# R2GUESS: a GPU-based R package for a Bayesian variable selection model accommodating multivariate responses

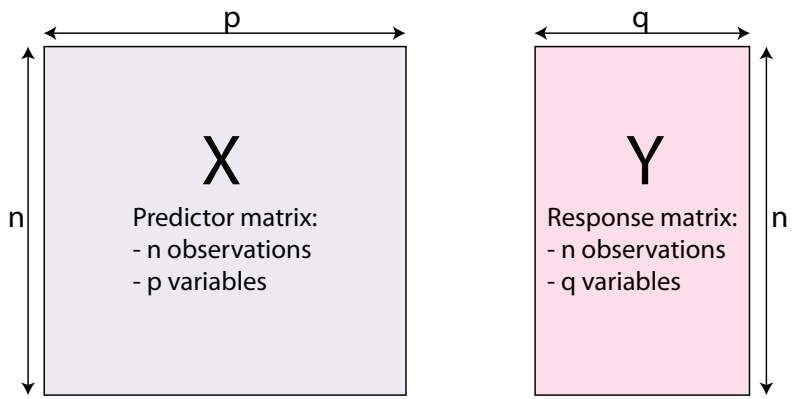
Habib Saadi [2], Benoît Liquet [1], Marc Chadeau-Hyam [2], Leonardo Bottolo [2], and Sylvia Richardson [1]

[1] MRC Biostatistics Unit, Cambridge

[2] Imperial College, London

# Aims and constraint

## Data definition:




Aim: identify which of the $p$ variables in $X$ are significantly associated with the outcome $Y$

- Very large variety of models: $2^p$
- Multivariate outcomes: $q > 1$
- Special case: $n < p$

# Sparse Bayesian Multiple Regression (SBRM)

$$\boldsymbol{Y} - \boldsymbol{X}_\gamma \boldsymbol{B}_\gamma \sim \mathcal{N}(I_n, \Sigma)$$

- $\Sigma$ is a $q \times q$ covariance matrix
- $\boldsymbol{X}_\gamma \boldsymbol{B}_\gamma$ a linear predictor based on $\boldsymbol{X}_\gamma$
- $\boldsymbol{X}_\gamma$: original design matrix deprived of the columns that are not used to predict $\boldsymbol{Y}$.

$$\boldsymbol{X} = \begin{array}{c} \begin{array}{ccccccc} 1 & \dots & \dots & \dots & \dots & \dots & p \end{array} \\ \begin{pmatrix} \times & \dots & \dots & \times & \dots & \dots & \times \\ \times & \dots & \dots & \times & \dots & \dots & \times \\ \vdots & \dots & \dots & \vdots & \dots & \dots & \vdots \\ \times & \dots & \dots & \times & \dots & \dots & \times \end{pmatrix} \end{array} \Longrightarrow \underset{(n \times p_\gamma)}{\boldsymbol{X}_\gamma} = \begin{array}{c} \begin{array}{ccc} 1 & \dots & p_\gamma \end{array} \\ \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \vdots & \vdots & \vdots \\ \times & \times & \times \end{pmatrix} \end{array}$$

- Sparsity is then induced by a latent binary variable $\gamma$

# Prior specification

- Sparsity prior on the model space: Beta binomial on $\gamma$

$$p(\gamma_i|\omega) \sim B(\omega) \quad \omega \sim Beta(a_\omega, b_\omega)$$

- Matrix of regression coefficients $\boldsymbol{B}_\gamma$:

$$\boldsymbol{B}_\gamma|\gamma, \Sigma \sim \mathcal{N}(H_\gamma, \Sigma)$$

- $\Sigma \sim \mathcal{IW}(a_\Sigma, Q_\Sigma)$, with $E(\Sigma) = Q_\Sigma/(a_\Sigma - 2)$, $Q_\Sigma = sI_q$
- Matrix $H_\gamma = g\left(\boldsymbol{X}_\gamma^T \boldsymbol{X}_\gamma\right)^{-1}$ regulates the dependencies among the $p$ predictors.
- The parameter $g$ ("$g$-prior") influence the variable selection procedure (link to the model size, e.g. small g favors saturated models) $\Longrightarrow$ included in the MCMC scheme

$$g \sim InvGa(1/2, n/2)$$

## Implementation of SBMR : Evolutionary Monte Carlo sampler

- Use an Evolutionary Stochastic Search (ESS) algorithms
  $\hookrightarrow$ make model search and implementation feasible and robust.
  $\hookrightarrow$ open source C$^{++}$ code (Bottolo et al. Bioinformatics, 2011)

- Key features of ESS$^{++}$
  - $\boldsymbol{B}_\gamma$ and $\Sigma$ are integrated to obtain marginal likelihood
  - Fully Bayesian inference for $g$ and $\gamma$
  - Use evolutionary Monte Carlo (EMC), (Liang and Wong, 2000; Jasra et al., 2007)
    $\hookrightarrow$ combines MCMC and genetic algorithms using different chains with tempering.

# Implementation of ESS$^{++}$

- The computation of the marginal likelihood requires repeated evaluation of $\boldsymbol{Y}^T \boldsymbol{X}_\gamma \left( \boldsymbol{X}_\gamma^T \boldsymbol{X}_\gamma \right)^{-1} \boldsymbol{X}_\gamma^T \boldsymbol{Y}$
- To compute this efficiently the algorithm uses QR decomposition $\hookrightarrow$ becomes prohibitively computationally intensive on the CPU in the case of GWAS,
- To overcome this, we use CUDA which enables to use the multiprocessor power of the GPU (graphics processing unit). $\Longrightarrow$ C$^{++}$ software GU-ESS based on previous ESS$^{++}$ algorithm.
- Specifically we use the CULA library, allowing linear algebra computations to be performed on the GPU.
- We can do roughly 1 million model evaluations in 10 hours (in 3 chains for example).
- Open source code available in http://www.bgx.org.uk/software/guess.html

# GUESS usage: parameters

Input: Most of the parameters are automatically set up and only the following values are required to run GUESS:

- $E(p_\gamma)$ and $\sigma_{p_\gamma}$: The mean and s.d. for the # of underlying factors.
- $L$: the number of chains
- The number of iterations & burn-in
- The specific parameters for each move:
  $\hookrightarrow$ These are more complex but some standard values are available

# GUESS usage: Ouput

Output: ESS provides a large variety of outputs to monitor the moves and their acceptance probability, but more importantly:

- Main outputs:
  - The history of all visited models and their conditional posterior
  - The posterior distribution of $g$
- From these output, the following are computed:
  - The posterior of all model visited (and their rank)
  - The posterior model size
  - The posterior shrinkage factor ($g$)
  - The marginal posterior probability of inclusion (MPPI)
  - Jeffrey's scale: $log_{10}(BF)$ (BF=Bayes Factors)
  - And many more ($R^2 \ldots$)

# R2GUESS: GPU-based R package

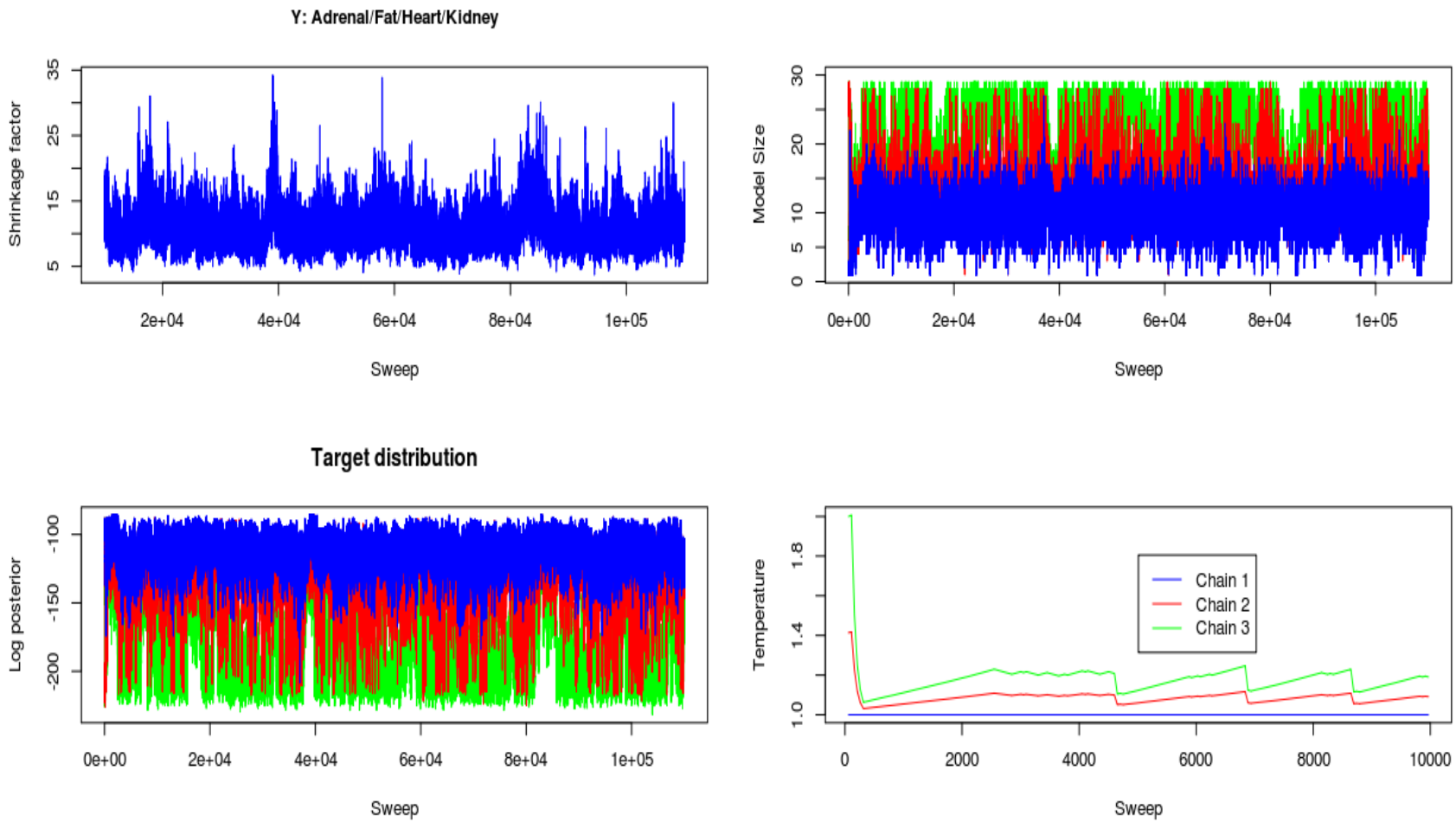
R2GUESS provide an easy way to

- run the C$^{++}$ program GUESS
  ↪ `R2GUESS()`, `R2GUESS.permut()`
  ↪ define an object of class `ESS`
- investigate the convergence of the algorithm
  ↪ `plot()`, `check.convergence()`
- analyse the results
  ↪ `as.ESS.object()` $\longrightarrow$ define an `ESS` object
  ↪ `print()`, `summary()` $\longrightarrow$ resume of the best models
- interpret the results
  ↪ `plot.MPPI()`: plot the marginal posterior probability of inclusion of each predictor $P(\gamma_j = 1|Y)$
  `plot.model()`, `plot.variable()`, `plot.cim()`, ...

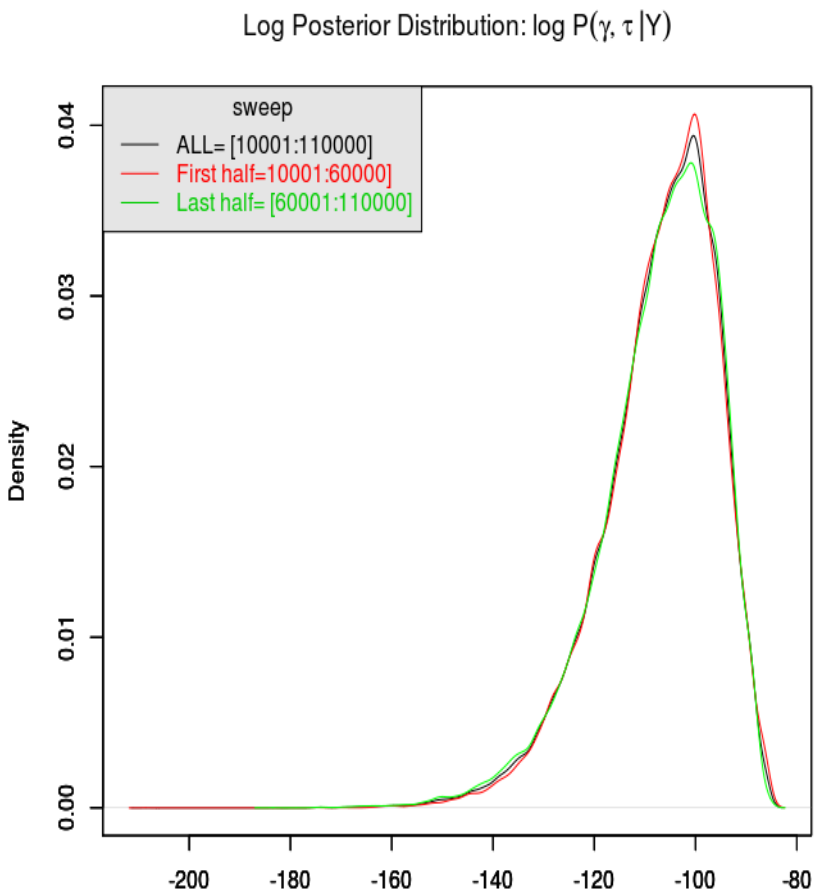
# Illustration: Genetic regulation of expression

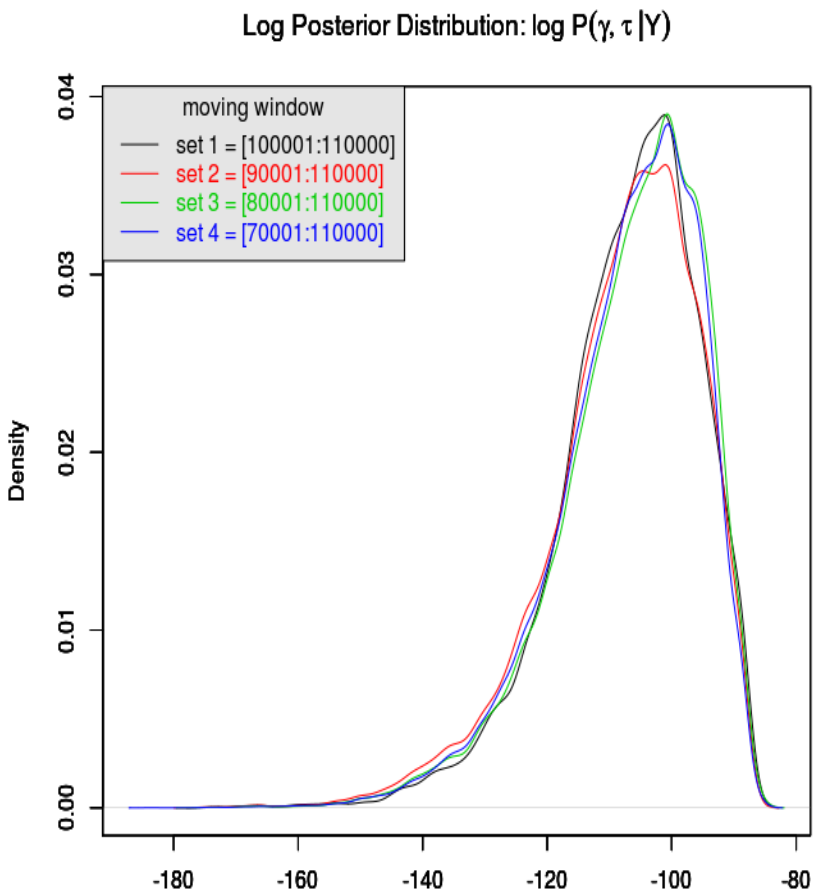
- Analyse two genes (Cd36 and Hopx) presented in Petretto et al. (2010)
- gene expression treated as quantitative phenotype ($\boldsymbol{Y}$)
- genotype data (SNPs) are used as predictors ($\boldsymbol{X}$)
- Investigate the ability of ESS++ to find a parsimonious set of predictors that explain the joint variability of gene expression in $q = 4$ tissues (adrenal, fat, heart, kidney) using 770 SNPs ($\boldsymbol{X}$) and $n = 29$ observations taken from the rat inbred lines.

# Monitoring ESS: `plot()`



# Investigate Convergence: `check.convergence()`



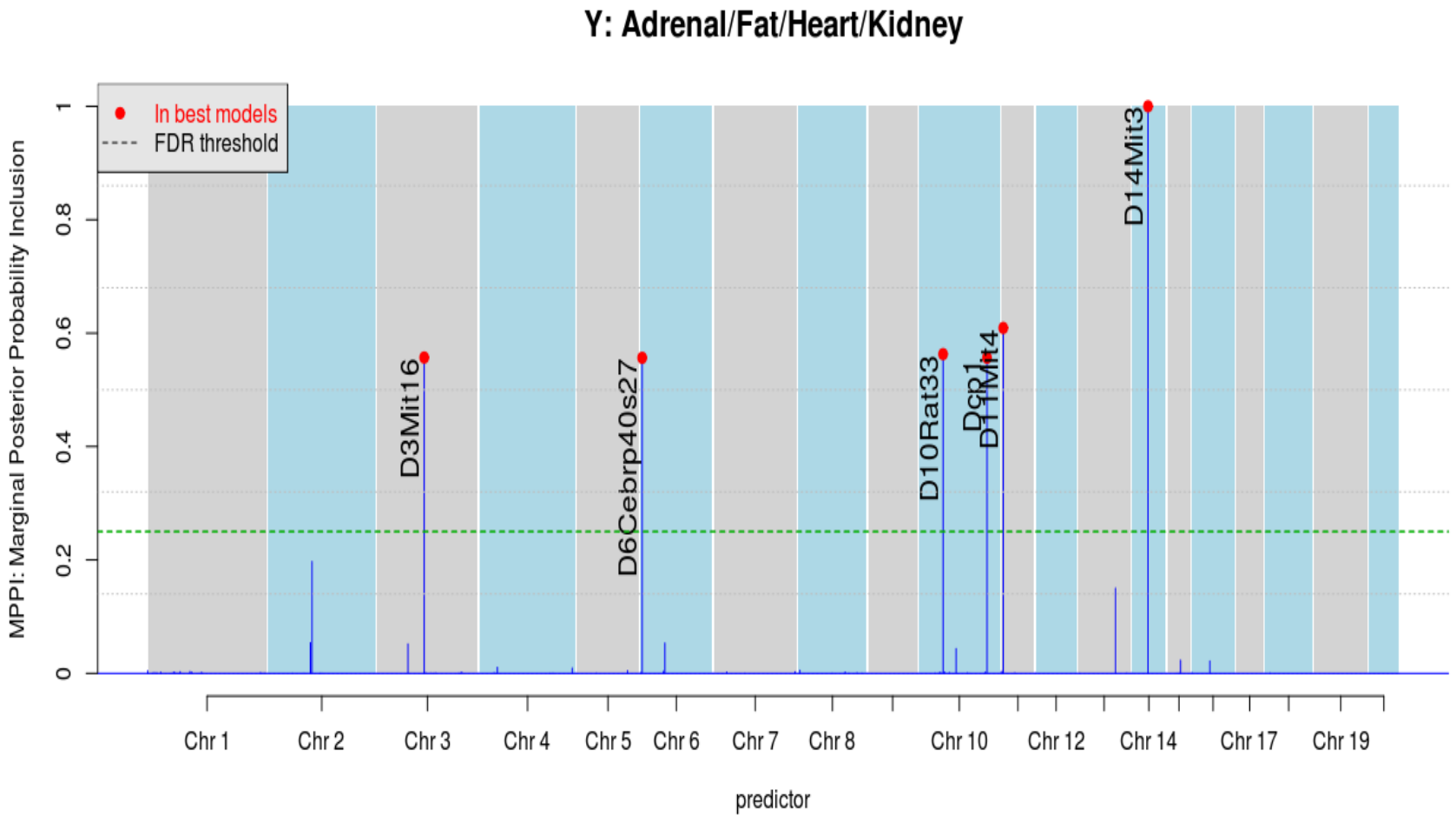
# How to select the best predictors ?

- Marginal Posterior Probability of Inclusion (MPPI) for each SNP
  $\hookrightarrow P(\gamma_j = 1|Y)$ measure the marginal contribution of each predictor
- Define threshold for MPPI with respect to a specified FDR level.
  For a threshod $c$
  (i) Define $R$ the number of declared association (MPPI$<c$)
  (ii) For each $j$ $(1, \ldots, N)$ artificial data set (permutation of the phenotype) compute the number $S_j$ of falsely declared associations.
  (iii) Choose the threshod $c$ such as the ratio $\frac{1}{N}\sum_{j=1}^{N}\frac{S_j}{R}$ is not greater than a FDR level (0.05)
- Use the function `FDR.permutation.parrallel()` which performs this computation by running GUESS on each permutation in a parallel way (argument *nbcpu*) using the R package `snowfall`

# How to select the best models ?

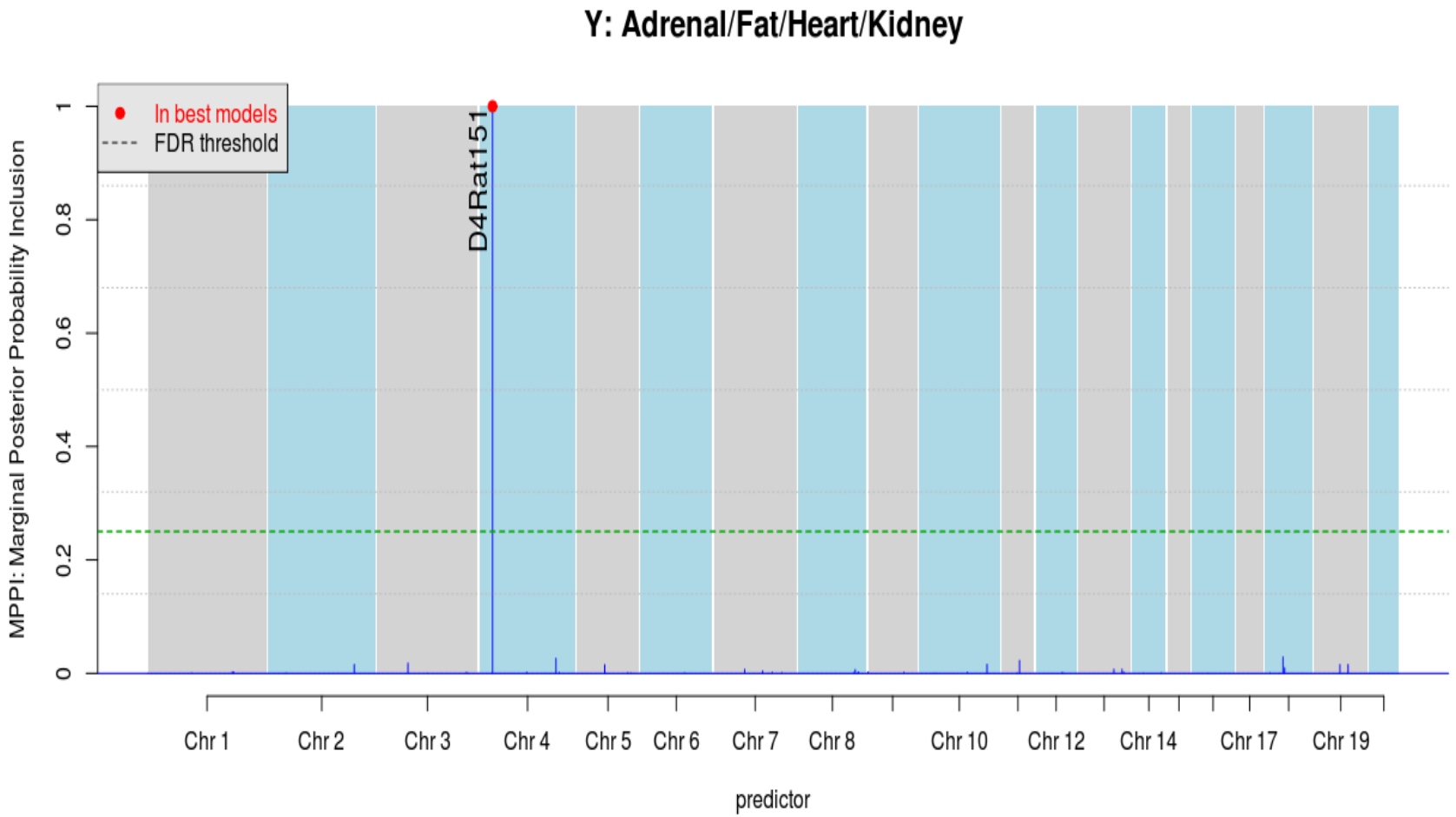
- Assessing model importance by the posterior model probability of each unique model ($s$): $P(\gamma^{(s)}|Y)$
- Compute the Jeffrey's Scale: $JF = log_{10}BF(\gamma^{(B)};\gamma^{(0)})$ where $BF(\gamma^{(B)};\gamma^{(0)}) = p(Y|\gamma^{(B)})/p(Y|\gamma^{(0)})$ represent the Bayes Factor comparing the best model ($\gamma^{(B)}$) with the null model ($\gamma^{(0)}$).
- Define threshold for Jeffrey's Scale to highlight best models by permutation method
  $\hookrightarrow$ threshold is defined by the $1-\alpha$ quantile of the distribution of the JF's scale based on the artificial data set (permutation of the phenotype).
- Use the function `FDR.permutation.parrallel()`

# Hopx gene: MPI (`plot.MPI()`)



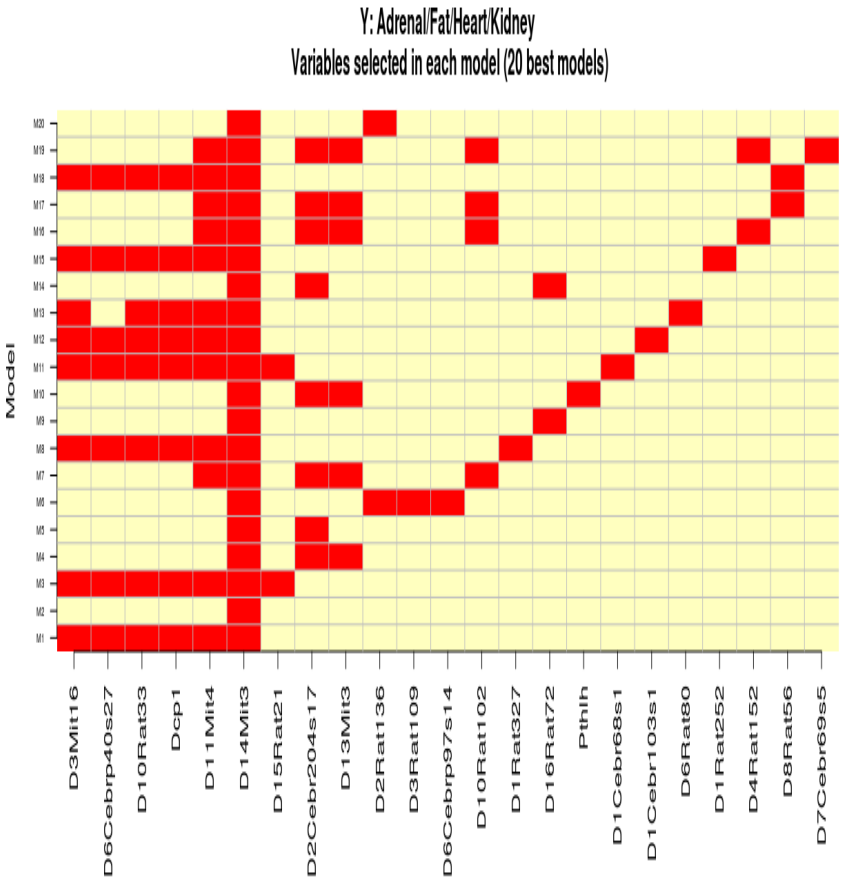

The 10 best models are polygenic.

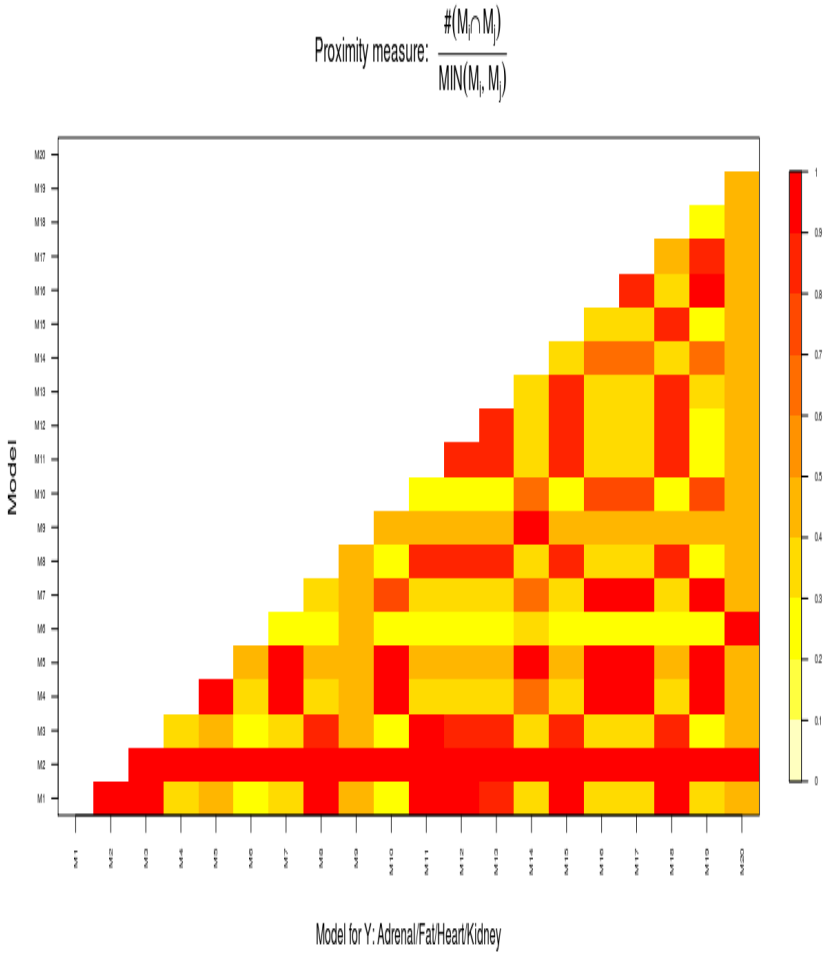
# CD36 gene: MPI (`plot.MPI()`)



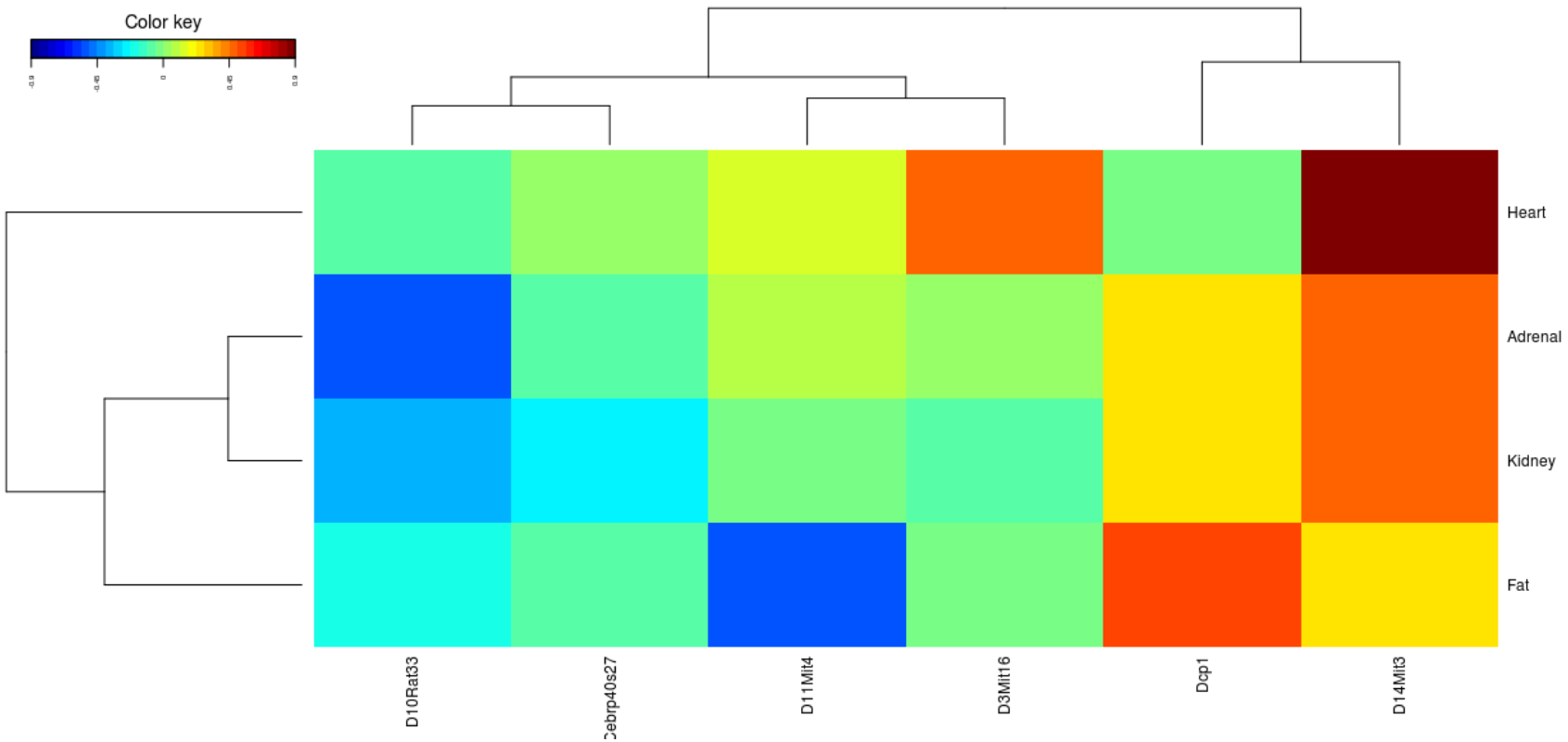

Single effect on the variation of the gene expression in the 4 tissues.

# Plot functions: `plot.model()`, `plot.model()`



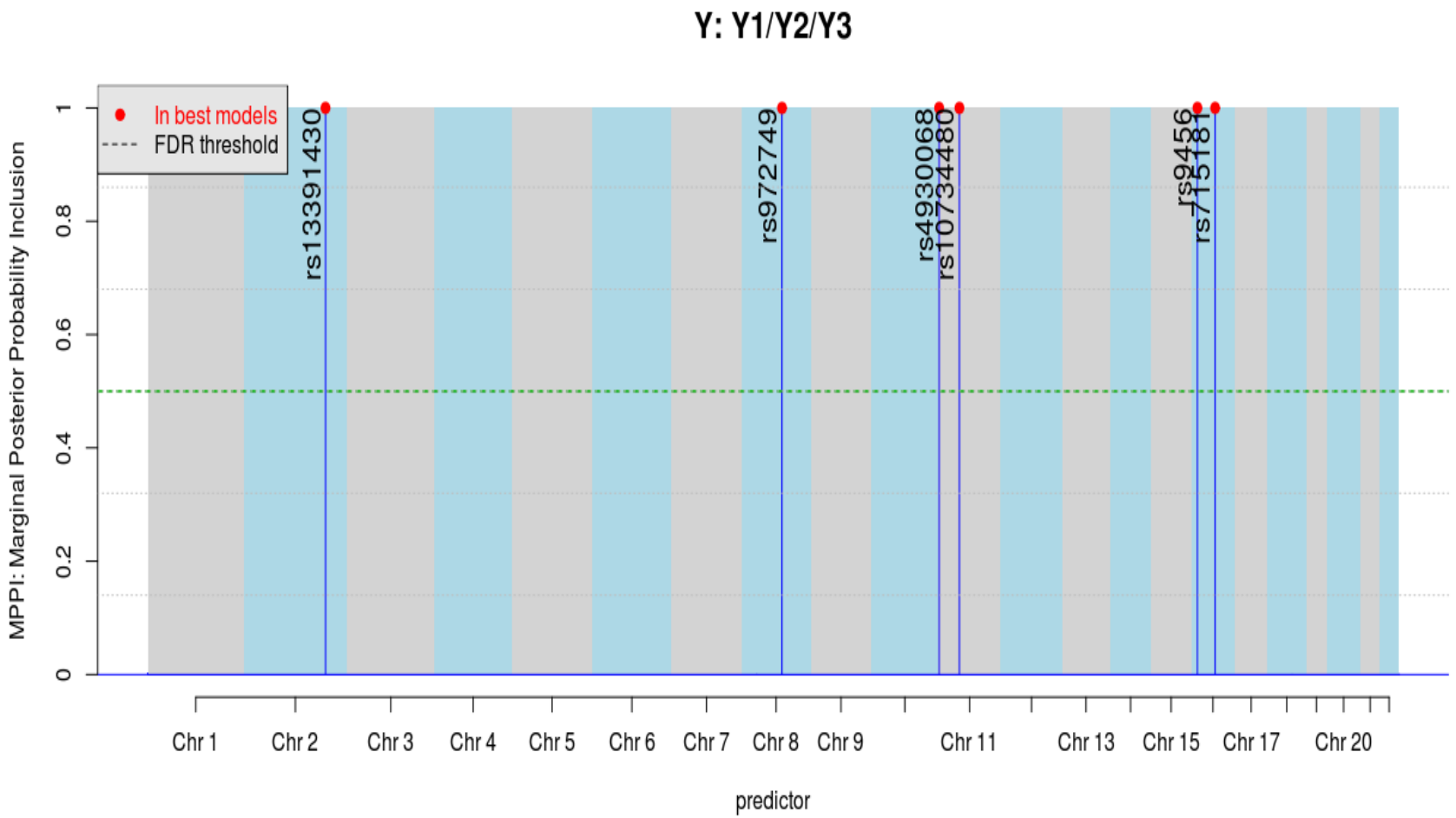
# Cluster image mapping: `plot.cim()`



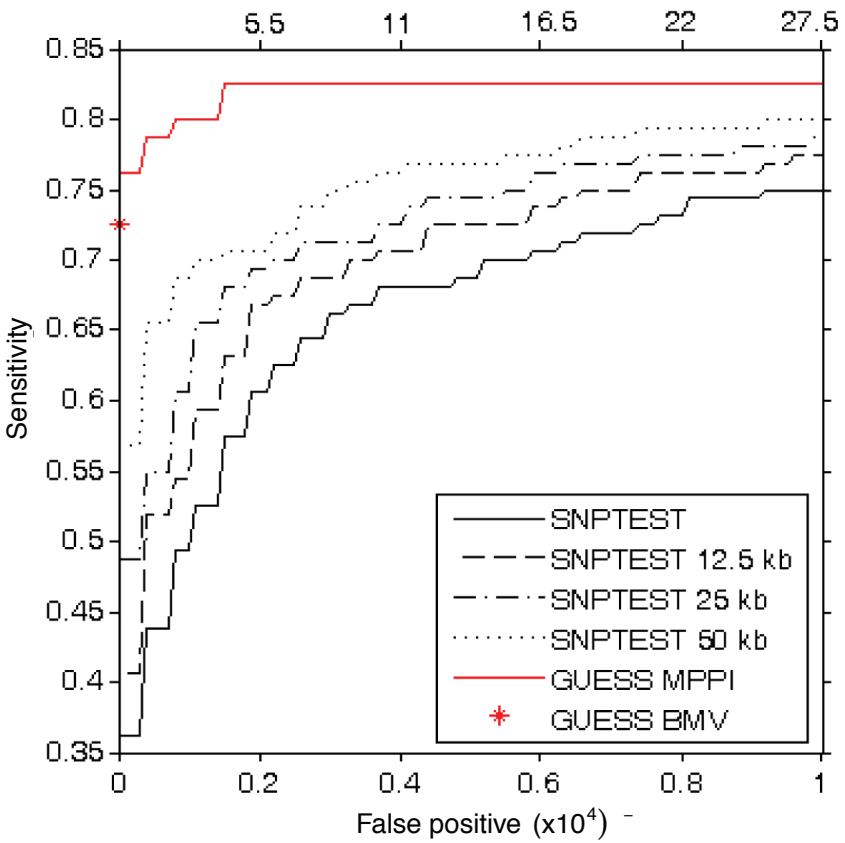
# Illustration: Genome Wide Association Study

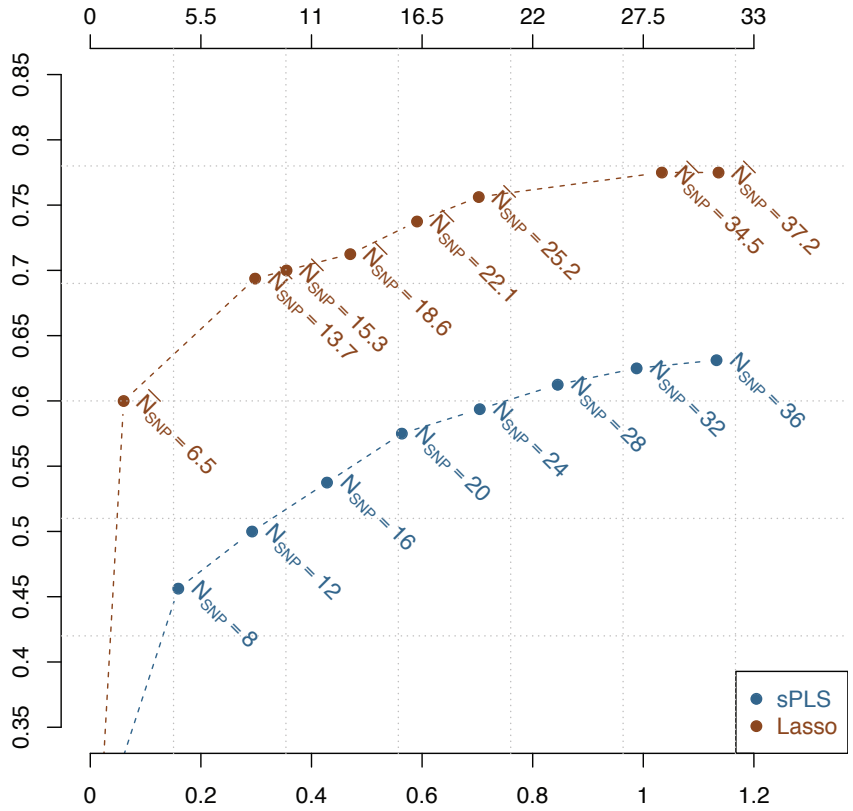
- Large data-set:
  - $X$: 273 675 SNPs;
  - $n = 3122$ subjects;
  - $Y$: 3 correlated phenotypes.
- Need to use GUESS instead of ESS
  $\hookrightarrow$ argument $CUDA = 1$ in `R2GUESS()`
- Simulation of the phenotypes:
  - based on 8 SNPs from the 273 675 SNPs
  - Highlights the specificity of GUESS
  - Comparison over alternative methods: (SNPTest, sPLS, Lasso)

# Marginal probability of Inclusion: `plot.MPPI()`



# Simulation results



# Conclusion and perspectives

- On the R2GUESS package:
  - Easy way to run the C++ code
  - Functions and methods to analyse the runs
  - Propose an R GUI
  - Submit in CRAN
  - Article in progress presented a tutorial of R2GUESS
- On the model:
  - High specificity of our approach
  - Increased power of GUESS over alternative methods (SNPTest, sPLS, Lasso).
  - incorporate in the model a frailty term (random effect) to take into count repeated measure or grouped data
  - Implement the model for binary response or qualitative response

# references


- Leonardo Bottolo and Sylvia Richardson. *Evolutionary stochastic search for bayesian model exploration*. Bayesian Analysis, 5(3):583-618, 2010.
- Leonardo Bottolo, Marc Chadeau-Hyam, David I. Hastie, Sarah R. Langley, Enrico Petretto, Laurence Tiret, David Tregouet, and Sylvia Richardson. *ESS++: a C++ objected-oriented algorithm for bayesian stochastic search model exploration.* Bioinformatics, 27(4):587-588, 2011.


ANY QUESTION ?