

A MULTI-TRAIT MIXED-MODEL FOR GENOME-WIDE ASSOCIATION STUDY IN
BARLEY POPULATION

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By
Qiang Li

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Statistics

December 2014

Fargo, North Dakota

North Dakota State University
Graduate School

Title

A MULTI-TRAIT MIXED-MODEL FOR GENOME-WIDE ASSOCIATION
STUDY IN BARLEY POPULATION

By

Qiang Li

The Supervisory Committee certifies that this *disquisition* complies with North
Dakota State University's regulations and meets the accepted standards for the
degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Yarong Yang
Chair

Dr. Megan Orr

Dr. Seung Won Hyun

Dr. Shaobin Zhong

Approved:

4/15/2015
Date

Dr. Rhonda Magel
Department Chair

ABSTRACT

Genome-wide association study (GWAS) is a standard approach for studying the genetics of natural variation, typically focusing on association between single-nucleotide polymorphisms (SNPs) and traits. For GWAS, if multiple traits are also correlated, some essential and important information among multi-traits can be lost in marginal analysis. Therefore, fully parameterized multi-trait mixed-model is emerged as a flexible approach that considers both the within-trait and between-trait variance components simultaneously for multiple traits. The phenotype data used in this research is 1100 barley entries with reaction of leaf spot diseases. In this dataset, there are two different traits, Rate (reaction of spot form net) and Rating (reaction of spot blotch). 3941 SNPs makers dataset is used as genotype data. A multi-trait mixed-model was applied for genome-wide association study. The results show that the multi-trait mixed model can detect much more associations between single-nucleotide polymorphisms (SNPs) and traits than single-trait analysis can do.

TABLE OF CONTENTS

ABSTRACT.....	iii
LIST OF TABLES.....	v
LIST OF FIGURES.....	vi
LIST OF ABBREVIATIONS.....	vii
CHAPTER 1. INTRODUCTION.....	1
CHAPTER 2. METHODOLOGY.....	3
2.1. Theory.....	3
2.1.1. Multiple traits mixed model.....	3
2.1.2. Estimating the variance parameters.....	4
2.2. Applied to GWAS.....	5
CHAPTER 3. CASE STUDY.....	6
CHAPTER 4. RESULTS.....	15
CHAPTER 5. CONCLUSION AND DISCUSSION.....	20
REFERENCES.....	22
APPENDIX A. MTMM.....	24
APPENDIX B. EMMA FUNCTION.....	38

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Part of Phenotype Data	7
2. Part of SNPs Position Data	10
3. New SNPs Position Data	11
4. Part of the Original Genotype Data	12
5. Part of Genotype Data.....	13
6. Part of IBS Data	14
7. Part of Output.....	16
8. Statistics of Data	17

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Distribution of the First Trait.....	8
2. Distribution of the Second Trait	8
3. Distribution of the Combined Trait.....	9
4. Manhattan Plots	18
5. QQ Plot	19

LIST OF ABBREVIATIONS

GWAS	genome-wide association study
MTMM	multi-trait mixed model
SNP	single nucleotide polymorphism
IBS	identity by state
MAF	minor allele frequency
MAC	minor allele count
GLS	generalized least square

CHAPTER 1. INTRODUCTION

Genome-wide association study (GWAS) is an examination of many common genetic variants in different individuals to see if any variant is associated with a trait. GWAS typically focus on associations between single-nucleotide polymorphisms (SNPs) and traits. GWAS helps us to understand the genetic basis of many complex traits better [McCarthy et al., 2008]. If we can understand the relationship between phenotypic traits and genetic variations for these quantitative and complex traits well, it is good for us to predict disease risk and develop management measures for plants.

At the beginning stage of GWAS, researchers mainly focused on conducting the simplest possible statistical model: a single-locus test of association between a SNP genotype and a single phenotype. Given that most traits of interest are multifactorial, this clearly amounts to model misspecification, and the resulting danger of biased results is well known [Korte et al., 2012]. Multi-trait models used in quantitative genetics have a long history, but rarely been applied to GWAS. There is almost no attention to the fact that phenotypes may also be correlated. Therefore recent analysis suggested that a substantial proportion of heritability was not missing but hidden in the common variants with small or moderate effects [Yang et al., 2010, Makowsky et al., 2011]. Researchers start to focus on simultaneously analyzing multiple correlated traits to increase ability to detect associations [Korte et al., 2012]. This is because the correlated traits may share common genetic factors, which is known as pleiotropy. This pleiotropy is the main cause of the genetic correlations between traits [Bolormaa et al., 2014, Sivakumaran et al., 2011].

Here, the mixed model is extended to handle correlated phenotypes by deriving a fully parameterized multi-trait mixed model (MTMM). It considers both the within-trait and between-trait variance components simultaneously for multiple traits, implementing it for GWAS [Korte

et al., 2012]. The increasing amount of nonhuman GWAS shows that the need for modeling correlated traits will grow. In this study, the multi-trait mixed model is applied to a real barley population data [Triticeae Coordinated Agricultural Project]. The results show that the MTMM can detect more SNPs and traits associations than the single-trait model can do.

CHAPTER 2. METHODOLOGY

2.1. Theory

2.1.1. Multiple traits mixed model

Following Henderson [Henderson et al., 1984], the mixed model can be written for the phenotypes of n individuals as

$$y = X\beta + g + \varepsilon$$

where y is a vector of the n phenotype values. In this equation, the design matrix X contains the trait mean together with other fixed effects. The β are the effect sizes of the fixed effects, $g \sim N(0, \sigma_g^2 R)$ are the random effects, and $\varepsilon \sim N(0, \sigma_\varepsilon^2 I)$ are random errors [Korte et al., 2012]. They follow that the covariance matrix for the trait values, y , is

$$\text{var}(y) = \sigma_g^2 R + \sigma_\varepsilon^2 I$$

where R is an $n \times n$ relatedness matrix. Here two traits were taken into consider, y_1 and y_2 , measured on the same set of individuals, then under the mixed model for the k th phenotype follows the partitions of the variance accordingly, i.e., $\text{var}(y_k) = \sigma_{gk}^2 R + \sigma_{\varepsilon k}^2 I$. However, for the covariance matrix between the two phenotypes, it is not obvious what the appropriate model is [Korte et al., 2012]. Henderson [Henderson, et al., 1976] suggests the following covariance model

$$\text{cov}(y_1, y_2) = \sigma_{g1} \sigma_{g2} \rho_g R + \sigma_{\varepsilon1} \sigma_{\varepsilon2} \rho_\varepsilon I$$

where ρ_g captures the genetic correlation between two phenotypes and the term ρ_ε captures the correlation caused by non-genetic sources of correlations.

2.1.2. Estimating the variance parameters

In order to estimate the variance components, we need to obtain the maximum likelihood estimates. If the covariance matrix can be written as $\text{cov}(y) = V\sigma_g, \sigma_e, \rho_g, \rho_e = V\sigma, \rho$ where σ_g, σ_e are vectors containing all genetic and error variance scalars in the model, ρ_g, ρ_e are matrices with all the genetic and error trait correlations in the model, and $\theta = (\sigma_g, \sigma_e, \rho_g, \rho_e)$ [Korte et al., 2012]. The general likelihood can then be written as

$$\ln(\beta, \theta) = C - \frac{1}{2} [\log |V_\theta| + (Y - X\beta)' V_\theta^{-1} (Y - X\beta)]$$

where C is a constant. Similarly, the restricted likelihood can be written as

$$\ln_{re}(\hat{\beta}, \theta) = C - \frac{1}{2} [\log |V_\theta| + (Y - X\hat{\beta})' V_\theta^{-1} (Y - X\hat{\beta}) + \log |X' V_\theta^{-1} X|]$$

where $\hat{\beta}$ is the generalized least square (GLS) estimate, and thus intrinsically dependent on the covariance matrix structure [Kang, et al., 2008, Harville, et al., 1974]. However, in practice, maximizing the likelihood is very important for large datasets, especially when the number of variance component parameters in θ is large. An approach used by ASReml is the average information algorithm [Gilmour, et al., 1995], which makes use of the first and second derivative of the likelihood to rapidly maximize the likelihood using a hill-climbing approach [Korte et al., 2012].

Computational efficiency can be increase greatly if we can pick good starting points. What we are going to do is using the genetic and error variance estimates in marginal analysis (single-trait analysis) as starting values for the corresponding variances, σ_{gi} and σ_{ei} , in the mixed model.

2.2. Applied to GWAS

For two traits (the i th and j th traits), the proposed approximation effectively assumes that the three variance ratios ($\sigma_{g_i}/\sigma_{\epsilon_i}, \sigma_{g_j}/\sigma_{\epsilon_j}$ and $\sigma_{g_i}/\sigma_{g_j}$) and the two correlations $\rho_{g_{ij}}$ and $\rho_{\epsilon_{ij}}$ are fixed with and without the marker in the model [Korte et al., 2012]. For multiple traits, we can search for causal loci with common effects (across all traits) as well as trait-specific loci or loci with opposite effects in different traits [Korte et al., 2012]. Depending on what we are interested in, a GLS F-test can be constructed to compare two models. For two traits we can write the single marker model as

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \sum_1^2 s_i \mu_i + x\beta + (x * \sum_1^2 s_i)\alpha + v$$

where x is the vector of SNPs and s_i is a vector with 1 for all values belonging to the i th trait and 0 otherwise. The $v \sim N(0, \text{cov}(y))$ is a random variable capturing both the error and genetic random effects. Depending on what kind of loci we are interested in, three different F-tests tests are built:

- The full model (include the effect of the marker genotype and its interaction) tested against a null model (include either) where $\beta = 0$ and $\alpha = 0$. This identifies both loci with common and interaction effects in one model.
- To identify the interaction effects between the traits, we propose to test the full model against a genetic model where $\alpha = 0$.
- Finally, to identify common genetic effects we propose to test the genetic model (include marker genotype, $\alpha = 0$) against a null model where $\beta = 0$ and $\alpha = 0$.

As both the interaction test and the common effect test are sensitive to scaling of the phenotype values, we need to make sure their normality.

CHAPTER 3. CASE STUDY

We applied the MTMM on a barley population dataset collected in Triticeae Coordinated Agricultural Project (T-CAP), which is funded by the National Institute for Food and Agriculture (NIFA) of the United States Department of Agriculture (USDA).

The phenotype data we used in the study are “Trial NSGC SFNB 2013 NDSU DEN2.6” and “Trial CAP2row Spot Blotch 2012 Fargo GH”. SNPs markers data and SNPs position data were provided by Plant Pathology Department of North Dakota State University (NDSU). These two datasets were combined as genotype data.

The phenotypes of the barley population are shown in Table 1. There are 1943 barley entries. The first column of Table 1, `ecotype_id`, is the individual names of the barley. The second column is the phenotypic values of the first trait, which is the infection response rating (1-5) to spot form net blotch. It is “Trial NSGC_SFNB_2013_NDSU_DEN2.6”. The third column is the phenotypic values of the second trait, which is the infection response rating (0-9) to spot blotch. It is “Trial CAP2row_SpotBlotch_2012_FargoGH”. These two separated phenotype data were combined together by the same individual name, as multi-trait data used in this research.

Table 1. Part of Phenotype Data

ecotype_id	RATE	RATING
Clho455	2.5	7.444444444
Clho497	1.2	5.916666667
Clho521	2.8	7.555555556
Clho1138	2.2	6.583333333
Clho1172	3	7.444444444
Clho1388	3.8	3.777777778
Clho1458	3.7	5.666666667
Clho1461	3.3	6
.....
PI636084	2.3	7.666666667

The probability distribution of the first trait, Rate, is shown in Figure 1 as black dots. The red curve in Figure 1 denotes the probability density function of the normal distribution with mean 2.936 and standard deviation 0.7093, which are as same as Rate's. Since the black dots strictly on the red curve, we assume the first trait follows a normal distribution.

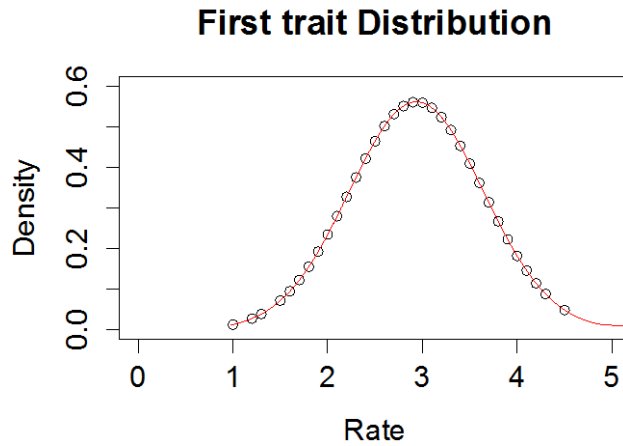


Figure 1. Distribution of the First Trait

The probability distribution of the second trait, Rating, is shown in Figure 2 as black dots. The red curve in Figure 2 denotes the probability density function of the normal distribution with mean 6.646 and standard deviation 0.8725, which are as same as Rating's. Since the black dots strictly on the red curve, we assume the second trait follows a normal distribution.

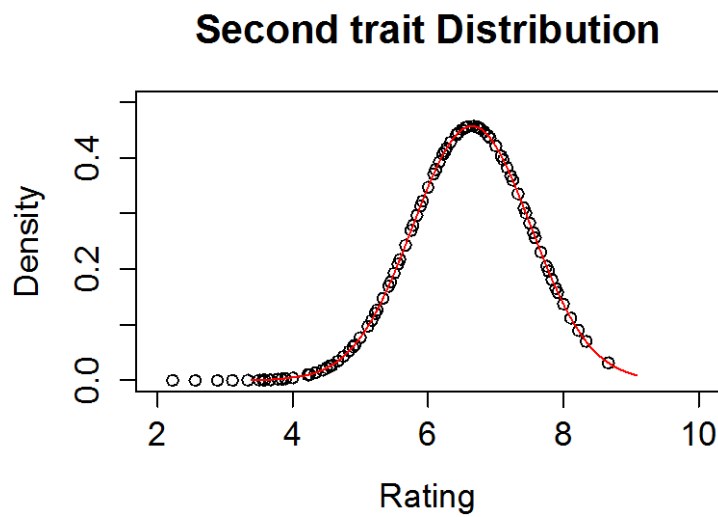


Figure 2. Distribution of the Second Trait

The two traits were combined as the combined trait for further study. The probability distribution of the combined trait is shown in Figure 3 as black dots. The red curve in Figure 3 denotes the probability density function of the normal distribution with mean 4.791 and standard deviation 2.0186, which are as same as combined traits'. Since the black dots strictly on the red curve, we assume the combined trait follows a normal distribution.

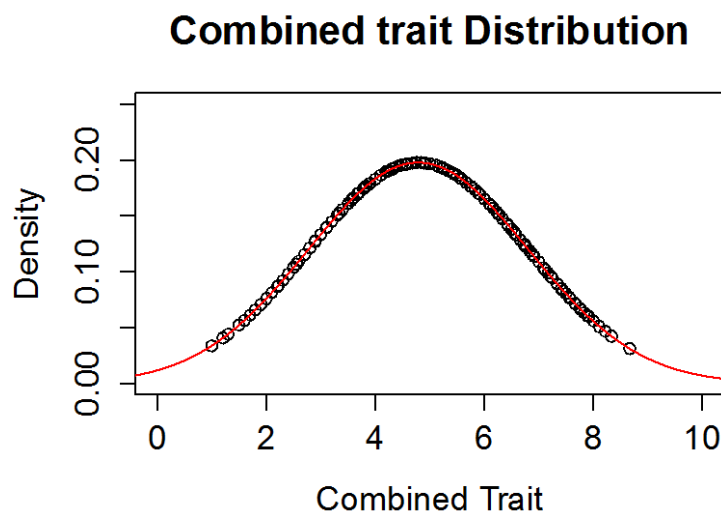


Figure 3. Distribution of the Combined Trait

The original SNPs position data is shown in Table 2. There are 7843 rows in this dataset representing 7843 SNPs. The first column of Table 2 is the index of the SNPs. The second column is the original name of the SNPs. The third column is the chromosome the corresponding SNP belongs to. The fourth column is the position of the SNPs. Since the algorithms implemented in the R package don't deal with the cases that multiple SNPs share the same position (for example, the 3121st and the 3122nd SNPs share position 0.2), we made some slight modification to the position data to make each SNP has a unique position. For example, if we keep the position of the 3121st SNP, then we modify the position of the 3122nd SNP to be 0.21.

However, the corresponding index number should be kept in mind, it helps us to recognize the original name of SNPs. Table 3 shows the unique SNPs Position Data. The chromosome and position columns in Table 2 are combined to be SNPs columns of “chromosome-position” format in Table 3. In Table 2, there are some SNPs that are not identified with its chromosome and position (e.g., the 7862nd SNP). We eliminate those SNPs with ending up with 3941 rows in Table 3.

Table 2. Part of SNPs Position Data

Index	Original Name	chromosome	position
2531	BOPA2_12_30969	1H	0
3121	SCRI_RS_120053	1H	0.2
3122	SCRI_RS_120059	1H	0.2
2631	BOPA2_12_31149	1H	10.9
2932	SCRI_RS_106754	1H	100.1
4046	SCRI_RS_146959	1H	100.1
6338	SCRI_RS_201086	1H	100.1
.....
7862	SCRI_RS_99836	U	0

Table 3. New SNPs Position Data

Index	SNP
2531	1H_0
3121	1H_0.20
3122	1H_0.21
2631	1H_10.9
2932	1H_100.10
4046	1H_100.11
6338	1H_100.12
.....
5342	7H_99.9

The original maker data is shown in Table 4. There are 6916 rows and 2421 columns in this table. The first row of Table 4 gives the individual name, which is the first column of Table 1. The first column of Table 4 is the index of SNPs, which is the first column of Table 3. The second column of Table 4 is the name of the SNPs, which is the Original Name in Table 2. The AA or BB in Table 4 means different SNP in the DNA sequence. For computational purpose, we convert AA and BB to be 0 and 1 respectively. For example, suppose the reference DNA sequence is AAGGCCT, and the barley CIho455 DNA sequence is AAGGCCG. The difference between these two sequences is the seventh nucleotide. Because of the difference, we mark CIho455 as "BB", and transfer it to 1 for computational purpose. If these two sequences are the completely same, we mark CIho455 as "AA", and transfer it to 0 for computational purpose.

Table 4. Part of the Original Genotype Data

Index	SNP	CIho455	CIho497	PI636084
1	BK_01	BB	BB	BB
3	BK_03	BB	BB	BB
4	BK_04	BB	BB	BB
5	BK_05	BB	BB	AA
7	BK_07	AA	AA	AA
8	BK_08	BB	AA	BB
10	BK_10	BB	BB	BB
11	BK_11	BB	BB	BB
12	BK_12	BB	BB	BB
.....
7864	SCRI_RS_99965	BB	BB	NA

SNPs position data and original marker data were combined following the same index. Since the R package we use in this study does not deal with missing data. We randomly generated 0 or 1 to make up the missing data. Because the generated data account for less than 1% of entire dataset, we ignored the influence of the generated data on the original data.

The new genotype data used in this research is shown in Table 5. It is a 1100 by 3941 matrix, where 1100 is the number of individuals and 3941 is the number of SNPs. The first column of Table 5 is the individual names. The SNPs names are shown in the first row.

Table 5. Part of Genotype Data

ecotype_id	4H_27.50	3H_51.600	4H_114.7
Clho455	0	1		1
Clho497	0	1	1
Clho521	0	1	1
Clho1138	0	1	1
Clho1172	0	0	1
Clho1388	0	1	1
Clho1458	0	1	1
Clho1461	0	1	1
.....
PI636084	0	0	1

The Identity by State (IBS) relationship matrix is shown in Table 6. It is a 1100 by 1100 matrix, where 1100 is the number of individuals. The first column and first row of Table 6 are the individual names. It is calculated by using emma function [Korte et al., 2012]. High IBS values show strong relationships.

Table 6. Part of IBS Data

	Clho455	Clho497	PI356674
Clho455	1	0.804364375		0.760974372
Clho497	0.804364375	1	0.789647298
Clho521	0.678000507	0.689926415	0.645013956
Clho1138	0.707434661	0.718853083	0.680030449
Clho1172	0.750063436	0.749809693	0.752093377
Clho1388	0.623699569	0.623953311	0.567876174
Clho1458	0.678761736	0.697792438	0.666582086
Clho1461	0.648058868	0.650850038	0.650596295
.....
PI356674	0.760974372	0.789647298	1

CHAPTER 4. RESULTS

Part of outputs are shown as following in Table 7. The first column of Table 7 is the SNPs names. The second column is MAF value. We expect $MAF > 0.05$ to show its much stronger power to make meaningful statements about very rare alleles. The third column is the p-value of the first trait in marginal analysis. The fourth column is the p-value of the second trait in marginal analysis. The fifth column is the p-value of the joint MTMM analysis with the full model. The sixth column is the p-value of the joint MTMM analysis with the interaction effect. The seventh column is p-value of the joint MTMM analysis with the common effect. Here, the significant level is 0.01, if the p-value is less than 0.01 means that this SNP is significant, and there is an association between SNP and trait. Otherwise it is not significant, and there is no association between SNP and trait.

Table 7. Part of Output

SNP	MAF	pval_Y1	pval_Y2	pval_full	pval_trait_ specific	pval_trait_ common
1H_95.9	0.4167	0.0275	0.0870	0.0334	0.7611	0.0096
2H_74.40	0.2138	0.02229	0.0579	0.0034	0.0008	0.6962
3H_68.30	0.3085	0.1269	0.0378	0.0164	0.0044	0.7331
4H_54.30	0.2557	0.0113	0.0670	0.0142	0.6803	0.0039
5H_136.80	0.0828	0.0740	0.0244	0.0323	0.6456	0.0099
6H_13.2	0.0855	0.0340	0.0141	0.0011	0.0002	0.9605
7H_70.207	0.4877	0.0123	0.0272	0.0092	0.9076	0.0022
7H_70.8	0.4877	0.0163	0.0506	0.0203	0.8313	0.0054
7H_70.723	0.4886	0.0405	0.0282	0.0246	0.8134	0.0067
3H_75.72	0.1865	0.0169	0.1001	0.0069	0.0021	0.4832
5H_140.11	0.0992	0.0181	0.0533	0.0167	0.8639	0.0043
1H_5.2	0.3057	0.0892	0.0836	0.0253	0.0067	0.9837
1H_122.2	0.4177	0.0273	0.0747	0.0324	0.7362	0.0094
2H_74.42	0.2011	0.0240	0.0684	0.0044	0.0011	0.6867
...						

The statistics of this model is shown in Table 8. The first row of Table 8 is correlation value between two traits. It is 0.0908, means that there is little correlation between two traits. The second row, *gen_cor*, is genetic correlation value between two traits, which is 0.358. This means that there is moderate genetic correlation between two traits. The third row, *env_cor*, is

non-genetic, like environmental correlation, which is 0.0896. This means that there is not much environmental correlation between two traits. The forth row, phen_cor, is the correlation between genetic and non-genetic interaction effect, which is 0.24. This means there is little correlation between genetic and non-genetic interaction effect. The first four rows indicate the correlation between these two traits is mainly caused by genetics compared to non-genetics. The fifth row, h1_joint, is heritability of first trait, which is 0.608. The sixth row, h2_joint, is heritability of second trait, which is 0.525. The heritability of the first trait is higher than the second one, which means that we may need to pay more attention on the first trait. The last row shows that the maximum likelihood estimator is converged in the end.

Table 8. Statistics of Data

pearson	0.09084761
gen_cor	0.3582201
env_cor	0.08955061
phen_cor	0.2410485
h1_joint	0.6083152
h2_joint	0.5248898
converge	TRUE

Manhattan plots are shown in Figure 4 as following. The horizontal line is p value at level 0.01. The first plot is marginal, single-trait analysis for the first trait. The second plot is marginal, single-trait analysis for the second trait. The third plot is joint MTMM analysis with the full model. The fourth one is for interaction effect, and the fifth one is for common effect.

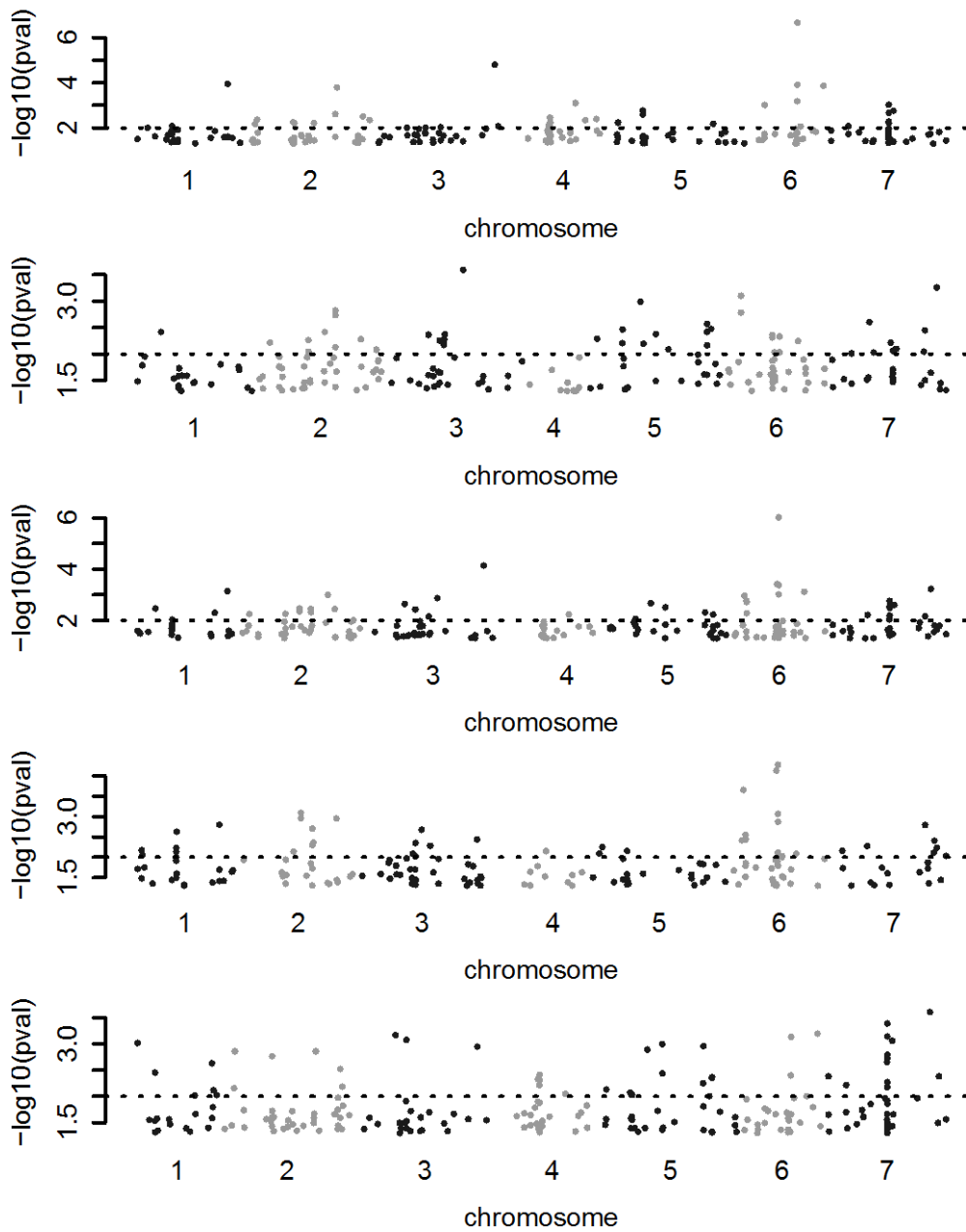


Figure 4. Manhattan Plots

QQ plot of the p-values is shown in Figure 5 as following. The blue line represents marginal, single-trait analysis for the first trait. The yellow line represents marginal, single-trait

analysis for the second trait. The red line represents joint MTMM analysis with full model. The purple line represents joint MTMM analysis with interaction effect. The black line represents joint MTMM analysis with common effect. It shows that neither the marginal, single analysis nor the MTMM test shows evidence of population structure confounding due to population structure.

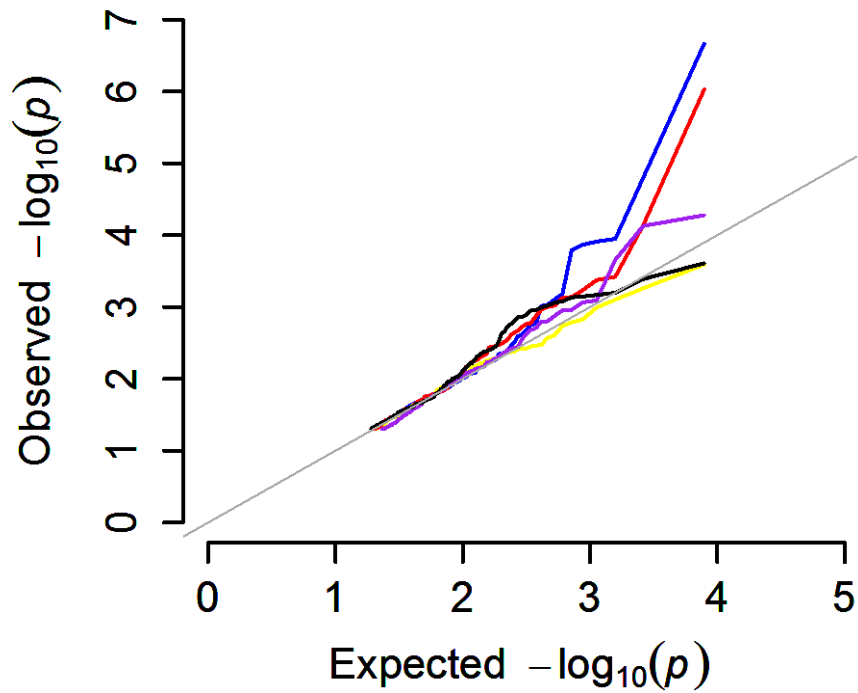


Figure 5. QQ Plot

CHAPTER 5. CONCLUSION AND DISCUSSION

From Table 7, there are several significant SNPs detected. At first locus, SNP: 1H_95.9, the p-values of the two traits are all greater than 0.01, which means that they do not show any significance at level 0.01 in single-trait analysis. On the contrary, MTMM shows its strong significance due to strong common effect. And at seventh locus, SNP: 7H_70.8, the two traits do not show significance at level 0.01 in single-trait analysis, however, it was significant in the MTMM analysis due to strong common effect. Almost every locus, except the fourth locus, we can find some significant SNPs by MTMM analysis while they are not significant in single-trait analysis. Therefore, the multi-trait mixed model can show greater power on detecting the significance of SNPs.

These findings are also supported by Manhattan plots. In general, the strongest associations have the smallest P-values. For example, the association in marginal analysis for the first locus is not clear, but there are highly significant associations using MTMM analysis because of strong common effect. However, for the fourth locus, the significant association is not obvious in this plot. In addition, almost all SNPs that were identified using marginal analysis were also detected using MTMM. Therefore, this plot still shows that the multi-trait mixed model can show greater power on detecting the significance of SNPs.

As we know, missing values were replaced by randomly generated values. However, QQ plot shows that neither the marginal, single analysis nor the MTMM test shows evidence of barley population structure confounding due to population structure. Above all, MTMM has more power to detect the nature of associations between SNPs and traits than marginal analysis does.

From Table 7, we also noticed that the p-values of the two traits in marginal analysis are just a little higher than 0.01, so the advantage of MTMM analysis is not significant. Other dataset with highly correlated phenotypes could be tried to re-apply for this model to show its benefits. Furthermore, only two traits were considered in this research, we can extend this approach to larger multi-trait experiments in the future. We can get lots of benefits in estimating error terms and declaring functional relationship between suits of traits. However, MTMM approach is not meaning to replace marginal analysis, just as a complement to our research.

REFERENCES

- Bolormaa, S., Pryce, J. E., Reverter, A., Zhang, Y., Barendse, W., Kemper, K., . . .
Goddard, M. E. (2014). A Multi-Trait, Meta-analysis for Detecting Pleiotropic Polymorphism for Stature, Fatness and Reproduction in Beef Cattle. *PLoS Genet.*
- Gilmour, A., Gogel, B., Cullis, B., & Thompson, R. (2006). ASReml User Guide Release 2.0.
- Harville, D. A. (1974). Bayesian inference for variance components using only error contrasts. *Biometrika*, 61:383–385.
- Henderson, C. (1984). *Applications of Linear Models in Animal Breeding*. University of Guelph, Guelph, ON.
- Henderson, C., & Quaas, R. (1976). Multiple trait evaluation using relatives' records. *Journal of Animal Science*, 43:1188–1197.
- Kang, H. M., Zaitlen, N. A., Wade, C. M., Kirby, A., Heckerman, D., Daly, M. J., & Eskin, E. (2008). Efficient control of population structure in model organism association mapping. *Genetics*, 178:1709–23.
- Korte, A., Vilhjalmsdottir, B. J., Segura, V., Platt, A., Long, Q., & Nordborg, M. (2012). A mixed-model approach for genome-wide association studies of. *Nature genetics*, 44(9):1066–1071.
- Makowsky, R., Pajewski, N. M., Klimentidis, Y. C., Vazquez, A. I., Duarte, C. W., Allison, D. B., & Campos, G. (2011). Beyond missing. *PLoS genetics*, 7(4):e1002051.
- McCarthy, M., Abecasis, G., Cardon, L., Goldstein, D., Little, J., Ioannidis, J., & Hirschhorn, J. (2008). Genome-wide association studies for complex traits: consensus. *Nat Rev Genet*, 9(5):356–369.

Sivakumaran, S., Agakov, F., Theodoratou, E., Prendergast, J. G., Zgaga, L., Manolio, T., . . . Campbell, H. (2011). Abundant pleiotropy in human complex diseases and traits. *The American Journal of Human Genetics*, 89(5):607–618.

Yang, J., Benyamin, B., McEvoy, B. P., Gordon, S., Henders, A. K., Nyholt, D. R., . . . Visscher, P. M. (2010). Common snps explain a large proportion of the heritability for human height. *Nature genetics*, 42(7):565–569.

APPENDIX A. MTMM

```
# Read genotype Data #
X<-read.csv("genotype.csv",header=T,sep=" ",row.names=1)
# If name starts with a number, it is shown in R in the format "X+number" #
# Need remove X in column names #
colnames(X)<- gsub("X","",colnames(X) )
# Pick up first 1100 individuals #
X<-X[1:1100,]
X<-as.matrix(X)
source('emma.r')
# Calculate IBS by EMMA package #
R<-emma.ibs(t(X))
# Save IBS data as a csv file #
write.csv(R,file="IBS.csv",row.names=row.names(X))
R<-read.table("IBS.csv",header=T,sep=" ",row.names=1)
# Make sure rownames(R)=colnames(R)=individual names #
colnames(R)<-rownames(R)
Y<-read.table("phenotype.csv",header=T,sep=" ",row.names=1)
save(Y,X,R,file="MTMM.Rdata")
## MTMM[Korte et al., 2012] functions for GWAS in barley population ##
## requires the R-package AsREML which needs a valid license ##
# AsREML package: Fit the linear mixed model(estimates variance components under a #
general linear mixed model by residual maximum likelihood) #
```

```

load("MTMM.Rdata")

Sys.setenv("ASREML_LICENSE_FILE= ")

library(asreml)

library(emma)

library(lattice)

source('emma.r')

## detect duplicated y ##

if(anyDuplicated(rownames(Y))>0) { cat('duplicated ecotypes detected', '\n')

stopifnot(anyDuplicated(rownames(Y))==0)}

## create SNP in specific format ##

options(stringsAsFactors = FALSE)

cat('SNP_INFO file created', '\n')

SNP_INFO<-

data.frame(cbind(colnames(X),matrix(nrow=ncol(X),ncol=2,data=unlist(strsplit(colnames(X),spl

it='H_')),byrow=T)))

colnames(SNP_INFO)<-c('SNP','Chr','Pos')

SNP_INFO[,2]<-as.numeric(SNP_INFO[,2])

SNP_INFO[,3]<-as.numeric(SNP_INFO[,3])

## remove missing y ##

Y_<-na.omit(Y)

cat(nrow(Y)-nrow(Y_), 'values excluded, leaving', nrow(Y_), 'values', '\n')

stopifnot(nrow(Y_)==nrow(Y))

## preparing genotype and phenotype data ##

```



```

## make sure rownames(X)=rownames(Y)=rownames(R)=colnames(R)=individual names ##
X<-X[which(rownames(X)%in%rownames(Y_)),]
Y<-Y_[which(rownames(Y_)%in%rownames(X)),]
### enter here filtering step for 2 different traits ##
Y1<-(Y[,1])
Y2<-(Y[,2])

## check normality for the first trait ##
plot(Y1,dnorm(Y1,mean=mean(Y1),sd=sd(Y1)),xlab="Rate",ylab="Density",main="First trait
Distribution",xlim=c(0,5),ylim=c(0,0.6))

## add a normal curve for comparison ##
s1<-rnorm(1099,mean=mean(Y1),sd=sd(Y1))
par(new=T)
plot(sort(s1),dnorm(sort(s1),mean=mean(Y1),sd=sd(Y1)),col="red",xlab="",ylab="",axes=F,xlim=
c(0,5),ylim=c(0,0.6),type="l")

## check normality for the second trait ##
plot(Y2,dnorm(Y2,mean=mean(Y2),sd=sd(Y2)),xlab="Rating",ylab="Density",main="Second
trait Distribution",xlim=c(2,10),ylim=c(0,0.5))

s2<-rnorm(1099,mean=mean(Y2),sd=sd(Y2))
par(new=T)
plot(sort(s2),dnorm(sort(s2),mean=mean(Y2),sd=sd(Y2)),col="red",xlab="",ylab="",axes=F,xlim=
c(2,10),ylim=c(0,0.5),type='l')
names(Y1)<-rownames(Y)
names(Y2)<-rownames(Y)

```

```

Y1_<-na.omit(Y1)
Y2_<-na.omit(Y2)
ecot_id1<-names(Y1_)
ecot_id2<-names(Y2_)
## preparing IBS data ##
R<-as.matrix(R[which(rownames(R)%in%names(Y1)),which(colnames(R)%in%names(Y1))])
R1<-as.matrix(R[which(rownames(R)%in%ecot_id1),which(colnames(R)%in%ecot_id1)])
R2<-as.matrix(R[which(rownames(R)%in%ecot_id2),which(colnames(R)%in%ecot_id2)])
n<-nrow(Y)
n1<-length(ecot_id1)
n2<-length(ecot_id2)
## combine the traits ##
Y_ok<-c(Y1,Y2)
plot(Y_ok,dnorm(Y_ok,mean=mean(Y_ok),sd=sd(Y_ok)),xlab="Combined
Trait",ylab="Density",main="Combined trait Distribution",xlim=c(0,10),ylim=c(0,0.2))
s3<-rnorm(2198,mean=mean(Y_ok),sd=sd(Y_ok))
par(new=T)
plot(sort(s3),dnorm(sort(s3),mean=mean(Y_ok),sd=sd(Y_ok)),col="red",xlab="",ylab="",axes=F,x
lim=c(0,10),ylim=c(0,0.25),type='l')
## initialize environment in R ##
Env<-c(rep(0,n),rep(1,n))
## standardize IBS data ##
R_stand<-(-n-1)/sum((diag(n)-matrix(1,n,n)/n)*R)*R

```

```

R_inv<-solve(R_stand)

dimnames(R_inv)<-dimnames(R)

R_stand1<-(n1-1)/sum((diag(n1)-matrix(1,n1,n1)/n1)*R1)*R1

R_stand2<-(n2-1)/sum((diag(n2)-matrix(1,n2,n2)/n2)*R2)*R2

## calculate MAF&MAC ##

## SNPs are expected not to be uniform in the population. The MAF will be used here to #
exclude SNPs. If MAF=0, then delete that SNP. ##

AC_1<- data.frame(colnames(X),apply(X,2,sum))

colnames(AC_1)<-c('SNP','AC_1')

MAF_1<-data.frame(AC_1,AC_0=nrow(X)-AC_1$AC_1)

MAF_2<-data.frame(MAF_1,MAC=apply(MAF_1[,2:3],1,min))

MAF_3<-data.frame(MAF_2,MAF=(MAF_2$MAC/nrow(X)))

MAF_ok<-merge(SNP_INFO,MAF_3,by='SNP')

rm(AC_1,MAF_1,MAF_2,MAF_3)

## Filter for MAF ##

MAF<-subset(MAF_ok,MAF==0)[,1]

X_ok<-X[,!colnames(X) %in% MAF]

rm(MAF)

## analysis single trait y1 and y2 ##

Xo1<-rep(1,n1)

ex1<-as.matrix(Xo1)

Xo2<-rep(1,n2)

ex2<-as.matrix(Xo2)

```

```

## compute heritability for each trait ##
null1<-emma.REMLE(Y1_,ex1,R_stand1)
herit1<-null1$vg/(null1$vg+null1$ve)
null2<-emma.REMLE(Y2_,ex2,R_stand2)
herit2<-null2$vg/(null2$vg+null2$ve)
X1<-X_ok[rownames(X_ok)%in%names(Y1_),]
count<-function(x) {length(unique(x))}
AC <- data.frame(colnames(X1),apply(X1,2,count))
colnames(AC)<-c('SNP','AC')
rm1<-which(AC$AC==1)
if (!length(rm1)==T) {X_ok1<-X1
}else {
X_ok1<-X1[,-rm1]}
M1<-solve(chol(null1$vg*R_stand1+null1$ve*diag(dim(R_stand1)[1])))
Y_t1<-crossprod(M1,Y1_)
int_t1<-crossprod(M1,(rep(1,n1)))
X_t1<-crossprod(M1,X_ok1)
## single analysis model for first trait ##
models1<-apply(X_t1,2,function(x){summary(lm(Y_t1~0+int_t1+x))$coeff[2,]})
out_models1<-
data.frame(SNP=colnames(models1),Pval_Y1=models1[4,],beta_Y1=models1[1,])
X2<-X_ok[rownames(X_ok)%in%names(Y2_),]
AC <- data.frame(colnames(X1),apply(X1,2,count))

```

```

colnames(AC)<-c('SNP','AC')
rm2<-which(AC$AC==1)
if (!length(rm2)==T) {X_ok2<-X2
}else {
X_ok2<-X2[,-rm2]}
M2<-solve(chol(null2$vg*R_stand2+null2$ve*diag(dim(R_stand2)[1])))
Y_t2<-crossprod(M2,Y2_)
int_t2<-crossprod(M2,(rep(1,n2)))
X_t2<-crossprod(M2,X_ok2)
## single analysis model for second trait ##
models2<-apply(X_t2,2,function(x){summary(lm(Y_t2~0+int_t2+x))$coeff[2,]})
out_models2<-
data.frame(SNP=colnames(models2),Pval_Y2=models2[4,],beta_Y2=models2[1,])
##### START MTMM #####
## intercept ##
Xo<-rep(1,2*n)
## preparing the data ##
data_asr<-data.frame(ecot_id=c(ecot_id1,ecot_id2),Env=Env,Y_ok=Y_ok)
data_asr$ecot_id<-as.factor(data_asr$ecot_id)
data_asr$Env<-as.factor(data_asr$Env)
## setting starting values ##
sigma1<-null1$vg
sigma2<-null2$vg

```

```

rho<-cor(Y1,Y2)/(sqrt(herit1)*sqrt(herit2))
## set borders for rho between -1 and 1 ##
if (rho>0.98) {rho<-0.98
}else { if (rho< -0.98) {rho<--0.98}}
alpha<-rho*sqrt(null1$vg*null2$vg)
beta1<-cor(Y1,Y2)*sqrt(null1$vg*null2$vg)
beta2<-cor(Y1,Y2)*sqrt(null1$ve*null2$ve)
## including autocorrelation ##
mixmod.asr<-
asreml(fixed=Y_ok~Env,random=~us(Env,init=c(sigma1,beta1,sigma2)):giv(ecot_id),ginverse=1
ist(ecot_id=R_inv),rcov=~us(Env,init=c(null1$ve,beta2,null2$ve)):id(ecot_id),data=data_asr,ma
xit=30)
varcov<-
matrix(data=c(summary(mixmod.asr)$varcomp$component[1:2],summary(mixmod.asr)$varcom
p$component[2:3]),nrow=2,ncol=2)
ve<-
matrix(data=c(summary(mixmod.asr)$varcomp$component[5:6],summary(mixmod.asr)$varcom
p$component[6:7]),nrow=2,ncol=2)
}
rho_g<-varcov[1,2]/sqrt(varcov[1,1]*varcov[2,2])
rho_e<-ve[1,2]/sqrt(ve[1,1]*ve[2,2])
rho_p<-((varcov[1,2]+ve[1,2])/sqrt((varcov[1,1]+ve[1,1])*(varcov[2,2]+ve[2,2])))
pearson<-cor(Y1,Y2)

```

```

heriti1<-varcov[1,1]/(varcov[1,1]+ve[1,1])
heriti2<-varcov[2,2]/(varcov[2,2]+ve[2,2])
correlation<-
list(pearson=pearson,gen_cor=rho_g,env_cor=rho_e,phen_cor=rho_p,h1_joint=heriti1,h2_joint=
heriti2,converge=mixmod.asr$converge)
#extract variances from asreml output
R_comb<-kronecker(varcov,R_stand)
rownames(R_comb)<-c(ecot_id1,ecot_id2)
colnames(R_comb)<-c(ecot_id1,ecot_id2)
I_comb<-kronecker(ve,diag(n))
rownames(I_comb)<-rownames(R_comb)
colnames(I_comb)<-colnames(R_comb)
bigR<-R_comb+I_comb
M<-solve(chol(bigR))
X_ok1<-X_ok*sqrt(varcov[1,1])
X_ok2<-X_ok*sqrt(varcov[2,2])
Y_t<-crossprod(M,Y_ok)
cof_t<-crossprod(M,cbind(Xo,Env))
RSS_env<-sum(lsfrit(cof_t,Y_t,intercept = FALSE)$residuals^2)
nbchunks<-5
m<-ncol(X_ok)
RSS_full<-list()
RSS_glob<-list()

```

```

for (j in 1:(nbchunks-1)){
X_<-rbind(X_ok1[,(j-1)*round(m/nbchunks)+1):(j*round(m/nbchunks)],X_ok2[,(j-
1)*round(m/nbchunks)+1):(j*round(m/nbchunks))]
X_t<-array(dim=c(nrow(X_),ncol(X_),2))
X_t[,1]<-crossprod(M,X_)
X_t[,2]<-crossprod(M,(Env*X_))
RSS_full[[j]]<-
apply(X_t,2,function(x){sum(lsf(cbind(cof_t,x),Y_t,intercept=FALSE)$residuals^2)})
RSS_glob[[j]]<-
apply(X_t[,1],2,function(x){sum(lsf(cbind(cof_t,x),Y_t,intercept=FALSE)$residuals^2)})
rm(X_,X_t)}
X_<-rbind(X_ok1[,(j)*round(m/nbchunks)+1]:m],X_ok2[,(j)*round(m/nbchunks)+1]:m])
X_t<-array(dim=c(nrow(X_),ncol(X_),2))
X_t[,1]<-crossprod(M,X_)
X_t[,2]<-crossprod(M,(Env*X_))
RSS_full[[nbchunks]]<-
apply(X_t,2,function(x){sum(lsf(cbind(cof_t,x),Y_t,intercept=FALSE)$residuals^2)})
RSS_glob[[nbchunks]]<-
apply(X_t[,1],2,function(x){sum(lsf(cbind(cof_t,x),Y_t,intercept=FALSE)$residuals^2)})
rm(X_,X_t,j)
RSS_full<-unlist(RSS_full)
RSS_glob<-unlist(RSS_glob)
## number of parameters used in each models ##

```



```

par_env<-ncol(cof_t)
par_glob<-par_env+1
par_full<-par_glob+1
##F-TESTS ## FULL vs NULL##
## full test: full model V.S. null model  $\beta=0$  and  $\alpha=0$ ##
F_full<-(rep(RSS_env,m)/RSS_full-1)*(2*n-par_full)/(par_full-par_env)
## interaction effect test: full model V.S. null model  $\alpha=0$  ##
F_ge<-(RSS_glob/RSS_full-1)*(2*n-par_full)/(par_full-par_glob)
## common effect test: genetic model  $\alpha=0$  V.S. null model  $\beta=0$  and  $\alpha=0$  ##
F_glob<-(rep(RSS_env,m)/RSS_glob-1)*(2*n-par_glob)/(par_glob-par_env)
pval_full<-pf(F_full,par_full-par_env,2*n-par_full,lower.tail=FALSE)
pval_ge<-pf(F_ge,par_full-par_glob,2*n-par_full,lower.tail=FALSE)
pval_glob<-pf(F_glob,par_glob-par_env,2*n-par_glob,lower.tail=FALSE)
## outputs ##
outfull<-data.frame('SNP'=colnames(X_ok),pval=pval_full)
outge<-data.frame('SNP'=colnames(X_ok),pval=pval_ge)
outglob<-data.frame('SNP'=colnames(X_ok),pval=pval_glob)
data.out<-
merge(MAF_ok,cbind(out_models1[1:2],out_models2[,2],outfull[,2],outge[,2],outglob[,2],out_m
odels1[,3],out_models2[,3]),by='SNP')
colnames(data.out)[8:14]<-
c('pval_Y1','pval_Y2','pval_full','pval_trait_specific','pval_trait_common','beta_Y1','beta_Y2')
data.out_<-data.out[order(data.out[,3]),]

```

```

out<-data.out_[order(data.out_[,2],)]
results<-list(phenotype=Y,pvals=out,statistics=correlation,relationship=R)
output<-results$pvals
##### function for plotting #####
## Manhattan plots ##
plot_gwas<-function(output,h=8,maf=0.05,black=T) {
  colnames(output)[h]<-'Pval'
  new_<-subset(output,output$MAF>maf&output$Pval<0.05)
  vu<-ceiling(-log10(min(new_[,8]))+1)
  output_<-new_[order(new_$Pos),]
  output_ok<-output_[order(output_$Chr),]
  maxpos<-
c(0,cumsum(aggregate(output_ok$Pos,list(output_ok$Chr),max)$x+max(cumsum(aggregate(out
put_ok$Pos,list(output_ok$Chr),max)$x))/100))
  if(black==T) {
    plot_col<-rep(c('gray10','gray60'),ceiling(max(unique(output_ok$Chr))/2))
  } else { plot_col<-c('blue','darkgreen','red','cyan','purple')}
  size<-aggregate(output_ok$Pos,list(output_ok$Chr),length)$x
  a<-rep(maxpos[1],size[1])
  b<-rep(plot_col[1],size[1])
  for (i in 2:length(unique(output_ok$Chr))){
    a<-c(a,rep(maxpos[i],size[i]))
    b<-c(b,rep(plot_col[i],size[i]))}

```

```

output_ok$xpos<-output_ok$Pos+a
output_ok$col<-b
d<-
(aggregate(output_ok$xpos,list(output_ok$Chr),min)$x+aggregate(output_ok$xpos,list(output_o
k$Chr),max)$x)/2
plot(output_ok$xpos,-log10(output_ok$Pval),col=output_ok$col,pch=16,ylab='-
log10(pval)',xaxt='n',xlab='chromosome',axes=F,cex=0.8)
axis(1,tick=FALSE,at=d,labels=c(1:length(unique(output_ok$Chr))))
axis(2,lwd=2)
abline(h=-log10(0.01),lty=3,col='black',lwd=2)
}
## QQ plot ##
qq_plot_all<-
function(output,h=c(8,9,10,11,12),maf=0.05,farbe=c('blue','yellow','red','purple','black')) {
maxp<-ceiling(max(-log10(output[,h])))
for (z in 1: length(h)) {
colnames(output)[h[[z]]<-'Pval'
out<-subset(output,output$MAF>maf&output[,h[[z]]<0.05)
e<--log10(ppoints(nrow(output)))[1:nrow(out)]
o<--log10(sort(out[,h[[z]]))
plot(e,o,type='l',cex=0.8,col=farbe[[z]],xlab=expression(Expected~~-log[10](italic(p))),
ylab=expression(Observed~~-
log[10](italic(p))),xlim=c(0,max(e)+1),ylim=c(0,maxp),lwd=2,axes=FALSE)

```

```
abline(0,1,col="dark grey")

axis(1,lwd=2)

axis(2,lwd=2)

par(new=T)}

}

par(mfrow=c(5,1),mar=c(4,4,1,4))

## marginal analysis for first trait ##

plot_gwas(output,h=8)

## marginal analysis for first trait ##

plot_gwas(output,h=9)

## MTMM analysis with full model ##

plot_gwas(output,h=10)

## MTMM analysis with interaction effect ##

plot_gwas(output,h=11)

## MTMM analysis with common effect ##

plot_gwas(output,h=12)

## QQ Plot ##

par(mfrow=c(1,1),mar=c(4,5,1,3))

qq_plot_all(output)
```

APPENDIX B. EMMA FUNCTION

```
## EMMA function [Korte et al., 2012] for calculating IBS matrix and single-trait analysis ##
emma.ibs <- function(snps, method="additive", use="all") {
  n0 <- sum(snps==0,na.rm=TRUE)
  nh <- sum(snps==0.5,na.rm=TRUE)
  n1 <- sum(snps==1,na.rm=TRUE)
  nNA <- sum(is.na(snps))
  stopifnot(n0+nh+n1+nNA == length(snps))
  if ( method == "dominant" ) {
    flags <- matrix(as.double(rowMeans(snps,na.rm=TRUE) > 0.5),nrow(snps),ncol(snps))
    snps[!is.na(snps) & (snps == 0.5)] <- flags[!is.na(snps) & (snps == 0.5)]
  }
  else if ( method == "recessive" ) {
    flags <- matrix(as.double(rowMeans(snps,na.rm=TRUE) < 0.5),nrow(snps),ncol(snps))
    snps[!is.na(snps) & (snps == 0.5)] <- flags[!is.na(snps) & (snps == 0.5)]
  }
  else if ( ( method == "additive" ) && ( nh > 0 ) ) {
    dsnp <- snps
    rsnp <- snps
    flags <- matrix(as.double(rowMeans(snps,na.rm=TRUE) > 0.5),nrow(snps),ncol(snps))
    dsnp[!is.na(snps) & (snps==0.5)] <- flags[!is.na(snps) & (snps==0.5)]
    flags <- matrix(as.double(rowMeans(snps,na.rm=TRUE) < 0.5),nrow(snps),ncol(snps))
    rsnp[!is.na(snps) & (snps==0.5)] <- flags[!is.na(snps) & (snps==0.5)]
  }
}
```

```

snps <- rbind(dsnps,rsnps)
}
if ( use == "all" ) {
  mafs <- matrix(rowMeans(snps,na.rm=TRUE),nrow(snps),ncol(snps))
  snps[is.na(snps)] <- mafs[is.na(snps)]
}
else if ( use == "complete.obs" ) {
  snps <- snps[rowSums(is.na(snps))==0,]
}
n <- ncol(snps)
R <- matrix(nrow=n,ncol=n)
diag(R) <- 1
for(i in 2:n) {
  for(j in 1:(i-1)) {
    x <- snps[,i]*snps[,j] + (1-snps[,i])*(1-snps[,j])
    R[i,j] <- sum(x,na.rm=TRUE)/sum(!is.na(x))
    R[j,i] <- R[i,j]
  }
}
return(R)
}
emma.REMLE <- function(y, X, R, Z=NULL, ngrids=100, llim=-10, ulim=10,
  esp=1e-10, eig.L = NULL, eig.R = NULL) {

```

```

n <- length(y)
t <- nrow(R)
q <- ncol(X)
# stopifnot(nrow(R) == t)
stopifnot(ncol(R) == t)
stopifnot(nrow(X) == n)
if ( det(crossprod(X,X)) == 0 ) {
  warning("X is singular")
  return (list(REML=0,delta=0,ve=0,vg=0))
}
if ( is.null(Z) ) {
  if ( is.null(eig.R) ) {
    eig.R <- emma.eigen.R.wo.Z(R,X)
  }
  etas <- crossprod(eig.R$vectors,y)
  logdelta <- (0:ngrids)/ngrids*(ulim-llim)+llim
  m <- length(logdelta)
  delta <- exp(logdelta)
  Lambdas <- matrix(eig.R$values,n-q,m) + matrix(delta,n-q,m,byrow=TRUE)
  Etasq <- matrix(etas*etas,n-q,m)
  LL <- 0.5*((n-q)*(log((n-q)/(2*pi))-1-log(colSums(Etasq/Lambdas)))-
colSums(log(Lambdas)))

```

```

dLL <- 0.5*delta*((n-q)*colSums(Etasq/(Lambdas*Lambdas))/colSums(Etasq/Lambdas)-
colSums(1/Lambdas))

optlogdelta <- vector(length=0)

optLL <- vector(length=0)

if ( dLL[1] < esp ) {

  optlogdelta <- append(optlogdelta, llim)

  optLL <- append(optLL, emma.delta.REML.LL.wo.Z(llim,eig.R$values,etas))

}

if ( dLL[m-1] > 0-esp ) {

  optlogdelta <- append(optlogdelta, ulim)

  optLL <- append(optLL, emma.delta.REML.LL.wo.Z(ulim,eig.R$values,etas))

}

for( i in 1:(m-1) )

{

  if ( ( dLL[i]*dLL[i+1] < 0-esp*esp ) && ( dLL[i] > 0 ) && ( dLL[i+1] < 0 ) )

  {

    r <- uniroot(emma.delta.REML.dLL.wo.Z, lower=logdelta[i], upper=logdelta[i+1],
lambda=eig.R$values, etas=etas)

    optlogdelta <- append(optlogdelta, r$root)

    optLL <- append(optLL, emma.delta.REML.LL.wo.Z(r$root,eig.R$values, etas))

  }

}

# optdelta <- exp(optlogdelta)

```



```

}
else {
  if ( is.null(eig.R) ) {
    eig.R <- emma.eigen.R.w.Z(Z,R,X)
  }
  etas <- crossprod(eig.R$vector,y)
  etas.1 <- etas[1:(t-q)]
  etas.2 <- etas[(t-q+1):(n-q)]
  etas.2.sq <- sum(etas.2*etas.2)
  logdelta <- (0:ngrids)/ngrids*(ulim-llim)+llim
  m <- length(logdelta)
  delta <- exp(logdelta)
  Lambdas <- matrix(eig.R$values,t-q,m) + matrix(delta,t-q,m,byrow=TRUE)
  Etasq <- matrix(etas.1*etas.1,t-q,m)
  dLL <- 0.5*delta*((n-
q)*(colSums(Etasq/(Lambdas*Lambdas))+etas.2.sq/(delta*delta))/(colSums(Etasq/Lambdas)+et
as.2.sq/delta)-(colSums(1/Lambdas)+(n-t)/delta))
  optlogdelta <- vector(length=0)
  optLL <- vector(length=0)
  if ( dLL[1] < esp ) {
    optlogdelta <- append(optlogdelta, llim)
    optLL <- append(optLL, emma.delta.REML.LL.w.Z(llim,eig.R$values,etas.1,n,t,etas.2.sq))
  }
}

```

```

if ( dLL[m-1] > 0-esp ) {
  optlogdelta <- append(optlogdelta, ulim)
  optLL <- append(optLL, emma.delta.REML.LL.w.Z(ulim,eig.R$values,etas.1,n,t,etas.2.sq))
}
for( i in 1:(m-1) )
{
  if ( ( dLL[i]*dLL[i+1] < 0-esp*esp ) && ( dLL[i] > 0 ) && ( dLL[i+1] < 0 ) )
  {
    r <- uniroot(emma.delta.REML.dLL.w.Z, lower=logdelta[i], upper=logdelta[i+1],
lambda=eig.R$values, etas.1=etas.1, n=n, t1=t, etas.2.sq = etas.2.sq )
    optlogdelta <- append(optlogdelta, r$root)
    optLL <- append(optLL, emma.delta.REML.LL.w.Z(r$root,eig.R$values, etas.1, n, t,
etas.2.sq ))
  }
}
# optdelta <- exp(optlogdelta)
}
maxdelta <- exp(optlogdelta[which.max(optLL)])
maxLL <- max(optLL)
if ( is.null(Z) ) {
  maxva <- sum(etas*etas/(eig.R$values+maxdelta))/(n-q)
}
else {

```

```
maxva <- (sum(etas.1*etas.1/(eig.R$values+maxdelta))+etas.2.sq/maxdelta)/(n-q)
}
maxve <- maxva*maxdelta
return (list(REML=maxLL,delta=maxdelta,ve=maxve,vg=maxva))
}
```