A Parallel Distributed Computational Pipeline for maxT Permutation Testing in GWAS

Tinozivashe Sibanda and Andisani Nemavhola
November 14, 2019

Abstract

This report describes the capstone project for COMS7057A - Large Scale Computing Systems and Scientific Programming at the University of the Witwatersrand. The code can be found at the github repository for this project by visiting: https://github.com/tino-sibanda/next-gwas.git

1 Introduction

Genome Wide Association Studies (GWAS) look for variances in DNA sequences that occur more frequently in people with a certain disease (i.e. cases) than in people without the disease (i.e. controls)[1] by comparing the frequency of alleles at each loci, usually taken as single-nucleotide polymorphisms (SNPs) [2]. Essentially, association testing is done per SNP using the χ^2 test for independence under the null hypothesis of no association; p-values are derived from the underlying contingency table. Typically, because millions of SNPs need to be tested - causing inflated Family-Wise Error Rates (FWER) amongst other issues, the obtained p-values are adjusted via Bonferroni correction [3].

Permutation testing methods are an alternative and handle FWER well, despite being computationally expensive. In particular, maxT based permutation corrects the p-values by simulating the null distribution through permuting the phenotype values (*i.e. case or control*) [4]; adjusted p-values are then calculated using the single step max-t procedure [5].

1.1 Problem Statement

The objective of this project is to design a parallel distributed computational pipeline which performs maxT based permutation testing for association studies. From our perspective the pipeline should be portable and analytical results should be reproducible.

1.2 Solution Approach

The solution chosen in this project was to build a computational pipeline using Nextflow to orchestrate statistical computations written using the Plink tool for genome association analysis; Python functions were written to visualise the manhattan plots and qqplots.

Essentially, in this approach, blocks of 100 SNPs each are extracted and processed in parallel across different nodes on the cluster and the results are then merged together. The pipeline is available on Github at https://github.com/tino-sibanda/next-gwas.git.

1.3 Hardware

The Wits Core Cluster (ZA-WITS-CORE) was used to run the parallel distributed pipeline. The cluster runs CentOS 7.5 and provided upto 45 worker nodes, 1000 hyper-threaded cores and, between 24GB-1TB of RAM. This cluster is setup to use slurm as the job scheduling system and does not support containerization using Docker.

2 Results and Discussion

2.1 Statistical Analysis

Figure 1 below shows the Manhattan plot (left) and the qqplot (right) generated from 100 000 permutation tests.

The Manhattan plot shows on the y-axis the negative log-base-10 of the P value for each of the SNPs in the genome (along the x-axis), when tested for differences in frequency between 56 cases and 56 controls across 1 457 897 variants/SNPs. The line shows the threshold for genome-wide significance ($P < 5 \times 10^{-8}$).

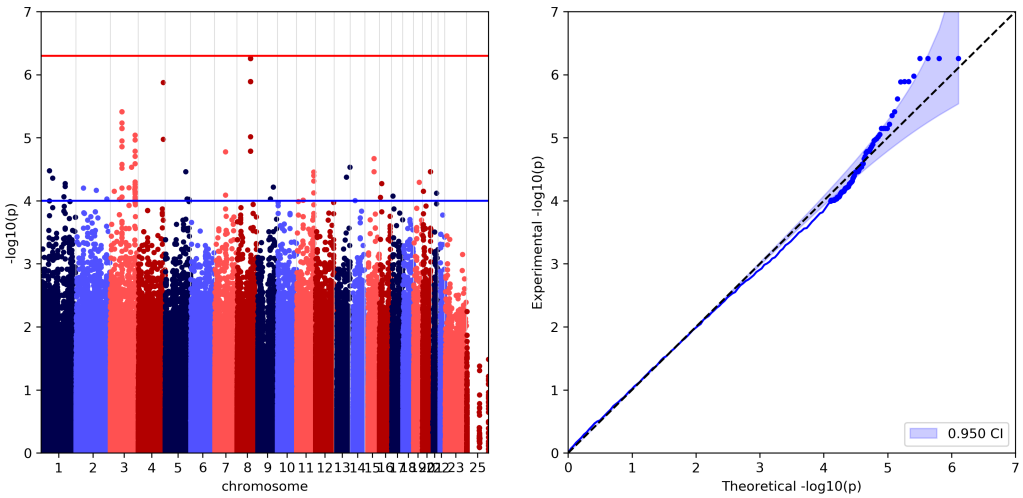

Figure 1: Manhattan plot and quantile-quantile plot for multiple Testing using 100 000 maxT Permutations

The qqplot shows the departure of the upper tail of the distribution from the expected trend along the diagonal is due to the presence of substantially more large test statistic values than would be expected if all null hypotheses were true.The graph suggests that it is unlikely that all the null hypotheses are true. For future study, a list of all SNPs that support the alternate hypothesis should be extracted. Note that no results were available for a million permutations due to memory error.

2.2 Computational Analysis

The performance of the entire pipeline including the maxT permutation was measured by collecting data on the Pipeline Duration in seconds and the CPU hours for different permutations. The configuration for these tests were kept constant at 8GB RAM, 4 CPUs and up-to a maximum of 100 parallel tasks.

Permutation	Duration (s)	CPU Hours
1	96	0.1
10	109	0.1
100	91	0.1
1000	86	0.1
10 000	158	0.6
100 000	151	2.6
1 000 000	No Results	

Table 1: Performance evaluation for fixed process configuration

References

[1] Michelle Chang, Lin He, and Lei Cai. An overview of genome-wide association studies. In *Methods in Molecular Biology*, pages 97–108. Springer New York, 2018.

[2] Thorsten Dickhaus, Klaus Straßburger, Daniel Schunk, Carlos Morcillo-Suarez, Thomas Illig, and Arcadi Navarro. How to analyze many contingency tables simultaneously in genetic association studies. *Statistical Applications in Genetics and Molecular Biology*, 11(4), January 2012.

[3] Arun Sethuraman, Nicolette M. Gonzalez, Christy E. Grenier, Khyati S. Kansagra, Ken K. Mey, Stefany B. Nunez-Zavala, Bryce E. W. Summerhays, and Gwendalyn K. Wulf. Continued misuse of multiple testing correction methods in population genetics-a wake-up call? *Molecular Ecology Resources*, 19(1):23–26, January 2019.

[4] Jelle J. Goeman and Aldo Solari. Multiple hypothesis testing in genomics. *Statistics in Medicine*, 33(11):1946–1978, January 2014.

[5] V. Steiss, T. Letschert, H. Schafer, and R. Pahl. PERMORY-MPI: a program for high-speed parallel permutation testing in genome-wide association studies. *Bioinformatics*, 28(8):1168–1169, February 2012.