QTL DETECTION WITH GS3.

A Legarra, 19/06/2013

Hélène Gilbert simulated a data set with 2 chromosomes and 5 QTLs at positions 20 cL, 73.14 cM on chromosome 1 and 20, 33.24 and 59.60 in chromosome 2. Each chromosome has 1 Morgan=100cM. The last QTL is fixed after the generations so we will focus on the first four. These positions correspond to SNPs number 1000,3657,6000,6662, 7980. There are 5000 markers in the 1st chromosome and 4997 in the second.

We know that the h^2=0.3 and the total variance 33.

The files are on `dga12:/home/alegarraalb/QTL_gs3` and `dga11:/home/alegarraalb/QTL_gs3` .

Copy the whole folder:

```
cp -r /home/alegarraalb/QTL_gs3 ./
```

Data reordering

GS3 has a file format shared with blupf90 programs but different from the LDSO or QTLmap. I put here a reminder of what I did to her original data files, but this must not be done in the exercise:

Treatment data from LDSO:

a) Reorder SNPs:
```
alegarraalb@dga12:/prodanr/alegarraalb/QTL_gs3# ./ldso2gs3.awk
genotypes1 > genotypes1.uga
```
b) Add overall mean
```
awk 'NR>1 {print 1,$0}' sim_ped_perf > sim_ped_perf_mean
```

Data analysis

Parameter files, variance components

Using GS3 correctly needs a few steps before proceeding blindly to BayesCPi. The first one is to correctly set up the parameter files for a BLUP_SNP (or RR-BLUP, or GBLUP).
We will include a model with an overall mean + SNPs. No pedigree.

Rewrite parameter file `ldso_gs3_vce.par` so that we use the correct variances.
Total genetic variance = 33.2*0.3=10
This variance has to be split among all the SNPs :
Variance by SNP = vara = $\sigma_a^2 = \frac{10}{2\sum p_i q_i} = \frac{10}{2*1273.75} = 3.93\ 10^{-4}$
Residual variance=23
Set varg, varp to 0.

First I lance a VCE work to see if it estimates the correct variances:

```
./gs3_linux64bit_executable ldso_gs3_vce.par | tee ldso_gs3_vce.par.out
```

After it has finished, I verify using R:

```
> a=read.table("var",header=TRUE)
> summary(a)
      vara                 vard         varg          varp            vare      
 Min.   :0.0004295   Min.   :0   Min.   :0   Min.   :2.15   Min.   :15.78  
 1st Qu.:0.0032457   1st Qu.:0   1st Qu.:0   1st Qu.:2.15   1st Qu.:18.25  
 Median :0.0036667   Median :0   Median :0   Median :2.15   Median :18.97  
 Mean   :0.0036349   Mean   :0   Mean   :0   Mean   :2.15   Mean   :19.08  
 3rd Qu.:0.0041041   3rd Qu.:0   3rd Qu.:0   3rd Qu.:2.15   3rd Qu.:19.82  
 Max.   :0.0055413   Max.   :0   Max.   :0   Max.   :2.15   Max.   :26.01  
      pa_1        pd_1      X2varapqpi              lambda2    
 Min.   :1   Min.   :1   Min.   : 1.094   0.18171-315:1000  
 1st Qu.:1   1st Qu.:1   1st Qu.: 8.269                     
 Median :1   Median :1   Median : 9.341                     
 Mean   :1   Mean   :1   Mean   : 9.260                     
 3rd Qu.:1   3rd Qu.:1   3rd Qu.:10.455                     
 Max.   :1   Max.   :1   Max.   :14.117                     
> plot(a$X2varapqpi)
```

The plot shows that convergence is quite good. On the other hand, we find correct parameters. The genetic variance explained by all SNPs ("X2varapqpi") is almost 10, as expected.
Also, try to find in the output file `ldso_gs3_vce.par.out` several things:
-sum(p_i q_i), the frequencies of the markers
-how many polygenic markers
-the description of the model and of the priors for variances

BLUP_SNP

Now that we are sure that everything is correct, we can run a BLUP_SNP (also called RR-BLUP, GBLUP, etc) using parameter file `ldso_gs3_blup.par` . This is *very* fast. Then we can analyze the SNP solutions using R:

```
> a=read.table("solutions",header=T)
> summary(a)
> snps=subset(a,effect==2)
> plot(snps**)
> plot(snps$solution**2)
```

Apparently we can't find the QTLs:

```
points(1000,0,col="red",pch=19)
points(3657,0,col="red",pch=19)
points(6000,0,col="red",pch=19)
points(6662,0,col="red",pch=19)
```

BayesCPi

Now we can do the same using BayesCPi. This is parameter file `ldso_gs3_bayescpi.par`
For BayesCPi we need to fix the expected number of "causal" SNPs so that we will have a clear signal. A way to do this is to play with the prior distribution of this proportion. For instance we can set it to $Beta(10^8, 999\ 10^8)$. This means that, on expectation, $\pi = \frac{10^8}{10^8+999\ 10^8} = 0.001$, and the a priori variance of π is extremely small so that in practice is fixed to 0.001.

Then we need to explain that “all” the genetic variance is due to a small fraction of SNPs (1/1000), so the variance of each SNP effect must be larger by a factor of 1000.

This will be:
 vara= 10/((1/1000)* 2*sum(p_i q_i)) = 10*1000/(2*1273.75)=0.393
Variance by SNP = vara = $\sigma_a^2 = \frac{10}{\pi\, 2\sum p_i q_i} = \frac{10}{0.001*2*1273.75} = 3.93\ 10^{-4}$

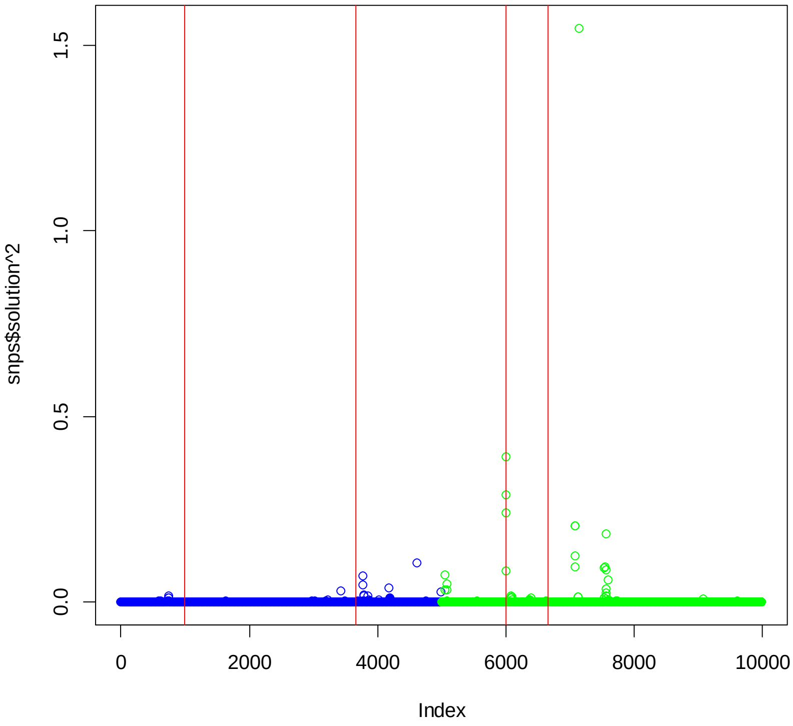
The parameter file has this modification and also the option “Use mixture TRUE “ to do BayesCPi.
I usually put a number of iterations of at least 10 times the number of SNPs; and 20% of them of burn-in. The “thin” has no importance for QTL mapping but it is better to put as thin interval (number of iterations)/1000 so we will have 1000 final samples of variance components (than one should check as before).
Today, your number of iterations should be much less (e.g., 10000) to have time.

Posterior analisis

After running we get this graph (I used 100,000 iterations but with 10,000 the graph is similar):

```
a=read.table("solutions ",header=T)
col=rep(c("blue","green"),each=5000)

# effects
snps=subset(a,effect==2)
plot(snps$solution**2,col=col)
abline(v=1000,col="red")
abline(v=3657,col="red")
abline(v=6000,col="red")
abline(v=6662,col="red")
```



So we capture very well QTL3, almost the 2nd , and a “parasite” (?) signal associated to the 4th.
We can also use “windows” of SNPs to partition the variance into chunks of consecutive SNPs. This is the segment mapping of Perez-Enciso and Varona (2000) re-discovered by Haley and colleagues in Roslin recently (“local heritability”).
It needs the allelic frequency of the SNPs, that is in file “`freq`” (created by GS3 as a byproduct). A script to plot this is in `postBayesCPi.r` , and it gives very similar graphs to the precedent one.

An analysis with something similar to p-values uses the Bayes Factor. The posterior probability of a SNP to be “real” is in the solutions file, in column “p”. Values close to 1 indicate that a SNP is always in the model. To declare if a SNP is “real” or not we can use the Bayes Factor. This is the ratio of “posterior” to “prior” probability of a SNP to be in the model, and it is in this case:

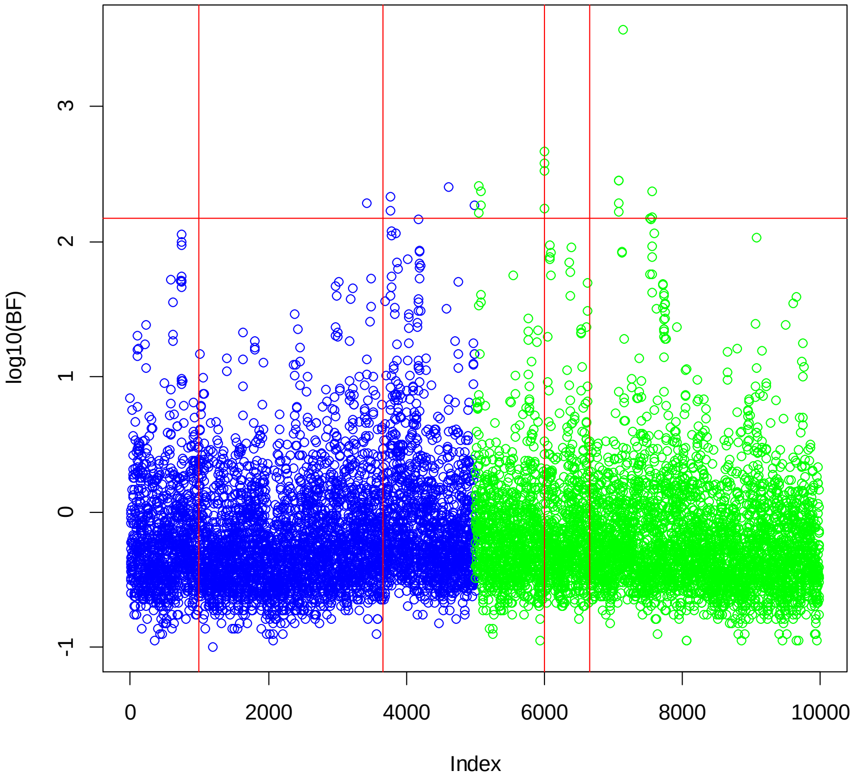
$$BF = \frac{\left(\frac{p}{1-p}\right)}{\left(\frac{\pi}{1-\pi}\right)} = \frac{\left(\frac{p}{1-p}\right)}{\left(\frac{1}{999}\right)}$$

in this case that the prior probabilities were 1 (of being included) to 999 (of being out). According to Vidal et al. (2005: http://www.journalofanimalscience.org/content/83/2/293.full), we choose this scale:

BF= 3-20	"suggestive"
BF= 20-150	"strong"
BF>150	"very strong"

The BF already accounts for multiple testing because we take all the SNPs at the same time. Let’s see the graph in logarithmic scale:

```
# BF, logarithmic scale
BF=999*a$p/(1-a$p)
plot(log10(BF),col=col)
abline(v=1000,col="red")
abline(v=3657,col="red")
abline(v=6000,col="red")
abline(v=6662,col="red")
# strong signal
abline(h=log10(150),col="red")
```



The horizontal bar is the BF>150 rejection threshold; there are 20 “Bayesianly significant” SNPs. The signal from QTL 2 and 3 is clearly captured. The signal in SNP 7138 is very strong yet there is no QTL there, this QTL might be in strong LD with the QTL.