

MULLER
0.0.1
Guide de l'Utilisateur

Olivier Filangi ^{*†‡} Anne Ricard ^{§¶} Jean-Michel Elsen ^{||¶}

28 août 2013

*`olivier.filangi [at] rennes.inra.fr`

†INRA, UMR1348 PEGASE, Domaine de la Prise, F-35590 Saint Gilles, France

‡Agrocampus Ouest, UMR1348 PEGASE, Domaine de la Prise, F-35590 Saint Gilles, France

§`anne.ricard [at] toulouse.inra.fr`

¶INRA, UMR 1313, 78352 Jouy-en-Josas, France

||`Jean-Michel.Elsen [at] toulouse.inra.fr`

Table des matières

1 Généralités

1.1 Introduction

Müller et al (2011)¹ ont proposé une méthode pour le contrôle de l'erreur de première espèce dans les analyses d'association ou de liaison, basée sur une méthode rapide de simulation. Cette méthode est applicable aux approches de type « genome scan » dans lesquels chaque chromosome est exploré en testant successivement la présence d'un QTL en une succession de localisations. La méthode prend en compte la non-indépendance, engendrée par la liaison génétique, entre les distributions des statistiques de test aux différentes positions. Elle le fait très rapidement par le calcul des éléments de la matrice des covariances entre ces statistiques de test et sa décomposition spectrale.

Le logiciel développé dans le cadre de Rules & Tools est une application directe de l'algorithme proposé et permet d'exécuter des BLUP (programme BLUPF90 de Ignacy Misztal and collaborators, University of Georgia) afin d'obtenir les tests statistiques pour chaque SNP du groupe de liaison testé et enfin de calculer des seuils de rejets objectifs de l'hypothèse d'absence de QTL sur un groupe de liaison.

1.2 Modèle

Le GWAS est réalisé avec un modèle mixte pour chaque SNP.

$$y = \mathbf{1}\mu + \mathbf{X}b + \mathbf{Z}u + \mathbf{e} \quad (1)$$

y le vecteur des performances

μ l'effet moyen général à la population

b l'effet de l'allèle

\mathbf{X} le vecteur des génotypes (=0/1/2)

\mathbf{Z} la matrice d'incidence des performances sur les valeurs polygéniques.

\mathbf{e} la résiduelle

$V(\mathbf{u}) = \mathbf{A}\sigma_u^2$, \mathbf{A} étant la matrice de parenté.

2 Installation

2.1 Pré-requis

- Un compilateur fortran : gfortran (GNU) ou ifort (Intel)
- L'outil de génération de makefile CMake - Cross Platform Make
- L'une des bibliothèques suivantes :
 - Automatically Tuned Linear Algebra Software (ATLAS)
 - Librairie LAPACK (Linear Algebra PACKage)

1. Müller BU, Stich B, Piepho HP, 2011. A general method for controlling the genome-wide type I error rate in linkage and association mapping experiments in plants. *Heredity* 106 : 825-831

- Intel Math Kernel Library (Intel MKL)
- Le logiciel R (The R Project for Statistical Computing) pour la génération des graphiques

2.2 Téléchargement

Le programme est régi sous la licence GPLv3. Les versions figées de MULLER sont librement téléchargeables sur la forge du département de Génétique Animale de l'INRA.

2.2.1 Obtenir la version de développement sous Linux

```
svn export https://forge-dga.jouy.inra.fr/svn/lmuller muller-dev
```

Il faut générer l'environnement de développement en utilisant cmake :

```
mkdir -p build/debug;cd build/debug
cmake -DCMAKE_BUILD_TYPE=Debug -DCMAKE_Fortran_COMPILER=gfortran ../..
```

2.3 Compilation

Si les bibliothèques ATLAS/LAPACK/MKL ne sont pas installées dans `/usr/lib` :

```
export LD_LIBRARY_FLAGS=<path-lapack-ou-atlas-ou-mkl>:${LD_LIBRARY_FLAGS}
```

Pour compiler et installer le programme dans le répertoire `<path-for-pls4snp>` :

```
cmake -DCMAKE_INSTALL_PREFIX=<path-install> ../..;make;make install
```

2.4 Arborescence du package MULLER

Description du contenu du répertoire d'installation de MULLER :

- `bin/muller` L'exécutable ;
- `share/doc/muller.pdf` Cette documentation ;
- `share/muller/README` Information sur le logiciel ;
- `share/muller/example` Jeu de donnée exemple.

3 Format des fichiers utilisateurs

Le programme MULLER utilise un fichier de génotype (au format AIPL), un fichier de phénotype et un fichier de généalogie. Les identifiants des animaux sont numériques.

3.1 Le fichier des génotypes

Le fichier des données de génotype contient l'information au marqueur pour chacun des descendants. Ce format renseigne l'identifiant de l'individu en première colonne (numérique) et en deuxième colonne une chaîne de caractère décrivant l'information moléculaire. Ce fichier ne contient pas d'entête

3.1.1 Le format AIPL

L'identifiant des animaux (entier positif) est codé sur les 10 premiers caractères. Une option d'exécution permet de réduire ou d'augmenter ce champ. L'identifiant et le génotype sont séparés par un espace. Le génotype d'un individu est codé sur une chaîne de caractères. Ce codage utilise 4 valeurs entières. Une valeur est associée à chaque marqueur SNP :

- 0 pour le statut homozygote au premier allèle ;
- 1 le statut hétérozygote ;
- 2 le statut homozygote au deuxième allèle ;
- 5 le statut "valeur manquante".

```
4500 11211211
4501 01112102
4502 01211102
4503 01111102
...
```

3.2 Le fichier des animaux phénotypés

Ce fichier contient plusieurs colonnes avec les données suivantes (il n'est pas obligatoire de donner ces informations dans cet ordre) :

- identifiant
- effet fixé
- covariable
- mesure des phénotypes

L'exemple ci-dessous décrit un fichier avec l'identifiant en première colonne, un effet fixé en deuxième colonne et une mesure en troisième colonne.

```
10538 2 24.80373324
10722 2 20.58580806
9960 1 24.32563858
10239 2 18.3071405
...
```

3.3 Le fichier de description du modèle pour les BLUP

Ce fichier décrit (avec la même syntaxe que GS3 ou BLUPF90) le fichier de performance et permet de prendre en compte les effets de nuisance dans le modèle.

- **NUMBER_OF_EFFECTS** : nombre d'effet de nuisance dans le modèle
- **RECORD ID** : indice de la colonne où se trouve l'identifiant dans le fichier des phénotypes
- **OBSERVATION(S)** : indice de la colonne de la performance à étudier (indiquez une seule colonne!)
- **EFFECTS : POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT** : description des effets de nuisances
 - indice de la colonne de l'effet

- nombre de niveaux de l'effet (0 si covariable)
- nature de l'effet : *cross* (effet fixé), *cov* (covariable)

```

NUMBER_OF_EFFECTS
1
RECORD ID
1
OBSERVATION(S)
3
EFFECTS: POSITIONS_IN_DATAFILE NUMBER_OF_LEVELS TYPE_OF_EFFECT
2 2 cross

```

3.4 La gestion des données manquantes

Seul le fichier de génotype peut contenir des valeurs manquantes. Les individus sans status au génotype ne seront pas utilisés dans le BLUP à cette position.

4 Exécution

4.1 Arguments en entrée du programme

- `--geno <pathfile>`, `-g <pathfile>` : Le fichier des génotypes ;
- `--model <patternfile>`, `-z <pathfile>` : Le fichier décrivant le fichier des performances ;
- `--pedig <pathfile>`, `-p <pathfile>` : Le fichier pédigré ;
- `--data <pathfile>`, `-d <pathfile>` : Le fichier des performances ;
- `--solution <pathfile>`, `-b <pathfile>` : Le fichier des résultats des BLUP sur le groupe de liaison ;
- `--start <integer>`, `-a <integer>` : Index de la première position à tester ;
- `--end <integer>`, `-n <integer>` : Index de la dernière position à tester ;
- `--h2 <integer>`, `-h <integer>` : Héritabilité du caractère étudié.

4.1.1 Arguments optionnels

- `--idlength <integer>`, `-i <integer>` : Nombre de caractère utilisé pour le codage de l'identifiant numérique dans le fichier de génotypage (10 par défaut).
- `--out <pathfile>`, `-o <pathfile>` : Le fichier des résultats de l'analyse (sortie standard par défaut) ;
- `--vary <float>`, `-v <float>` : La variance phénotypique (par défaut on utilise l'estimateur classique) ;
- `--nsim <integer>`, `-s <integer>` : nombre de simulation pour le calcul des seuils de rejets de H0 (1000 par défaut) ;
- `--maf <float>`, `-m <integer>` : fréquence minimum pour le calcul des BLUP. (les autres positions sont ignorées) ;
- `--nmaxpval <integer>`, `-x <integer>` : nombre de position à afficher dans un ordre décroissant (pour l'affichage des plus gros effets) (30 par défaut) ;

- `--nthread <integer>`, `-N <integer>` : Nombre de processus utilisé par l'application.

4.1.2 Ligne de commande

Execution du programme en ligne de commande :

```
>${MULLER_PATH}/muller --geno <genotype-file> --model <parameter-file>
--pedig <pedigree-file> --data <data-file> --solution <solution-file>
--start <first-snp-position> --end <last-snp-position> --h2 <real>
```

4.2 Résultat de l'analyse

4.2.1 Interprétation de la sortie standard

La sortie standard peut être redirigée dans un fichiers avec l'option *-out*. La sortie standard donne :

- un récapitulatif des données en entrée du programme
- l'état de l'exécution et son étape
- les seuils de rejets (Muller et al)
- la liste et le rang des marqueurs ayant un effets les plus forts

Affichage des options disponibles et information sur le jeux de donnée utilisateur

```
===== OPTIONS =====
--geno,      -g      : genotype file
--model,     -z      : Model file
--pedig,     -p      : Pedigree file
--data,      -d      : Data file
--solution,  -b      : Solution file
--out,       -o      : Result file (<standart ouput> by default)
--idlength,  -i      : Length ID in the genotype file (default=10)
--start,     -a      : first snp to study (>=1)
--end,       -n      : last snp to study
--vary,      -v      : phenotypic variance (optional, by default
                    this quantity is compute with the classical estimator)
--h2,        -h      : heritability
--nsim,      -s      : Number of simulation (default=1000)
--maf,       -m      : Minimum MAF (default=0.02)
--nmaxpval,  -x      : Maximum number of pvalue to rank (default=30).
--nthread,   -N      : Number of thread usable by the program (default=1).
```

```
[out file=<stdout>]
[genotype file=genotypes1.uga]
[sizeid=10]
[start=      1 ]
[number of pos=      4998 ]
[nmaxpval=      30 ]
* no phenotyped variance defined *
[h2= 0.84999999999999998 ]
[maf= 1.99999995529651642E-002 ]
[number of simulation=      10000 ]
[model file=parm]
[pedigree file=genealogie]
[data file=performances]
[solution file=performances]
[nthread=      6 ]
```

```
=====
```

```
read model file...
[data] number of record =      781
```

```

[phenotypic variance= 33.211308229158668      ]
[pedigree] Number of record=      17000
[residual variance= 4.9816962343738007      ]
[genetic variance= 28.229611994784868      ]
read      1609 genotype.
Genotype First ID [      6625 ] =>      1      1      0      0
0      2      0      0      0      0 ...      2
Genotype Last ID [ 15500 ] =>      0      0      1      1
0      2      1      0      0      1 ...      2

```

Les deux dernières lignes donnent l'informations aux premiers marqueurs et dernier marqueur du premier et du dernier individu. Il permet de vérifier que le logiciel interprète correctement le fichier de typage. Attention, L'ID de l'individu est codé sur les 10 premiers caractères de la ligne (format fixe). L'option *-idlength* permet de changer ce codage.

Pour obtenir un format valide, vous pouvez appliquer ce script awk à votre fichier (un nouveau fichier de génotypage est crée) :

```
>awk '{printf("%10s %s\n", $1, $2)}' mon_fichier_genotypes > mon_fichier_genotypes.new
```

Calcul des BLUP

```

=====
call      3704 BLUPF90.
====> [100%]
Writing sol.txt file [id snp maf perfo test pvalue sol solmu std stdmu].

```

Calcul des BLUP avec BLUPF90. suite à cet étape un fichier solution est crée. Ce fichier contient une entête plus une ligne par BLUP calculé (SNP ayant une MAF suffisamment grande))

- indice du BLUP
- indice du SNP dans le calcul du BLUP
- la MAF du SNP
- le nombre d'animaux
- le test de Fisher
- la pvalue
- la solution pour b et μ et leurs écarts types.

Construction de la matrice des covariances entre les statistiques de test

```

===== Prepare covariance matrix to compute rejection thresholds =====
===
====> [100%]
Apply an ACP (DSYEV, Matrix dim=      3704      3704 )
=====
size(Eigenvalues<0) =      2831

```

Phases de Simulation pour l'obtention des seuils

```

===== SIMUL =====
# ID      1
# Jump Ahead by (1)*2^256
# ID      3
# Jump Ahead by (3)*2^256
# ID      7
# Jump Ahead by (7)*2^256
# ID     11

```

```

# Jump Ahead by (11)*2^256
# ID          9
# Jump Ahead by (9)*2^256
# ID          5
# Jump Ahead by (5)*2^256
# ID          12
# Jump Ahead by (12)*2^256
# ID          6
# Jump Ahead by (6)*2^256
# ID          8
# Jump Ahead by (8)*2^256
# ID          4
# Jump Ahead by (4)*2^256
# ID          2
# Jump Ahead by (2)*2^256
# ID          10
# Jump Ahead by (10)*2^256
==> [ 97%] ==> [100%]

```

Affichage des seuils de rejets Suite à l'étape de simulation, le programme affiche les seuils de rejets (10, 5 et 1%) et les SNP ayant les plus grands effets.

```

-----
+++++ REJECTION THRESHOLDS +++++
-----

```

```

=====
| Level | PValue | -Log10(Pvalue) |
-----
| 10% | 0.00014 | 3.841 |
| 5% | 0.00007 | 4.174 |
| 1% | 0.00001 | 4.879 |
=====

```

```

-----
+++++ HIGHEST PVALUES +++++
-----

```

```

=====
| Rank | SNP | -Log10(Pvalue) |
-----
| 1 | 3777 | 4.159 |
| 2 | 3785 | 4.048 |
| 3 | 3778 | 3.968 |
| 4 | 3781 | 3.807 |
| 5 | 3787 | 3.698 |
| 6 | 3784 | 3.650 |
| 7 | 3779 | 3.383 |
| 8 | 3776 | 3.236 |
| 9 | 3807 | 3.178 |
| 10 | 3432 | 3.090 |
| 11 | 3435 | 3.090 |
| 12 | 3766 | 3.063 |
| 13 | 1591 | 2.977 |
| 14 | 3699 | 2.920 |
| 15 | 3769 | 2.915 |
| 16 | 3772 | 2.915 |
| 17 | 3795 | 2.901 |
| 18 | 3815 | 2.854 |
| 19 | 3817 | 2.854 |
| 20 | 3786 | 2.833 |
| 21 | 3775 | 2.811 |
| 22 | 3801 | 2.772 |
| 23 | 3803 | 2.738 |
| 24 | 3804 | 2.738 |
| 25 | 3805 | 2.738 |

```

p_values observées en fonction des théoriques

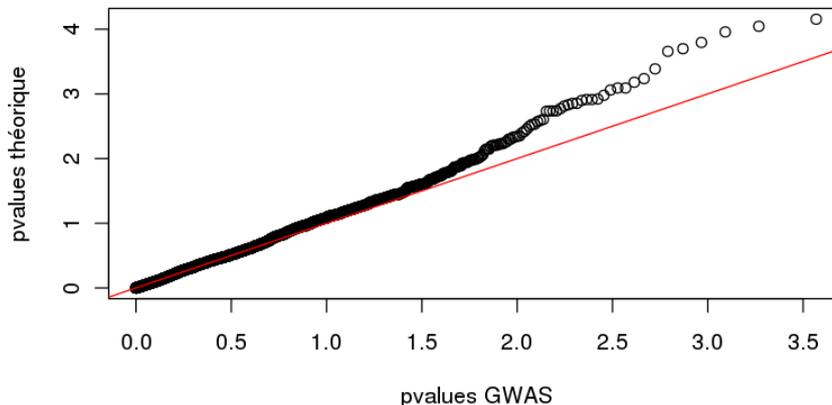


FIGURE 1 – Exemple de QQ plot

26	3774	2.733
27	3197	2.607
28	3170	2.588
29	4024	2.571
30	3179	2.532

=====
The GWAS solutions in file :sol.txt

+++++++ VISUALIZATION (USING R) ++++++

Pvalues distribution (compare theoretical distribution with real pvalues) / Manhattan plot

run "R -f genGraphR.R" to generate all graphics.

+++++++ END GWAS ++++++

Interprétation des Graphiques le script genGraphR.R est généré par l'application pour créer deux graphiques :

- Un QQplot pour vérifier la distribution du test (sous H0 les pvalues sont distribuées uniformément).
- Un Manhattan plot (pvalues en fonction de la position du SNP)

Pour obtenir les graphes :

```
>R -f genGraphR.R
```

4.2.2 Paramétrage applicatif

Redirection de la sortie standard L'option `-out` permet de rediriger les résultats de la sorties standard dans un fichier (seuils de rejets et ordonnancement des SNP ayant les

Manhattan plot - Seuils Muller/Bonferroni

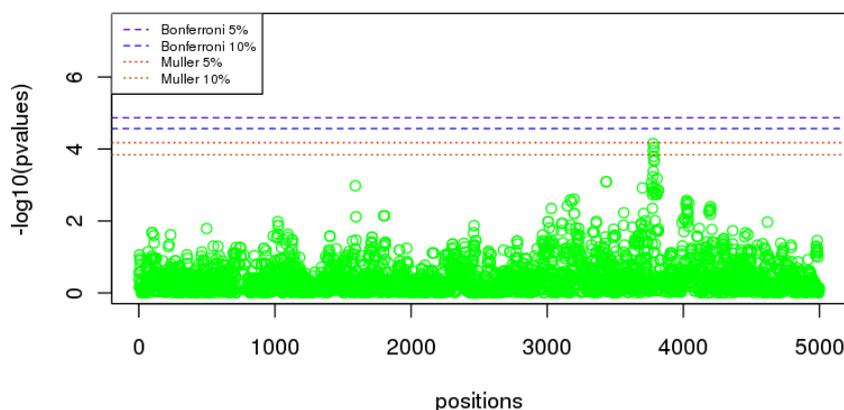


FIGURE 2 – Exemple de Manhattan plot

plus grand effets).

Filtre sur la MAF des SNP Par défaut le programme enlève les SNP ayant une MAF < 0.02 . Ce comportement est paramétrable avec l'option `-maf`.

Tableau des SNP ayant un fort effet Par défaut l'application affiche les 30 meilleurs SNP. L'option `-nmaxpval` permet de changer ce comportement.

Performance pour l'exécution La vitesse d'exécution est fortement dépendant du nombre de thread donné en argument du programme (option `-nthread`). Vous devez vous assurer que la machine dispose des ressources suffisantes pour paramétrer cette option.

5 Exemple d'analyse

5.1 Description du jeu de données

Un phénotype est simulé sur 2 chromosomes (9995 marqueurs, 1 Morgan/Chr). 1609 animaux sont génotypés et 781 sont phénotypés. La généalogie comporte 17000 animaux. L'héritabilité est de 0.85.

5.1.1 Position des marqueurs sur le génome

CHR	Start	End
1	1	4998
2	4999	9995

5.1.2 Effets des QTL sur le génome

CHR	Pos Morgan	SNP flanquant	Effet additif	Effet de dominance
1	20.00	1000	0.5	0.0
1	73.14	3655	1.0	0.0
2	20.00	5999	1.5	0.2
2	33.24	6660	0.5	0.2
2	59.60	7977	0.5	0.0

5.2 GWAS et obtention des seuils

Analyse sur le premier groupe de liaison

```
>${MULLER_PATH}/muller -g genotypes1.uga --start 1 --end 4998 -z parm -p genealogie  
-d performances --h2 0.85 --nsim 10000 --solution sol.txt --nthread 6
```

Analyse sur le deuxieme groupe de liaison

```
>${MULLER_PATH}/muller -g genotypes1.uga --start 4999 --end 9995 -z parm -p genealogie  
-d performances --h2 0.85 --nsim 10000 --solution sol.txt --nthread 6
```