Genome Wide Association Mapping using *gemma*

Genetic basis of colour-pattern polymorphism in *T. cristinae*

http://romainvilloutreix.alwaysdata.net/romainvilloutreix/workshop-material/

Víctor Soria-Carrasco
v.soria-carrasco@sheffield.ac.uk
PIP Zoology Fellow

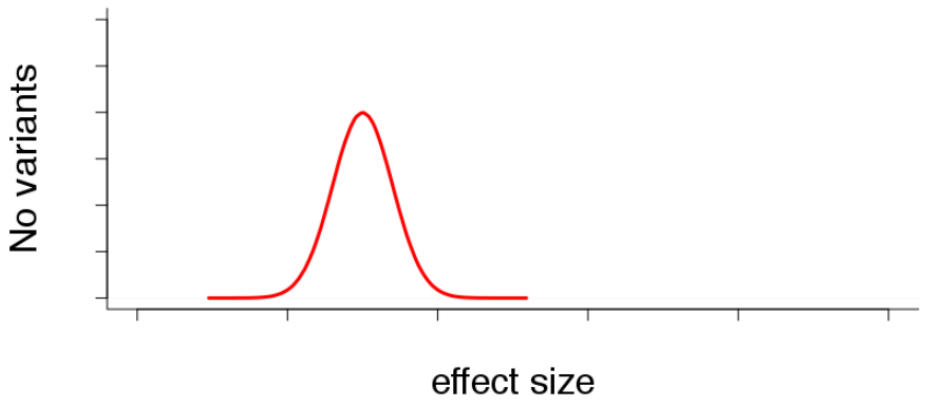
GWAS MULTI-SNP MODELS

Linear Mixed Model (LMM)

Assume polygenic basis:

all variants affect the phenotype

Effect sizes normally distributed

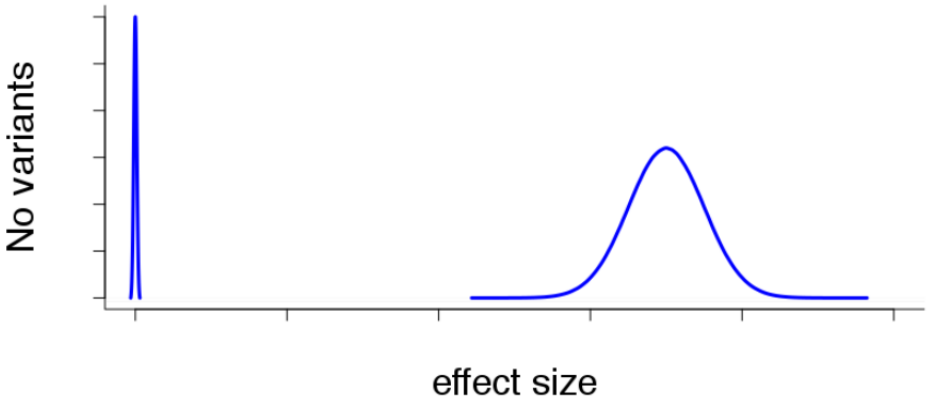


Bayesian Variable Selection Regression model (BVSR)

Assume mono/oligogenic basis:

a small proportion of variants affect the phenotype

Effect sizes as mixture of point mass at 0 and normal distribution



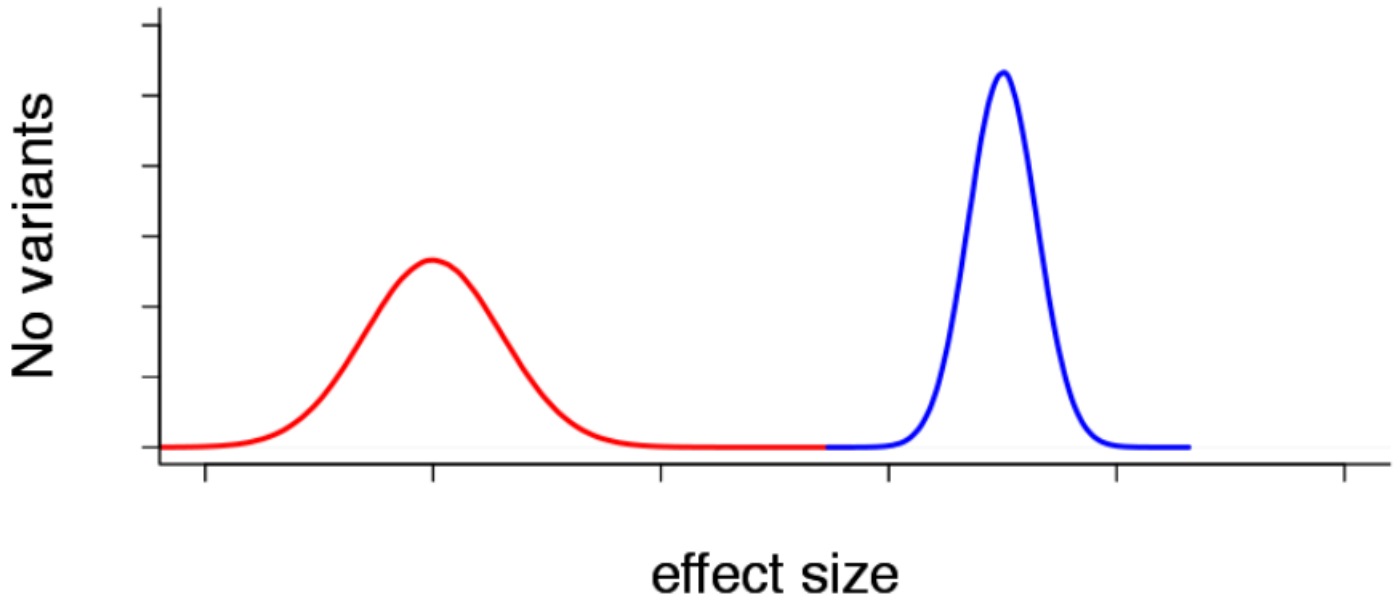
GWAS MULTI-SNP MODELS

Hybrid general model:
Bayesian Sparse Linear Mixed Model
(BSLMM)

Mixture of polygenic (LMM) and mono/oligogenic basis (BVSR)

Two distribution of effect sizes:
1) small effect size of all variants (α)
2) additional large effect size of some variants (β)

effect size of a given variant = $\alpha_i + \beta_i$



GEMMA

Genome-wide Efficient Mixed Model Association

Three models:

- Univariate Linear Mixed Model (LMM)
- Multivariate Linear Mixed Model (mvLMM)
- **Bayesian-Sparse Linear Mixed Model (BSLMM)**

Manual – read it!

www.xzlab.org/software/GEMMAmanual.pdf

Publications

- Xiang Zhou and Matthew Stephens (2012). Genome-wide efficient mixed-model analysis for association studies. Nature Genetics. 44: 821–824. http://goo.gl/pFb7Qy
- Xiang Zhou and Matthew Stephens (2014). Efficient multivariate linear mixed model algorithms for genome-wide association studies. Nature Methods. 11(4): 407–409. http://goo.gl/9pWM1Y
- **Xiang Zhou, Peter Carbonetto and Matthew Stephens (2013). Polygenic modeling with Bayesian sparse linear mixed models. PLoS Genetics. 9(2): e1003264. http://goo.gl/YStR2a**

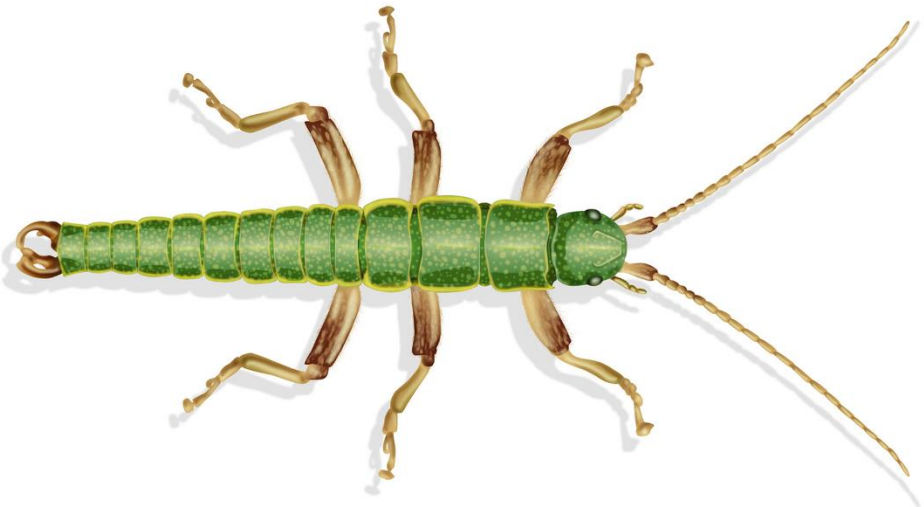
Also, BSLMM accounts for sample relatedness and population stratification


M. Muschick photo


A. Comeault photo

A *ADENOSTOMA* ECOTYPE

B *CEANOTHUS* ECOTYPE



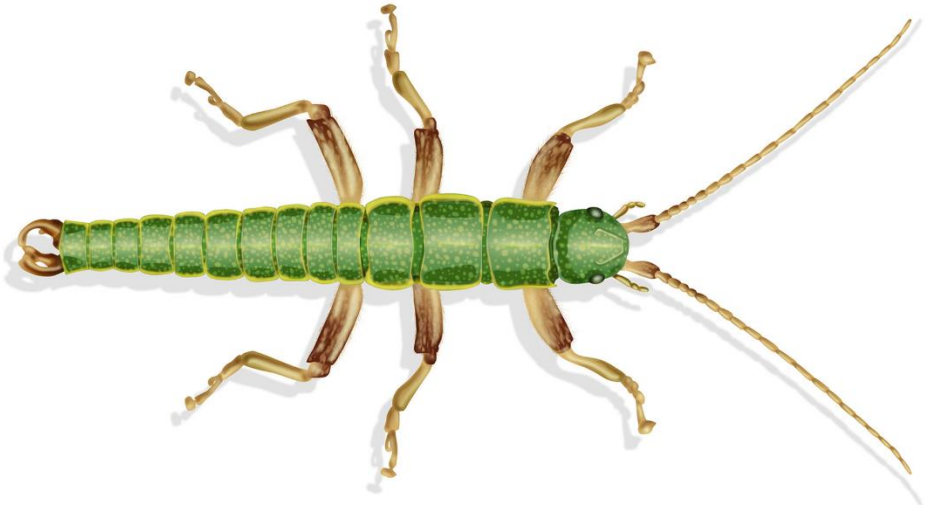

M. Muschick photo


M. Muschick photo

A *ADENOSTOMA* ECOTYPE

Genetic basis ?

B *CEANOTHUS* ECOTYP



Exercise steps:

1) Input files: phenotypic file – genetic file and how to generate it from a vcf file

2) Running *gemma*: generating relatedness matrix - setting a run

3) Running *gemma*: handling outputs

Exercise steps:

1) Input files: phenotypic file – genetic file and how to generate it from a vcf file

2) Running *gemma*: generating relatedness matrix - setting a run

3) Running *gemma*: handling outputs

Get data

```
# change to user data directory
$ cd /data/$USER/

# create directory
$ mkdir gwas_gemma

# copy scripts
$ cp -r /usr/local/extras/Genomics/workshops/March2016/gwas_gemma/scripts ./gwas_gemma/

# copy data
$ cp -r /usr/local/extras/Genomics/workshops/March2016/gwas_gemma/data ./gwas_gemma/

# copy results
$ cp -r /usr/local/extras/Genomics/workshops/March2016/gwas_gemma/results ./gwas_gemma/

$ cd gwas_gemma
```

Input files – genotypes (1)

Gemma needs an input file with the genotypes and another one with the phenotypes. For genotypes, we are going to use the mean genotype format based on BIMBAM, where genotypes are encoded as posterior mean genotypes. A posterior mean genotype is a value between 0 to 2 that can be interpreted as the minor allele dosage: 0 is homozygous for the major allele, 1 is a heterozygote, and 2 is a homozygote for the minor allele.

We are going to use a custom Perl script to calculate empirical mean genotypes from the genotype likelihoods in the VCF and using inferred allele frequencies to set Hardy-Weinberg Equilibrium priors ($p(AA) = p^2$; $p(aa) = (1-p)^2$; $p(Aa) = 2p(1-p)$)

```
# Convert VCF to BIMBAM using a custom Perl script

# have a look at the VCF file we are going to use
$ zcat data/fha.vcf.gz | less -S

# show help
$ ./scripts/bcf2bbgeno.pl -h

# Execute script
$ ./scripts/bcf2bbgeno.pl -i data/fha.vcf.gz -o fha.bbgeno -p H-W -s -r

# then compress the file
$ gzip fha.bbgeno
```

Input files – genotypes (2)

```
# This may take a while, it might be better to cancel it (ctrl+c),
# remove the unfinished output file if necessary:
$ rm fha.bbgeno*

# and instead have a look at the gemma input file in data/ (containing the
mean genotypes):

$ zcat data/fha.bbgeno.gz | less -S

    lg13_ord45_scaf428-158031 C T 0.01108 0.00279 0.00140 0.00140 0.01108
    lg13_ord45_scaf428-48027 T C 0.00299 0.00597 0.00150 0.00019 0.00597
    lg13_ord45_scaf428-80879 G A 0.00163 0.00163 0.00325 0.00041 0.00325
    lg13_ord45_scaf428-94107 G A 0.00358 0.00358 0.02771 0.01408 0.05530
    lg13_ord45_scaf428-158069 T A 0.16220 0.04246 0.02174 0.99552 0.16220
    lg13_ord45_scaf428-325672 A T 0.00784 0.00393 0.00197 0.00197 0.00784
    lg13_ord45_scaf428-466300 T G 0.00884 0.03430 0.03430 0.00014 0.03430
    lg13_ord45_scaf428-337230 G C 0.13098 0.03401 0.01734 0.13098 0.13098
```

Input files - phenotypes

The format for the phenotypes is very simple: a list of values in the same order than the samples in the genotypes file. We are going to use a single continuous trait in this exercise.

```
# Have a look at the phenotype file we are going to use

$ less -S data/fha.pheno

    0.866078916198945
    -1.17516992488642
    NA
    NA
    NA
    -3.11627693813939
    -0.348977465694598
```

Samples with missing phenotypes (NA) will be used for calculating the relatdness matrix (see next step), but will not be included in the BSLMM analysis.

Exercise steps:

1) Input files: phenotypic file – genetic file and how to generate it from a vcf file

2) Running *gemma*: generating relatedness matrix - setting a run

3) Running *gemma*: handling outputs

GEMMA

GEMMA is a complex piece of software with many options

```
# Have a look at the options

$ gemma -h

# we will be using the Bayesian sparse linear mixed model (BSLMM)
$ gemma -h 9

# and will calculate the relatedness matrix beforehand
$ gemma -h 8

# it may also be interesting to do some filtering
# (e.g. exclude rare variants with low minor allele frequency)
$ gemma -h 3
```

Calculate relatedness matrix (1)

```
# Open the script to calculate the relatedness matrix and edit
# orange text if need be

$ nano scripts/gemma_relmatrix.sh

    #!/bin/bash
    #$ -l h_rt=1:00:00
    #$ -j y
    #$ -o gemma_relmatrix.log

    GEMMA='gemma'
    DIR="/data/$USER/gwas_gemma"
    GENOTYPES='data/fha.bbgeno.gz'
    PHENOTYPES='data/fha.pheno'
    # centered matrix preferred in general, accounts better for population structure
    # standardized matrix preferred if SNPs with lower MAF have larger effects
    MATRIXTYPE=1 # 1=centered matrix, 2=standardized matrix
    OUTBASE='relmatrix'
```

Calculate relatedness matrix (2)

```
$ nano scripts/gemma_relmatrix.sh (cont.)


    hostname
    uname -a
    date
    echo "------------------------------------------------------"
    echo
    cd $DIR

    $GEMMA \
    -g $GENOTYPES \
    -p $PHENOTYPES \
    -gk $MATRIXTYPE \
    -o $OUTBASE
    echo
    echo "------------------------------------------------------"
    date



# Submit job to Iceberg:
$ qsub scripts/gemma_relmatrix.sh

# It should run in just a few minutes
```

Calculate relatedness matrix (3)

```
# Output files in output directory
$ ls output/

    relmatrix.log.txt -> log file
    relmatrix.cXX.txt -> relatedness matrix

# Have a look at the log
$ less -S output/relmatrix.log.txt

    ## Command Line Input = -g fha.bbgeno.gz -p fha.pheno -gk 1 -o relmatrix
    ##
    ## Summary Statistics:
    ## number of total individuals = 602
    ## number of analyzed individuals = 546
    ## number of covariates = 1
    ## number of phenotypes = 1
    ## number of total SNPs = 518232
    ## number of analyzed SNPs = 346660
    ##
    ## Computation Time:
    ## total computation time = 3.44783 min
    ## computation time break down:
    ##      time on calculating relatedness matrix = 2.11 min
```

Run BSLMM analysis (1)

```
# Open the script to fit BSLMM and edit orange text if need be

$ nano scripts/gemma_bslmm.sh

    #!/bin/bash
    #$ -l h_rt=07:00:00
    #$ -l rmem=2g
    #$ -l mem=4g
    #$ -j y
    #$ -o gemma_bslmm.log

    GEMMA='gemma'

    DIR="/data/$USER/gwas_gemma"
    GENOTYPES='data/fha.bbgeno.gz'
    PHENOTYPES='data/fha.pheno'
    RELMATRIX='output/relmatrix.cXX.txt'
    BSLMM=1 # 1=BSLMM, 2=standard ridge regression/GBLUP, 3=probit BSLMM (requires 0/1 phenotypes)
    OUTBASE='bslmm'
```

Run BSLMM analysis (2)

```
$ nano scripts/gemma_bslmm.sh (cont.)

    # priors
    # ---------------

    # h -> approximation to PVE: proportion of phenotypic variance
    explained by loci
    HMIN=0
    HMAX=1

    # rho -> approximation to PGE: proportion of genetic variance
    explained by sparse effect terms (~major effect loci)
    # rho=0 -> pure LMM, highly polygenic; rho=1 => pure BVSR, few loci
    RHOMIN=0
    RHOMAX=1

    # pi -> proportion of variants with non-zero effects (random + sparse
    effects)
    PIMIN=0
    PIMAX=1

    # gamma -> Number of variants with sparse effects (~ number of major
    effect loci)
    GAMMAMIN=0
    GAMMAMAX=300

    # ---------------
```

Run BSLMM analysis (3)

```
$ nano scripts/gemma_bslmm.sh (cont.)

    # proposals
    # ----------------
    # don't need to tweak them unless you have convergence problems

    GEOMMEAN=2000

    HSTEP=$(Rscript -e 'cat(min(c(10/sqrt('$NVARS'),1)))') # 0-1, default:
    min(10/sqrt(no_variants),1)

    RHOSTEP=$(Rscript -e 'cat(min(c(10/sqrt('$NVARS'),1)))') # 0-1,
    default: min(10/sqrt(no_variants),1)

    PISTEP=$(Rscript -e 'cat(min(c(5/sqrt('$NVARS'),1)))') # 0-1, default:
    min(5/sqrt(n),1)
    # ----------------
```

Run BSLMM analysis (4)

```
$ nano scripts/gemma_bslmm.sh (cont.)

    # chain parameters
    # ----------------

    BURNIN=250000 # No MCMC initial steps to be discarded (suggested: 10-
    25% MCCMC length)
    MCMCLEN=1000000 # No MCMC steps after burnin
    RECORDPACE=100 # Record states every X steps
    WRITEPACE=1000 # Write to file every X steps (suggested:
    >=MCMCLEN/1000)

    # ----------------



    # QC filters
    # ----------------

    MAF='0.01' # exclude very rare variants

    # ----------------
```

Run BSLMM analysis (5)

```
$ nano scripts/gemma_bslmm.sh (cont.)

    hostname
    uname -a
    date
    echo "------------------------------------------------------"
    echo
    cd $DIR

    $GEMMA \
    -g $GENOTYPES \
    -p $PHENOTYPES \
    -k $RELMATRIX \
    -bslmm $BSLMM \
    -w $BURNIN \
    -s $MCMCLEN \
    -rpace $RECORDPACE \
    -wpace $WRITEPACE \
    -maf $MAF \
    -o $OUTBASE

    echo
    echo "------------------------------------------------------"
    dat
```

Run BSLMM analysis (6)

```
$ nano scripts/gemma_bslmm.sh (cont.)

    # This is an example of how you can specify priors and proposals

    $GEMMA \
    -g $GENOTYPES \
    -p $PHENOTYPES \
    -k $RELMATRIX \
    -bslmm $BSLMM \
    -hmin $HMIN \
    -hmax $HMAX \
    -rmin $RHOMIN \
    -rmax $RHOMAX \
    -pmin $PIMIN \
    -pmax $PIMAX \
    -gmean $GEOMMEAN \
    -hscale $HSTEP \
    -rscale $RHOSTEP \
    -pscale $PISTEP \
    -w $BURNIN \
    -s $MCMCLEN \
    -rpace $RECORDPACE \
    -wpace $WRITEPACE \
    -maf $MAF \
    -o $OUTBASE
```

Run BSLMM analysis (7)

```
# Submit job to Iceberg queue

$ qsub scripts/gemma_bslmm.sh

# Run time should be around 15 min, you can have a look at
results/ if you don't want to wait

# Output files in output directory
$ ls output/
    bslmm.bv.txt -> posterior samples of breeding values(~estimated random
    effects)
    bslmm.gamma.txt -> posterior samples of gamma
    bslmm.hyp.txt -> posterior samples of hyperparameters
    bslmm.log.txt -> log file
    bslmm.param.txt -> posterior samples of parameters

# Have a look at the log, the hyperparameters, and the parameters

$ less -S output/bslmm.log.txt

$ less -S output/bslmm.param.txt

$ less -S output/bslmm.hyp.txt
```

Run BSLMM analysis (8)

```
$ less -S output/bslmm.log.txt
```

```
## GEMMA Version = 0.94
##
## Command Line Input = -g fha.bbgeno.gz -p fha.pheno -k output/relmatrix.cXX.txt -bslmm 1 -hmin 0 -hmax 1 -rmin 0 -
rmax 1 -pmin -5.53990
##
## Summary Statistics:
## number of total individuals = 602
## number of analyzed individuals = 546
## number of covariates = 1
## number of phenotypes = 1
## number of total SNPs = 518232
## number of analyzed SNPs = 346660
## REMLE log-likelihood in the null model = -737.564
## MLE log-likelihood in the null model = -737.31
## pve estimate in the null model = 0.889028
## se(pve) in the null model = 0.0499207
## vg estimate in the null model = 2.44228e-306
## ve estimate in the null model = 4.94066e-324
## beta estimate in the null model =
## se(beta) =
## estimated mean = 1.17936e-16
##
## MCMC related:
## initial value of h = 0.889028
## initial value of rho = 0.626463
## initial value of pi = 0.000865401
## initial value of |gamma| = 300
## random seed = 42003
## acceptance ratio = 0.15753
##
## Computation Time:
## total computation time = 19.5195 min
## computation time break down:
##      time on calculating relatedness matrix = 0 min
##      time on eigen-decomposition = 0.00733333 min
##      time on calculating UtX = 2.56967 min
##      time on mcmc = 14.5967 min
##      time on Omega = 6.4025 min
```

Run BSLMM analysis (9)

```
$ less -S output/bslmm.param.txt
```

```
chr     rs      ps      n_miss  alpha   beta    gamma
-9      lg13_ord45_scaf428-94107        -9      0       2.909899e-05    0.000000e+00    0.000000e+00
-9      lg13_ord45_scaf428-158069       -9      0       -1.401044e-05   0.000000e+00    0.000000e+00
-9      lg13_ord45_scaf428-466300       -9      0       1.450053e-05    0.000000e+00    0.000000e+00
-9      lg13_ord45_scaf428-337230       -9      0       2.330630e-05    0.000000e+00    0.000000e+00
```

```
$ less -S output/bslmm.hyp.txt
```

```
h       pve     rho     pge     pi      n_gamma
2.898484e-01    4.025551e-01    9.946412e-01    9.966603e-01    1.266772e-05    5
2.841103e-01    3.685523e-01    9.760376e-01    9.827668e-01    1.230484e-05    4
2.731768e-01    4.114452e-01    9.459294e-01    9.729468e-01    1.237195e-05    5
2.513739e-01    3.821418e-01    9.080483e-01    9.508555e-01    1.308855e-05    7
2.654772e-01    3.888621e-01    9.071505e-01    9.462713e-01    1.569321e-05    6
```

Exercise steps:

1) Input files: phenotypic file – genetic file and how to generate it from a vcf file

2) Running *gemma*: generating relatedness matrix - setting a run

3) Running *gemma*: handling outputs

Analysing BSLMM output (1)

We are going to summarize the posterior distributions of hyperparameters and parameters using in R.

```
# Open the script and copy and paste the commands line by line in R;
# change orange text as required

$ nano scripts/gemma_hyperparam.R

setwd("/data/$USER/gwas_gemma/output")


# Load hyperparameters
# ================================================================================
hyp.params<-read.table("bslmm.hyp.txt",header=T)
# ================================================================================
```

Analysing BSLMM output (2)

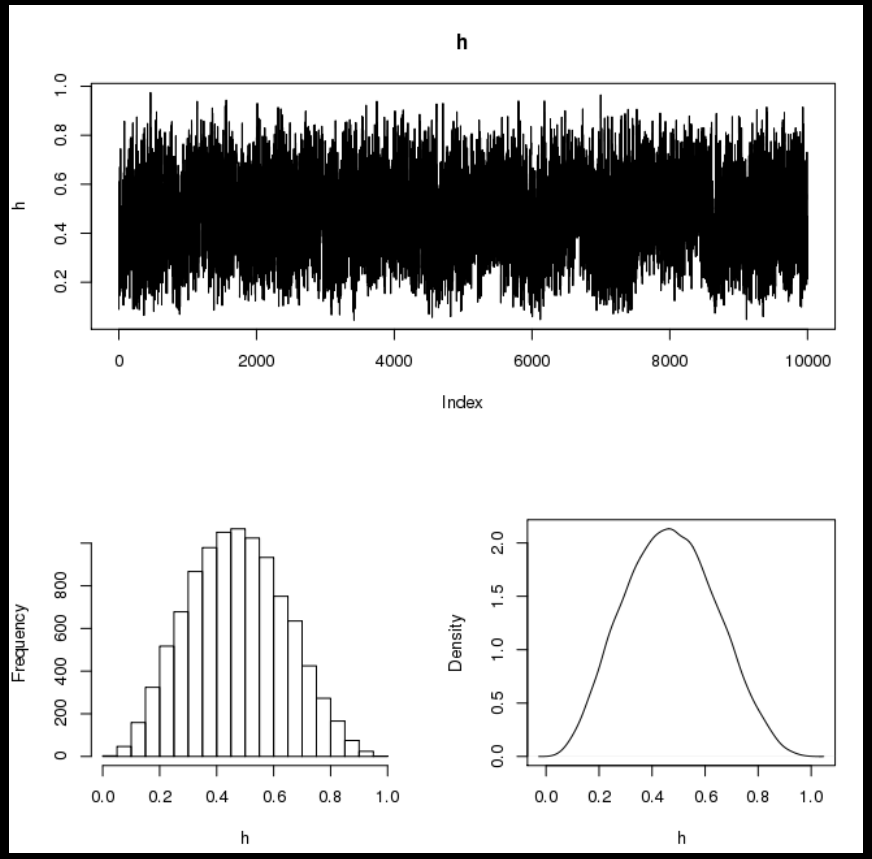
```
# Get mean, median, and 95% ETPI of hyperparameters
# ================================================================

# h-> approximation to proportion of phenotypic variance
#     explained by variants (PVE)
h<-c("h",mean(hyp.params$h),quantile(hyp.params$h, probs=c(0.5,0.025,0.975)))

# pve -> PVE
pve<-c("PVE", mean(hyp.params$pve),quantile(hyp.params$pve,
probs=c(0.5,0.025,0.975)))

# rho-> approximation to proportion of genetic variance explained by variants
#       with major effect (PGE)
#       rho=0 -> pure LMM, highly polygenic basis
#       rho=1 -> pure BVSR, few major effect loci
rho<-c("rho",mean(hyp.params$rho),quantile(hyp.params$rho, probs=c(0.5,0.025,0.975)))

# pge -> PGE
pge<-c("PGE",mean(hyp.params$pge),quantile(hyp.params$pge, probs=c(0.5,0.025,0.975)))

# pi -> proportion of variants with non-zero effects
pi<-c("pi",mean(hyp.params$pi),quantile(hyp.params$pi, probs=c(0.5,0.025,0.975)))

# n.gamma -> number of variants with major effect
n.gamma<-c("n.gamma",mean(hyp.params$n_gamma),quantile(hyp.params$n_gamma,
probs=c(0.5,0.025,0.975)))

# ================================================================
```

Analysing BSLMM output (3)

```
# get table of hyperparameters and save it to a file
# ================================================================================

hyp.params.table<-as.data.frame(rbind(h,pve,rho,pge,pi,n.gamma),row.names=F)
colnames(hyp.params.table)<-c("hyperparam", "mean","median","2.5%", "97.5%")

# show table
hyp.params.table

# write table to file
write.table(hyp.params.table, file="hyperparameters.dsv", sep="\t", quote=F)

# ================================================================================


# Table should look like this:

> hyp.params.table
  hyperparam              mean       median           2.5%           97.5%
1          h    0.470609736039     0.467607   0.1598727275    0.8033639375
2        PVE     0.51012475696    0.4945775     0.33735173    0.7534928675
3        rho    0.643165383114    0.6588942     0.22740654      0.97564247
4        PGE     0.71103881682    0.7117142    0.428942495     0.978721755
5         pi 1.32164779026e-05 1.060446e-05 3.20580365e-06 3.648634375e-05
6    n.gamma           4.7737            4              1              13
```

Analysing BSLMM output (4)

```
# plot traces and distributions of hyperparameters
# ================================================================================
# set up layout
layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE))

# h
# --------------------------------------------------------------------------------
plot(hyp.params$h, type="l", ylab="h", main="h")
hist(hyp.params$h, main="", xlab="h")
plot(density(hyp.params$h), main="", xlab="h")
# --------------------------------------------------------------------------------
```



Analysing BSLMM output (5)

```
# plot traces and distributions of hyperparameters
# ================================================================================
# set up layout
layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE))

# h
# --------------------------------------------------------------------------------
plot(hyp.params$h, type="l", ylab="h", main="h")
hist(hyp.params$h, main="", xlab="h")
plot(density(hyp.params$h), main="", xlab="h")
# --------------------------------------------------------------------------------

# PVE
# --------------------------------------------------------------------------------
plot(hyp.params$pve, type="l", ylab="PVE", main="PVE")
hist(hyp.params$pve, main="", xlab="PVE")
plot(density(hyp.params$pve), main="", xlab="PVE")
# --------------------------------------------------------------------------------

# rho
# --------------------------------------------------------------------------------
plot(hyp.params$rho, type="l", ylab="rho", main="rho")
hist(hyp.params$rho, main="", xlab="rho")
plot(density(hyp.params$rho), main="", xlab="rho")
# --------------------------------------------------------------------------------
```

Analysing BSLMM output (6)

```
# PGE
# ------------------------------------------------------------------------
plot(hyp.params$pge, type="l", ylab="PGE", main="PGE")
hist(hyp.params$pge, main="", xlab="PGE")
plot(density(hyp.params$pge), main="", xlab="PGE")
# ------------------------------------------------------------------------

# pi
# ------------------------------------------------------------------------
plot(hyp.params$pi, type="l", ylab="pi", main="pi")
hist(hyp.params$pi, main="", xlab="pi")
plot(density(hyp.params$pi), main="", xlab="pi")
# ------------------------------------------------------------------------

# No gamma
# ------------------------------------------------------------------------
plot(hyp.params$n_gamma, type="l", ylab="No gamma", main="No gamma")
hist(hyp.params$n_gamma, main="No gamma", xlab="No gamma")
plot(density(hyp.params$n_gamma), main="No gamma", xlab="No gamma")
# ------------------------------------------------------------------------
# ========================================================================
```

Analysing BSLMM output (7)

```
# summarize it all in a pdf
# ================================================================================
pdf(file="hyperparameters.pdf", width=8.3,height=11.7)
layout(matrix(c(1,1,2,3,4,4,5,6), 4, 2, byrow = TRUE))
# h
# --------------------------------------------------------------------------------
plot(hyp.params$h, type="l", ylab="h", main="h - trace")
hist(hyp.params$h, main="h - posterior distribution", xlab="h")
plot(density(hyp.params$h), main="h - posterior distribution", xlab="h")
# --------------------------------------------------------------------------------

# PVE
# --------------------------------------------------------------------------------
plot(hyp.params$pve, type="l", ylab="PVE", main="PVE - trace")
hist(hyp.params$pve, main="PVE - posterior distribution", xlab="PVE")
plot(density(hyp.params$pve), main="PVE - posterior distribution", xlab="PVE")
# --------------------------------------------------------------------------------

# rho
# --------------------------------------------------------------------------------
plot(hyp.params$rho, type="l", ylab="rho", main="rho - trace")
hist(hyp.params$rho, main="rho - posterior distribution", xlab="rho")
plot(density(hyp.params$rho), main="rho - posterior distribution", xlab="rho")
# --------------------------------------------------------------------------------

# PGE
# --------------------------------------------------------------------------------
plot(hyp.params$pge, type="l", ylab="PGE", main="PGE - trace")
hist(hyp.params$pge, main="PGE - posterior distribution", xlab="PGE")
plot(density(hyp.params$pge), main="PGE - posterior distribution", xlab="PGE")
# --------------------------------------------------------------------------------

# pi
# --------------------------------------------------------------------------------
plot(hyp.params$pi, type="l", ylab="pi", main="pi")
hist(hyp.params$pi, main="pi", xlab="pi")
plot(density(hyp.params$pi), main="pi", xlab="pi")
# --------------------------------------------------------------------------------

# No gamma
# --------------------------------------------------------------------------------
plot(hyp.params$n_gamma, type="l", ylab="n_gamma", main="n_gamma - trace")
hist(hyp.params$n_gamma, main="n_gamma - posterior distribution", xlab="n_gamma")
plot(density(hyp.params$pi), main="n_gamma - posterior distribution", xlab="n_gamma")
# --------------------------------------------------------------------------------
dev.off()
# ================================================================================
```

Analysing BSLMM output (8)

```
# Open the script and copy and paste the commands line by line in R;
# change orange text as required

$ nano scripts/gemma_param.R

setwd("/data/$USER/gwas_gemma/output")

# library to speed up loading of big tables
library(data.table)

# Load parameters
# ================================================================
params<-as.data.frame(fread("bslmm.param.txt",header=T,sep="\t"))
# ================================================================

# Get variants with sparse effect size on phenotypes
# ================================================================================
# add sparse effect size (= beta * gamma) to data frame
params["eff"]<-abs(params$beta*params$gamma)

# get variants with effect size > 0
params.effects<-params[params$eff>0,]

# show number of variants with measurable effect
nrow(params.effects)
    [1] 19984

# sort by decreasing effect size
params.effects.sort<-params.effects[order(-params.effects$eff),]
```

Analysing BSLMM output (9)

```
# show top 10 variants with highest effect

head(params.effects.sort, 10)
```

	chr	rs	ps	n_miss	alpha	beta	gamma	eff
4929	-9	lg8_ord45_scaf1036-131573	-9	0	-1.145693e-05	-1.1078320	0.6366	0.70524585
341474	-9	lg8_ord55_scaf1512-149001	-9	0	-4.435923e-05	-0.7819882	0.7327	0.57296275
4943	-9	lg8_ord45_scaf1036-131605	-9	0	-1.130519e-05	-1.0537550	0.3634	0.38293457
138641	-9	lg6_ord32_scaf531-110990	-9	0	6.521205e-05	1.0086120	0.0349	0.03520056
315197	-9	lgNA_ordNA_scaf784-237602	-9	0	9.862680e-05	0.7155290	0.0317	0.02268227
5125	-9	lg3_ord81_scaf488-29865	-9	0	7.033625e-05	0.9475088	0.0227	0.02150845
5105	-9	lg3_ord81_scaf488-29866	-9	0	7.056699e-05	0.9437216	0.0196	0.01849694
138636	-9	lg6_ord32_scaf531-110989	-9	0	6.473289e-05	1.0086750	0.0121	0.01220497
157503	-9	lg10_ord71_scaf134-639009	-9	0	7.311403e-05	0.8475411	0.0121	0.01025525
67311	-9	lg12_ord32_scaf239-410063	-9	0	4.265473e-05	1.4520500	0.0070	0.01016435

Analysing BSLMM output (10)

```
# variants with the highest sparse effects
# ------------------------------------------------------------------

# top 1% variants (above 99% quantile)
top1<-
params.effects.sort[params.effects.sort$eff>quantile(params.effects.sort$eff,0.99),]

# top 0.1% variants (above 99.9% quantile)
top01<-
params.effects.sort[params.effects.sort$eff>quantile(params.effects.sort$eff,0.999),]

# top 0.01% variants (above 99.99% quantile)
top001<-
params.effects.sort[params.effects.sort$eff>quantile(params.effects.sort$eff,0.9999),]
# ------------------------------------------------------------------

# write tables
write.table(top1, file="top1eff.dsv", quote=F, row.names=F, sep="\t")
write.table(top01, file="top0.1eff.dsv", quote=F, row.names=F, sep="\t")
write.table(top001, file="top0.01eff.dsv", quote=F, row.names=F, sep="\t")

# ==================================================================
```

Analysing BSLMM output (11)

```
# Get variants with high Posterior Inclusion Probability (PIP) == gamma
# ======================================================================
# PIP is the frequency a variant is estimated to have a sparse effect in the MCMC
# (the number of times it appears in the MCMC with a non-zero sparse effect)

# sort variants by descending PIP
params.pipsort<-params[order(-params$gamma),]

# Show top 10 variants with highest PIP
head(params.pipsort,10)
```

	chr	rs	ps	n_miss	alpha	beta	gamma	eff
341474	-9	lg8_ord55_scaf1512-149001	-9	0	-4.435923e-05	-0.7819882	0.7327	0.572962754
4929	-9	lg8_ord45_scaf1036-131573	-9	0	-1.145693e-05	-1.1078320	0.6366	0.705245851
4943	-9	lg8_ord45_scaf1036-131605	-9	0	-1.130519e-05	-1.0537550	0.3634	0.382934567
138641	-9	lg6_ord32_scaf531-110990	-9	0	6.521205e-05	1.0086120	0.0349	0.035200559
315197	-9	lgNA_ordNA_scaf784-237602	-9	0	9.862680e-05	0.7155290	0.0317	0.022682269
298599	-9	lg10_ord64_scaf380-30883	-9	0	-2.315351e-04	-0.3223986	0.0294	0.009478519
5125	-9	lg3_ord81_scaf488-29865	-9	0	7.033625e-05	0.9475088	0.0227	0.021508450
5105	-9	lg3_ord81_scaf488-29866	-9	0	7.056699e-05	0.9437216	0.0196	0.018496943
251899	-9	lg3_ord35_scaf22-16272	-9	0	8.211789e-05	0.5633320	0.0171	0.009632977
50207	-9	lg8_ord60_scaf2482-78371	-9	0	1.493102e-04	0.4555014	0.0154	0.007014722

Analysing BSLMM output (11)

```
# sets of variants above a certain threshold
# variants with effect in 1% MCMC samples or more
pip01<-params.pipsort[params.pipsort$gamma>=0.01,]
# variants with effect in 10% MCMC samples or more
pip10<-params.pipsort[params.pipsort$gamma>=0.10,]
# variants with effect in 25% MCMC samples or more
pip25<-params.pipsort[params.pipsort$gamma>=0.25,]
# variants with effect in 50% MCMC samples or more
pip50<-params.pipsort[params.pipsort$gamma>=0.50,]

# write tables
write.table(pip01, file="pip01.dsv", quote=F, row.names=F, sep="\t")
write.table(pip10, file="pip10.dsv", quote=F, row.names=F, sep="\t")
write.table(pip25, file="pip25.dsv", quote=F, row.names=F, sep="\t")
write.table(pip50, file="pip50.dsv", quote=F, row.names=F, sep="\t")
# ------------------------------------------------------------------
```

Analysing BSLMM output (12)

```
# plot variants PIPs across linkage groups/chromosomes
# ====================================================================

# Prepare data
# --------------------------------------------------------------------
# add linkage group column (chr)
chr<-gsub("lg|_.+","",params$rs)
params["chr"]<-chr

# sort by linkage group and position
params.sort<-params[order(as.numeric(params$chr), params$rs),]

# get list of linkage groups/chromosomes
chrs<-sort(as.numeric(unique(chr)))
# --------------------------------------------------------------------

# Plot to a png file because the number of dots is very high
# drawing this kind of plot over the network is very slow
# also opening vectorial files with many objects is slow
# --------------------------------------------------------------------
# --------------------------------------------------------------------

png(file="pip_plot.png", width=11.7,height=8.3,units="in",res=200)

# set up empty plot
plot(-1,-1,xlim=c(0,nrow(params.sort)),ylim=c(0,1),ylab="PIP",xlab="linkage group",
xaxt="n")
```

Analysing BSLMM output (13)

```
# plot grey bands for chromosome/linkage groups
# ------------------------------------------------------------------------
chrs<-sort(as.numeric(unique(chr)))
start<-1
lab.pos<-vector()
for (ch in chrs){
  size<-nrow(params.sort[params.sort$chr==ch,])
  cat ("CH: ", ch, "\n")
  colour<-"light grey"
  if (ch%%2 > 0){
    polygon(c(start,start,start+size,start+size,start), c(0,1,1,0,0), col=colour,
border=colour)
  }
  cat("CHR: ", ch, " variants: ", size, "(total: ", (start+size), ")\n")
  txtpos<-start+size/2
  lab.pos<-c(lab.pos, txtpos)

  start<-start+size
}

# Add variants outside linkage groups
chrs<-c(chrs,"NA")
size<-nrow(params.sort[params.sort$chr=="NA",])
lab.pos<-c(lab.pos, start+size/2)
# ------------------------------------------------------------------------

# Add x axis labels
axis(side=1,at=lab.pos,labels=chrs,tick=F)
```

Analysing BSLMM output (14)

```
# plot PIP for all variants
# ------------------------------------------------------------------------
# rank of variants across linkage groups
x<-seq(1,length(params.sort$gamma),1)
# PIP
y<-params.sort$gamma
# sparse effect size, used for dot size
z<-params.sort$eff
# log-transform to enhance visibility
z[z==0]<-0.00000000001
z<-1/abs(log(z))
# plot
symbols(x,y,circles=z, bg="black",inches=1/5, fg=NULL,add=T)
# ------------------------------------------------------------------------
```

Analysing BSLMM output (15)

```
# highligh high PIP variants (PIP>=0.25)
# ------------------------------------------------------------------------
# plot threshold line
abline(h=0.25,lty=3,col="dark grey")
# rank of high PIP variants across linkage groups
x<-match(params.sort$gamma[params.sort$gamma>=0.25],params.sort$gamma)
# PIP
y<-params.sort$gamma[params.sort$gamma>=0.25]
# sparse effect size, used for dot size
z<-params.sort$eff[params.sort$gamma>=0.25]
z<-1/abs(log(z))

symbols(x,y,circles=z, bg="red",inches=1/5,fg=NULL,add=T)
# ------------------------------------------------------------------------

# add label high PIP variants
text(x,y,labels=params.sort$rs[params.sort$gamma>=0.25], adj=c(0,0), cex=0.5)
# ------------------------------------------------------------------------
# ------------------------------------------------------------------------


# close device
dev.off()
# ========================================================================

# This is to be done outside the current R session. Launch another interactive session
# in Iceberg and execute:

$ display -resize 1920x1080 output/pip_plot.png
```

Analysing BSLMM output (16)