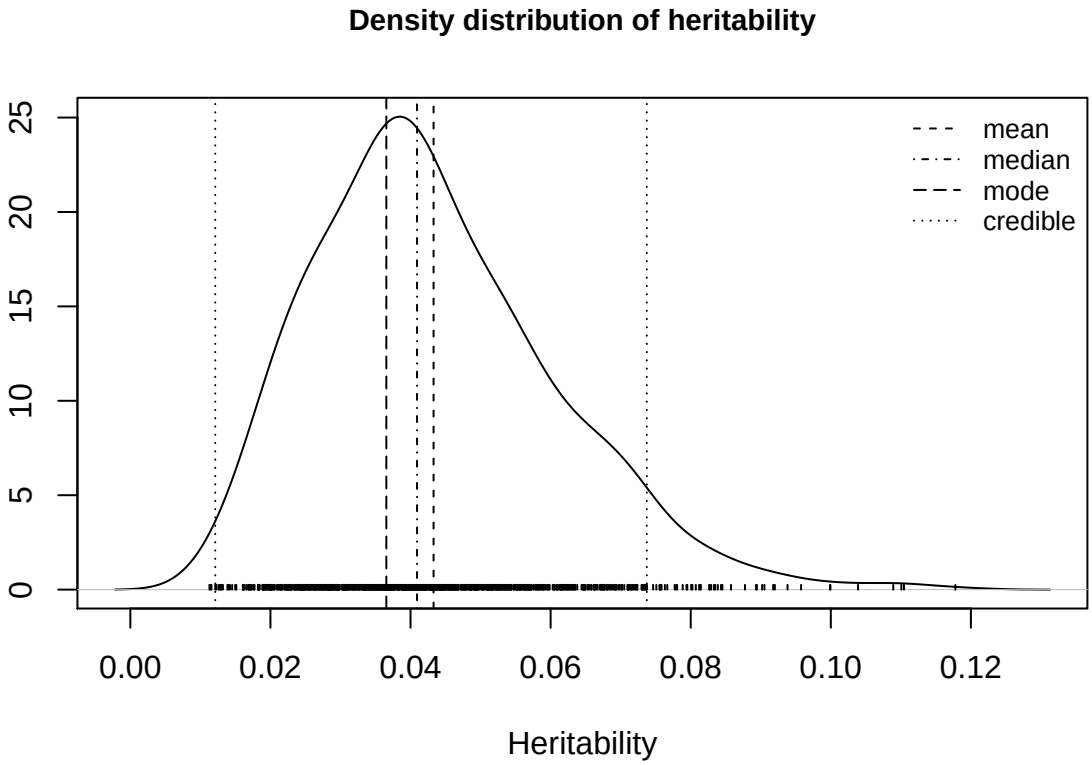
MCMC

The package `MCMCglmm` follows the Bayesian approach in statistics and applies MCMC techniques to fit generalised linear mixed models. The function `animalmodel.mcmc()` is a wrapper for this model. Note that both phenotypic and pedigree data need to be re-imported with `animalmodel.init()` when the REML model has been applied previously.

The basic model uses an *improper prior* and the default values for iterations, burnin' period and thinning, and restricts to the random individual (*animal*) effect.

```
animalmodel.mcmc()
```

```
## [ Reduced pedigree from 2117 to 639 records ]
##
## MCMCglmm animal model
## ===============================================================
##
## [Settings]
## ---------------------------------------------------------------
## Studbook                : animal
## Life history feature    : Litter
## Trait analysed          : SIZE
## Total observations      : 1846
## Number of individuals   : 634
## Mean value of trait     : 2.092091
## Variance (Vp[obs])      : 0.6798615
## Fixed effect(s)         : SIZE ~ 1
## Random effect(s)        : ~animal
## Prior model             : improper
## Trait distribution      : gaussian
## Iterations              : 13000
## Thinning                : 10
## Burn in                 : 3000
##
## [Results]
## ---------------------------------------------------------------
## Variances:
## VP                      : 0.6824227
## VA                      : 0.02960489
## VR                      : 0.6528178
```




```
## VA/VP[obs]               : 0.04354547
## Heritability - mean      : 0.04328748
##              - median    : 0.04092596
##              - mode      : 0.03654335
## Credible interval (95%) : 0.01211805 - 0.07373343
## Standard error           : 0.00153018
## z-ratio h2/se            : 28.28914
## CV_A                     : 8.224342 %
## DIC                      : 4501.472
## ------------------------------------------------------------------
```

The distribution pattern of heritability among iterations can be indicative of the performance of the selected prior and iteration settings. More diagnostic tools for MCMC are available through the (wrapper) function `animalmodel.diagnose()`.

Fixed and random effects, prior model and iterations are stored in memory. This facilitates to change a few setting model without re-entering all required data. These relevant functions are:

```
animalmodel.setFixed("SEX")                  # sex as fixed effect
animalmodel.setPrior("observed")             # prior based on observed phenotypic variance
animalmodel.setRandom("animal + LOCATION")   # individual and birth location
animalmodel.setRuns(25000,25,5000)           # iterations, thinning and burnin' period
```

Estimation of repeatability requires to create the data field `ID` which is a copy of the studbook ID`s in the field`animal(ID‘ is created automatically).

```
animalmodel.setFixed("1")                    # no fixed effects
animalmodel.setPrior("observed")             # prior based on observed phenotypic variance
animalmodel.setRandom("ID")                  # individual ID's
animalmodel.mcmc()
```

```
##
```

```
## MCMCglmm animal model
## ===================================================================
##
## [Settings]
## -------------------------------------------------------------------
## Studbook                : animal
## Life history feature    : Litter
## Trait analysed          : SIZE
## Total observations      : 1846
## Number of individuals   : 634
## Mean value of trait     : 2.092091
## Variance (Vp[obs])      : 0.6798615
## Fixed effect(s)         : SIZE ~ 1
## Random effect(s)        : ~ID
## Prior model             : improper
## Trait distribution      : gaussian
## Iterations              : 13000
## Thinning                : 10
## Burn in                 : 3000
##
## [Results]
## -------------------------------------------------------------------
## Variances:
## VP                      : 0.679147
## VI                      : 0.07179403
## VR                      : 0.607353
##
## Repeatability - mean    : 0.1056094
##               - median  : 0.1049769
##               - mode    : NA
## Credible interval (95%) : 0.06549399 - 0.1505302
## Standard error          : 0.001136465
## z-ratio r/se            : 92.92799
## DIC                     : 4468.625
## -------------------------------------------------------------------
```