Sarah Kostick
CROP_SCI545 HW5

Part (1) The purpose of part 1 was to compare the results of the GLM GWAS package developed as part of homework 4 with GLM in GAPIT. It was hypothesized that both packages should give about the same results since both packages were using GLM method for GWAS. The methods followed are outlined below.

Methods:

1. Simulated phenotype and sampled 10 QTNs (heritability = 0.75) from normal distribution using the G2P function.
2. Using GLM package developed by Kostick and Roeder
    a. PCA analysis
    b. GWAS using GWASbyGLM function. First 3 principle components were included in the model.
    c. Calculated cutoff p-value using Bonferroni.cutoff() function.
    d. Plotted QQplot
    e. Calculated the number of significant SNPs
    f. Determined the number of QTNs detected and the number of false positives
    g. Plotted results using a Manhattan plot
3. Carried out GWAS by GLM using GAPIT.

Results:

As expected, the results from the package developed in HW4 and the results from GAPIT were very similar. Most of the differences observed between the two can be attributed to the scale used on the y-axis. For example, the QQplot from GAPIT only goes to 15 whereas the QQplot from our package goes to 40 (Figure 1). Both packages successfully identified QTN on chromosomes 1, 2, and 9 with a false positive on chromosome 7 (Figure 2). The conclusion that can be drawn is that GWAS using GLM is equivalent regardless of whether you use GAPIT or the package developed in HW4.

Figure 1: QQplots for GWAS by GLM using the package developed in HW4 (left graph) and GAPIT (right graph).

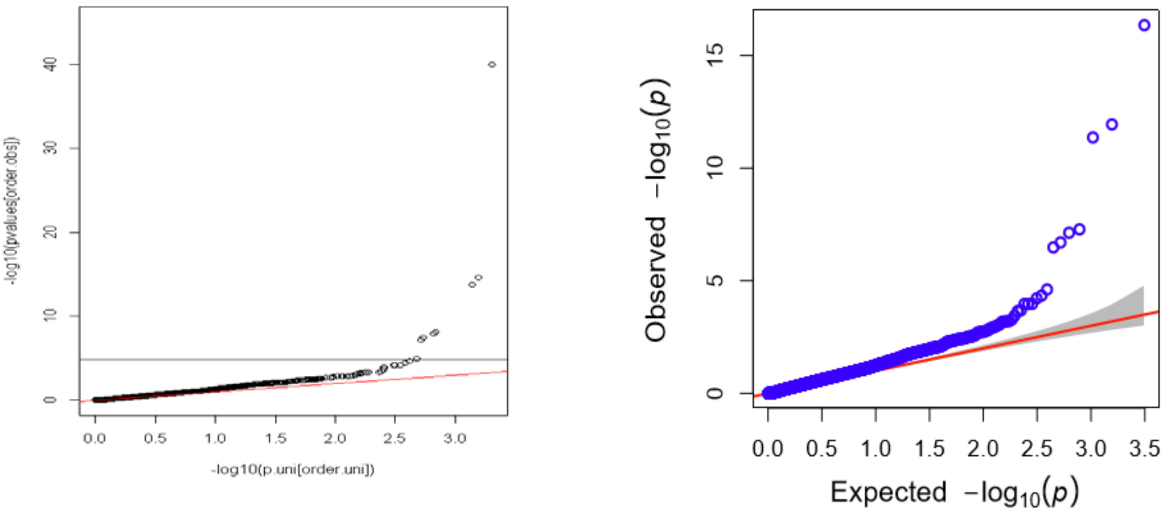
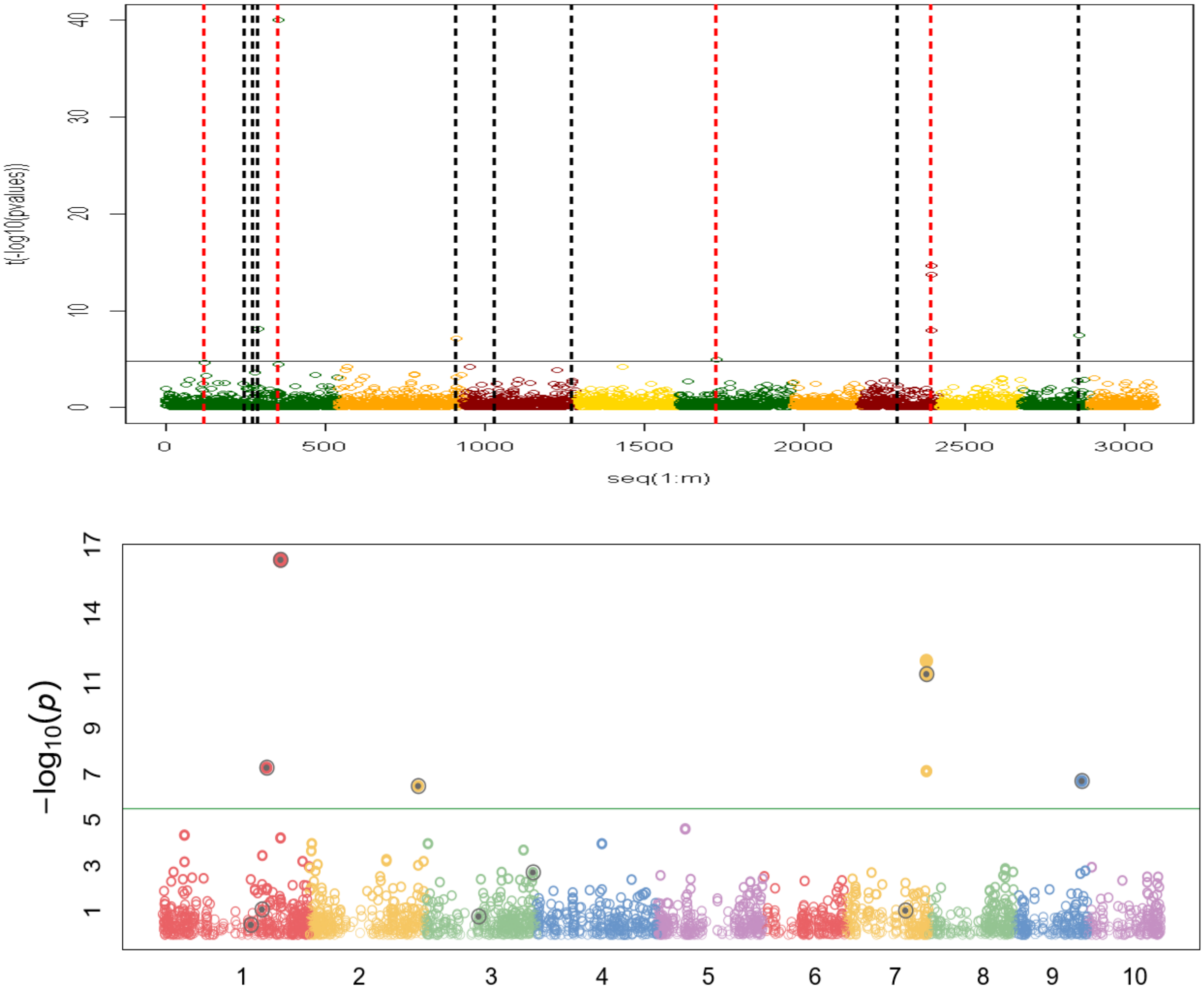


Figure 2: Manhattan plots for GWAS by GLM using package in HW4 (top graph) and GAPIT (bottom graph)



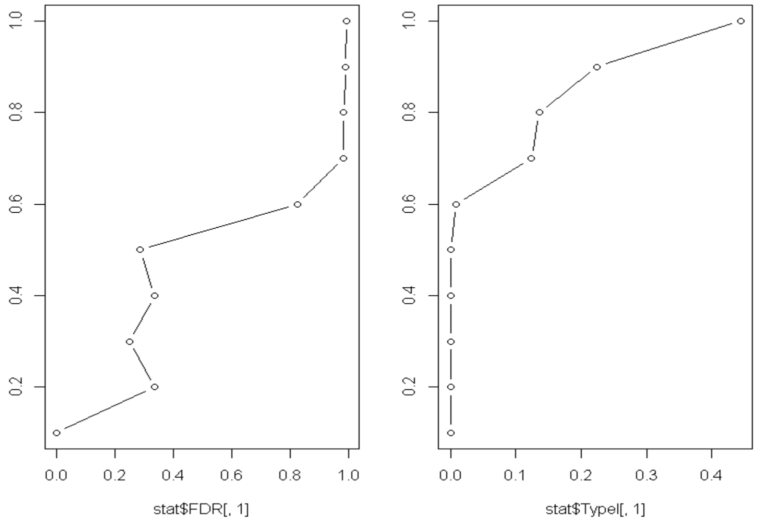
Part (2) The purpose of part 2 was to examine the statistical power vs. FDR and Type I error at a mapping resolution of 1 bp. It was hypothesized that with increasing FDR and Type I error would result in increased power. The methods that were followed are outlined below.

Methods:

1. Used the GAPIT.FDR.TypeI function in GAPIT.
2. Plotted Power against FDR as well as Power against Type I error.

Results: As was expected, as FDR and Type I error increased so did the power (Figure 3). One interesting thing that was observed was that for 2 to 4 QTNs, the associated FDR appears to increase and then decrease (Figure 3). Most likely this is just due to the fact that this is only for one replication. If the mean of more replicates were examined, we would probably see a smoothing of the Power against FDR curve. For Type I error, for 6 QTNs there was not a major increase in Type I error (Figure 3).

Figure 3: Power against FDR (left graph) and power against Type I error (right graph)



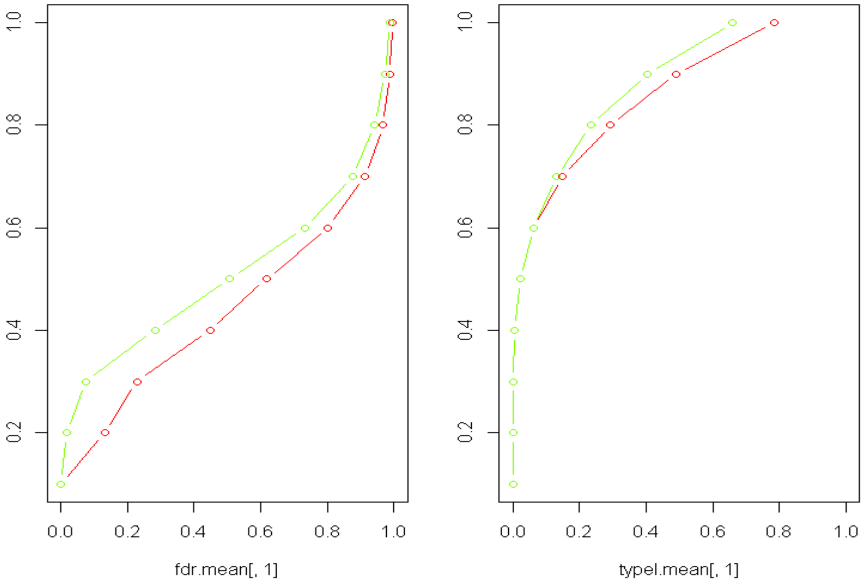
Part (3) The goal of part 3 was to repeat part 2 100 times and compare mapping resolutions of 1 bp and 100 KB. It was hypothesized that a mapping resolution of 100KB would have higher power than 1 bp resolution. The methods that were followed are outlined below.

Methods:

1. Simulated phenotype with a heritability of 0.75 and 10 QTN with the G2P function.
2. Used the GAPIT function with 3 principle components to do GWAS by GLM.
3. Used the GAPIT.FDR.TYPEI function with 1 bp and 100KB window sizes to calculate FDR and Type I error.
4. Repeated 1 through 3 100 times.
5. Calculated mean FDR and Type I errors.
6. Plotted Power against FDR and Power against Type I error.

Results: As expected, for both the power against FDR and power against Type I error graphs, the 100KB resolution had a slightly higher area under the curve (Figure 4). The conclusion that can be drawn from this is that a mapping resolution of 100KB can give you slightly more statistical power than a mapping resolution of 1 bp.

Figure 4: Power against average FDR (left graph) and power against average Type I error (right graph) for window sizes of 1 bp (red line) and 100 KB (green line), for 100 replicates.



Part (4) The aim of part 4 was to compare statistical power between GLM, MLM and SUPER methods of GWAS in GAPIT at a resolution of 100 KB. It was hypothesized that the SUPER method would have the highest relative power at a given FDR or Type I error mean. The methods followed are outlined below.

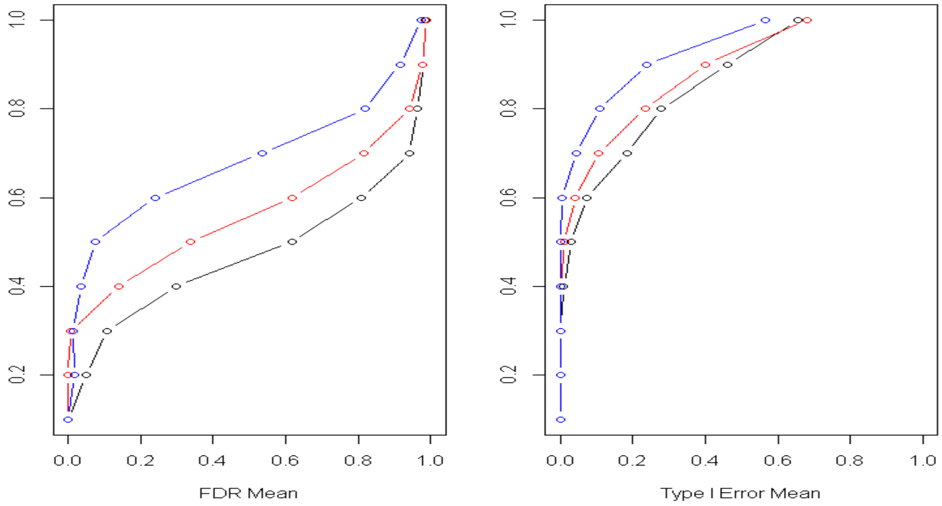
Methods:

For each method (i.e. GLM, MLM, and SUPER)

1. Phenotype was simulated using G2P function.
2. GWAS was done using GAPIT at a window size of 100 KB.
3. FDR and Type I error were determined using GAPIT.FDR.TypeI function.
4. Process was repeated for 30 replicates.
5. Power was plotted against mean FDR and against mean Type I error for each method.

Results: As can be observed in Figure 5, GLM, MLM and Super methods were relatively equivalent in regards to FDR for the first 2 QTN and diverged from each other for 3 to 5 QTNs with the SUPER method performing better than the other two methods (Figure 5). In other words, SUPER method had a larger area under the curve (AUC) than the other methods (Figure 5). However, in the FDR plot, one can observe that the methods begin to converge, FDR mean of around 1.00 (Figure 5). Similarly for power against Type I error, all three methods performed similarly initially but began to diverge with SUPER having the largest AUC (Figure 5). What can be concluded is that in regards to statistical power, the SUPER method performs better than both GLM and MLM.

Figure 5: Power against average FDR (left graph) and power against average Type I error (right graph) for GLM method (black line), MLM (red line), and SUPER (blue line).



Part (5) The goal of part 5 was to compare statistical power of MLMM and FarmCPU at a resolution of 100 KB. It was hypothesized that the statistical power of FarmCPU would be higher than MLMM. The methods used to test this hypothesis are outlined below.

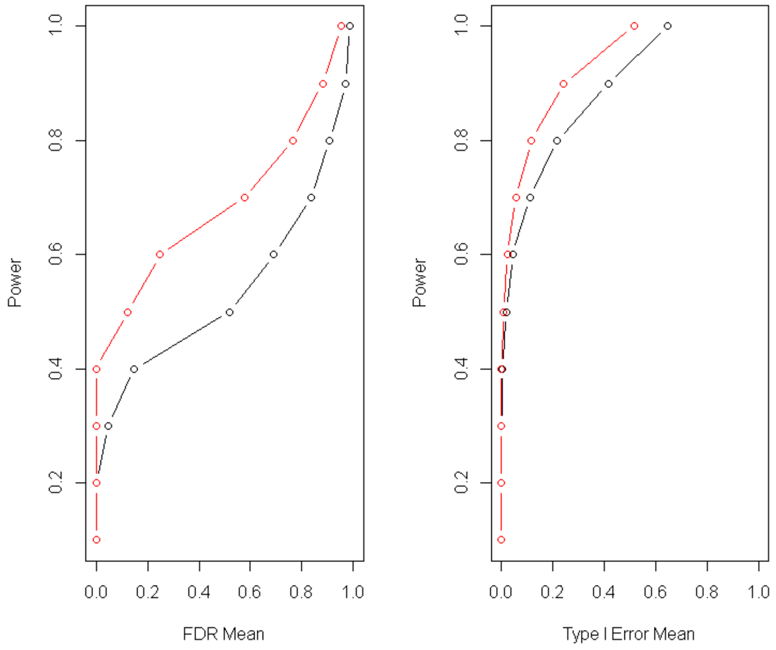
Methods:

For each method (i.e. MLMM and FarmCPU)

1. Phenotype was simulated with 10 QTN using the G2P function.
2. GWAS was done using FarmCPU or MLMM packages, depending upon the method of interest.
3. FDR and Type I error were determined using the GAPIT.FDR.TypeI function.
4. Process was repeated for 30 replications for each method.
5. Power was plotted against mean FDR and mean Type I error for each method.

Results: As expected, FarmCPU performed better, in regards to statistical power, than MLMM (Figure 6). FarmCPU ran a lot faster than MLMM. For power against FDR, FarmCPU and MLMM were relatively equivalent initially, FD mean of around 0.0, but began to diverge with Farm CPU having higher power and lower FDR than MLMM (Figure 6). In other words, the FarmCPU graph had a larger AUC than MLMM. However, the lines for the two methods began to converge, FDR mean of around 1.0 (Figure 6). For Type I error, the two methods were relatively equivalent (Figure 6). However, the graphs began to diverge with FarmCPU having higher statistical power (greater AUC) than MLMM (Figure 6).

Figure 6: Power against average FDR (left graph) and power against average Type I error (right graph) for MLMM method (black line) and FarmCPU method (red line).


Power
FDR Mean
Power
Type I Error Mean