Gaston FAQ

Version 1.5.5

Hervé Perdry

April 1, 2019

1 Which functions are multi-threaded?

For the moment, multithreading affects only `GRM` computations and matrix products.

2 Can I use Gaston with data non-human data, in particular with more than 22 autosomes?

Most functions don't care about X/Y. When they take into account whether a SNP is autosomal, X or Y linked, or mitochondrial, they use the values of the options ‘gaston.autosomes‘, ‘gaston.chr.x‘, ‘gaston.chr.y‘, ‘gaston.chr.mt‘ to determine (values can be modified using options).

Note that currently, nothing special is done for association testing with sexual chromosomes – this may change in the future.

3 Which allele is the effect allele in association tests?

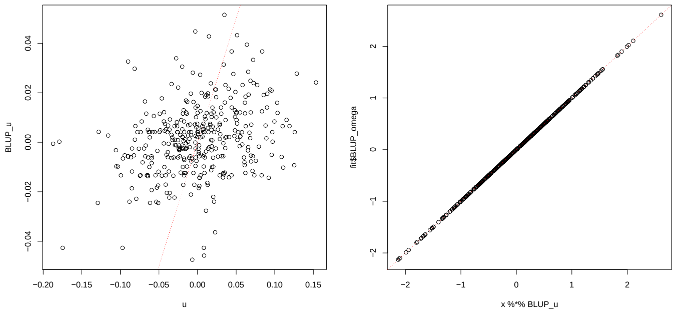
The effect allele is `A2`. This allele is left by Gaston as specified in the `bim` file read by `read.bed.matrix` ; in particular, nothing is done to ensure that one allele or the other is the minor allele.

4 Can I retrieve estimates of the SNP random effects after fitting a linear mixed model?

Yes, however the quality of the estimates is usually quite poor. Hereafter an example code to compute Best Linear Unbiased Predictors (BLUPs) of SNP effects. See *C. Dandine-Roulland, H. Perdry, The Use of the Linear Mixed Model in Human Genetics, 2015, Human Heredity 80:196-216* for some theoretical considerations.

```
> x <- as.bed.matrix(AGT.gen, AGT.fam, AGT.bim)
> standardize(x) <- "p"  # needed for matrix product below
```

```
>
> K <- GRM(x)
> set.seed(17);
> # SNP effects, drown in a normal distribution
> u <- rnorm( ncol(x), sd = sqrt(1/ncol(x)) );
> # Simulated phenotype
> y <- (x %*% u) + rnorm( nrow(x) , sd = 0.7)
> # fiting the linear model (note: above simulation is
> # done with tau = sigma2 = 1)
> fit <- lmm.diago(y, eigenK = eigen(K), verbose=FALSE )
> str(fit)
List of 9
 $ sigma2    : num 0.544
 $ tau       : num 0.686
 $ Py        : num [1:503] 1.3867 0.6467 1.2672 0.3935 0.0736 ...
 $ BLUP_omega: num [1:503] 0.517 0.372 0.77 0.513 -0.277 ...
 $ BLUP_beta : num 0.0174
 $ varbeta   : num [1, 1] 0.00108
 $ Xbeta     : num [1:503] 0.0174 0.0174 0.0174 0.0174 0.0174 ...
 $ varXbeta  : num -2.14e-18
 $ p         : int 0
> # retrieving BLUPs for u
> BLUP_u <- fit$tau * as.vector(fit$Py %*% x) / (ncol(x) - 1)
> # comparison with true effect values
> par(mfrow = c(1,2))
> plot(u, BLUP_u)
> abline(0, 1, col = "red", lty = 3)
> # these values allow to recompute the BLUP of omega
> plot(x %*% BLUP_u, fit$BLUP_omega)
> abline(0, 1, col = "red", lty = 3)
```



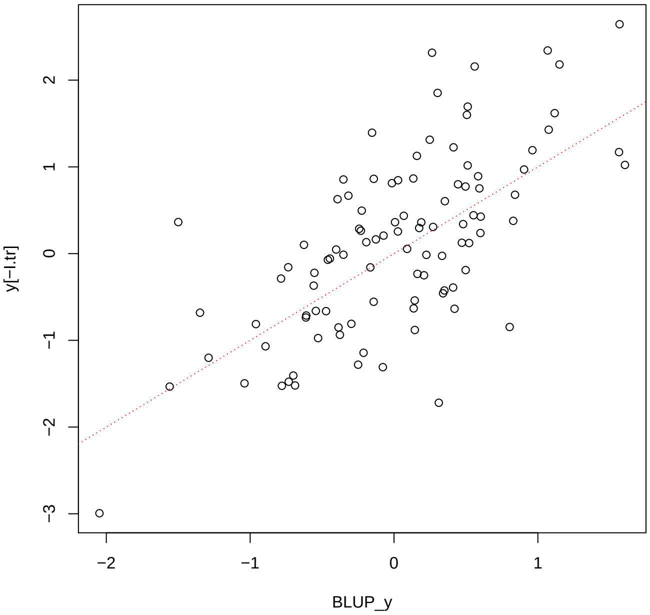
Note: If the number of individuals were (much) greater than the number of SNPs, the SNP effects estimates would be of good quality. Try to run the previous code with a 1000 × 5 random SNP matrix built as follows:

```
> x <- as.bed.matrix(matrix( rbinom(1000*5, 2, 0.5), ncol = 5))
> x@snps$chr <- 1  # needed to compute the GRM
```

5 Can I predict new phenotypes using a linear mixed model?

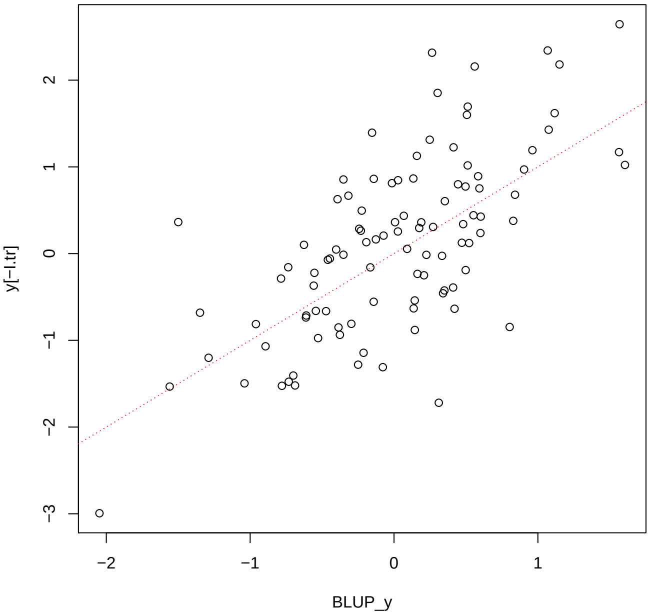
The previous question shows how to compute BLUPs for the SNP effects. They can be used in turn to predict new phenotypes. The only caveat is that you have to fix the `@p` slot of all samples you use to the same value. Indeed, this slot, which contains the frequency of alleles `A2`, is used for matrix standardization. Hereafter is some example code.

```
> x <- as.bed.matrix(AGT.gen, AGT.fam, AGT.bim)
> standardize(x) <- "p"
> p <-x@p # save the frequency of alleles A2
>
> set.seed(17);
> u <- rnorm( ncol(x), sd = sqrt(1/ncol(x)) );
> y <- (x %*% u) + rnorm( nrow(x) , sd = 0.7)
> # training set : 403 first individuals
> I.tr <- 1:403
> x.tr <- x[I.tr, ]
> x.tr@p <- p # use allele frequencies computed on whole sample
> K.tr <- GRM(x.tr)
> y.tr <- y[I.tr]
> fit <- lmm.diago(y.tr, eigenK = eigen(K.tr), verbose = FALSE)
> BLUP_u <- fit$tau * as.vector(fit$Py %*% x.tr) / (ncol(x.tr) - 1)
> # prediction on remaining individuals
> x1 <- x[-I.tr,]
> x1@p <- p # use same allele frequenciesa
> # predicted values for y
> BLUP_y <- x1 %*% BLUP_u
> # compare with simulated value
> plot(BLUP_y, y[-I.tr])
> abline(0, 1, col = "red", lty = 3)
```



A few matrix identities lead to an equivalent (and simpler) code avoiding the computation of the BLUPs of the SNP effects:

```
> K <- GRM(x)
> BLUP_y1 <- fit$tau * K[ -I.tr, I.tr ] %*% fit$Py
> plot(BLUP_y, y[-I.tr])
> abline(0, 1, col = "red", lty = 3)
```



Here again, see *C. Dandine-Roulland, H. Perdry, The Use of the Linear Mixed Model in Human Genetics, 2015, Human Heredity 80:196-216* and its supplementary for mixed model theory and some considerations on phenotype prediction.

6 How to use parametric bootstrap to determine confidence regions for ‘lmm‘ estimates?

Assume you have some data with GRM and a phenotype:

```
> x <- as.bed.matrix(AGT.gen, AGT.fam, AGT.bim)
> standardize(x) <- "p"  # needed for matrix product below
>
> K <- GRM(x)
> set.seed(17);
> # SNP effects, drawn in a normal distribution
> u <- rnorm( ncol(x), sd = sqrt(1/ncol(x)) );
> # Simulated phenotype
> y <- (x %*% u) + rnorm( nrow(x) , sd = 0.7)
```

You can analyze it with a linear mixed model:

```
> # fiting the linear model (note: above simulation is
> # done with tau = sigma2 = 1)
> fit <- lmm.diago(y, eigenK = eigen(K), verbose=FALSE )
```