BIOINFORMATICS APPLICATIONS NOTE

Vol. 29 no. 1 2013, pages 135–136
doi:10.1093/bioinformatics/bts647

Genetics and population analysis

Advance Access publication November 29, 2012

BlueSNP: R package for highly scalable genome-wide association studies using Hadoop clusters

Hailiang Huang[1,2], Sandeep Tata[3] and Robert J. Prill[1,*]

[1]Healthcare Informatics, IBM Almaden Research Center, San Jose, CA 95120, [2]Analytic and Translational Genetics Unit, Massachusetts General Hospital, Boston, MA 02114 and [3]Information Management, IBM Almaden Research Center, San Jose, CA 95120, USA

Associate Editor: Jeffrey Barrett

ABSTRACT

Summary: Computational workloads for genome-wide association studies (GWAS) are growing in scale and complexity outpacing the capabilities of single-threaded software designed for personal computers. The BlueSNP R package implements GWAS statistical tests in the R programming language and executes the calculations across computer clusters configured with Apache Hadoop, a *de facto* standard framework for distributed data processing using the MapReduce formalism. BlueSNP makes computationally intensive analyses, such as estimating empirical *p*-values via data permutation, and searching for expression quantitative trait loci over thousands of genes, feasible for large genotype–phenotype datasets.

Availability and implementation: http://github.com/ibm-bioinformatics/bluesnp

Contact: rjprill@us.ibm.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on July 18, 2012; revised on October 18, 2012; accepted on October 26, 2012

1 INTRODUCTION

Depending on the study design and statistical methodology, GWAS statistics can be handily computed on a personal computer, or require substantially greater computational resources. Usually, study designs and statistical methodologies that entail massive computation are avoided for practical reasons. Yet, there are times when intensive computation can not be avoided. An example of a computationally demanding task is estimating empirical *p*-values via data permutation when a test statistic does not follow a standard distribution. Up to 10^8 permutations of the data can be required to assess genome-wide significance, a massive computational burden. Another example is in the analysis of expression quantitative trait loci (eQTL) study designs with tens of thousands of gene expression phenotypes (Schadt *et al*., 2003).

The superb genetics software, PLINK (Purcell *et al*., 2007), is an open-source whole genome association analysis program written in C++ that is extremely fast and efficient at computing commonplace association statistics (e.g. allelic test, linear regression, etc.). However, it can be a logistical challenge to run PLINK on computer clusters, manually partitioning large jobs into sub-parts and stitching-together potentially thousands of output files.

MapReduce is a parallel programming methodology for splitting a large problem into sub-parts (the map step), computing partial solutions on sub-parts independently, then assembling the partial solutions into the overall solution (the reduce step). Fortunately, a GWAS analysis is usually decomposable into small independent sub-parts—for example, single-SNP association tests assume independence of SNPs—making it straightforward to apply MapReduce. Apache Hadoop (http://hadoop.apache.org), an open-source MapReduce implementation, solves three technical problems confronted by developers of parallel algorithms: it handles the distributed storage of large data, it handles the scheduling of jobs and re-scheduling of failed jobs, and the limited expressivity of the MapReduce formalism simplifies parallel program design. Hadoop is a *de facto* standard framework for big-data analytics and is gaining popularity in bioinformatics (Langmead *et al*., 2009; Schadt *et al*., 2010). Hadoop instances can be provisioned on-demand from 'cloud computing' service providers.

We introduce the BlueSNP R package which distributes GWAS computation over a cluster configured with the Hadoop framework, making computationally intensive analyses feasible for large genotype–phenotype datasets. Genetics researchers can utilize the association tests provided with BlueSNP or supply a novel association test as a user-defined R function. BlueSNP removes the complexity of interacting with the cluster, freeing the researcher to focus on advanced analytics using the R programming environment (R Core Team, 2011).

1.1 BlueSNP architecture and core functionality

The BlueSNP R package sits atop a software stack that hides the details of interacting with a multiplicity of multi-core processors from the user (Fig. 1). Proceeding from the bottom up, Hadoop

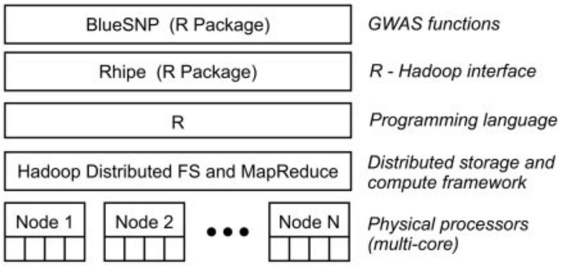

Fig. 1. BlueSNP builds upon packages and frameworks for parallel programming and big-data analytics

*To whom correspondence should be addressed.

Table 1. Minutes to test 1M SNPs in 10K individuals

Phenotypes	10 Nodes	20 Nodes	40 Nodes
1	3.6	2.7	1.6
10	14.0	8.2	4.7
100	128.2	65.2	33.7

5 Map cores, 2 reduce cores, 1 unused core per node.

handles the distributed data storage and the distributed task management. The RHIPE R package (http://www.datadr.org) provides a facility for authoring and running MapReduce programs from within the R environment (Guha, 2010). BlueSNP builds upon RHIPE, providing high-level GWAS functions suitable for genetics researchers who might not have an interest in the low-level details of parallel programming.

Example 1. Analysing many phenotypes

Some study designs, for example, expression QTL studies which treat gene expression values as quantitative phenotypes, can involve analysing thousands of phenotypes. Manually distributing tens to thousands of separate GWAS jobs over a computer cluster and collating the separate outputs would be time-consuming and error-prone. Using BlueSNP, scaling from one phenotype to thousands of phenotypes is automatic, as is the generation of collated reports of *p*-values, test statistics, etc.

The R syntax for analysing one phenotype or thousands of phenotypes is identical, the crucial difference being the dimensions of the phenotype data structure. Analysing many phenotypes is a simple matter of supplying a matrix of phenotype values, one phenotype per column (one sample per row), to the `gwas` function. The `gwas` function is called with parameters specifying the locations of the input and output files/directories on the Hadoop distributed file system (HDFS):

```
gwas(genotype.hdfs.path = "/snps",
    phenotype.hdfs.path = "/phenotypes.
    RData",
    output.hdfs.path = "/output",
    method = "qt.linear.regression")

P <- gwas.results("/output", type="p.value")
```

In this example, linear regression is performed for all SNP-phenotype pairs and a table of *p*-values is retrieved. Collating voluminous data (e.g. the outputs of many statistical tests) into actionable reports is a particular strength of the Hadoop framework.

On our 40-node cluster it took 34 min to analyse 100 phenotypes using a linear regression association test on 10^6 simulated SNPs in 10^4 individuals (Table 1). In most of our performance assessments, the time to analyse a given number of phenotypes was halved by doubling the number of compute nodes, indicating near-optimal scaling. (See Supplementary Material for a comprehensive performance assessment and machine specifications.)

Example 2. Empirical *p*-values by data permutation

When test statistic does not follow a standard distribution, or if small sample size renders a standard distribution unreliable, Monte Carlo methods can be used to estimate empirical *p*-values. Monte Carlo methods are conceptually simple but computationally expensive. The null distribution of any test statistic can be estimated by observing the test statistic for many instances of random data that preserve some essential aspects of the real data. In practice, the BlueSNP functions for estimating empirical *p*-values shuffle the phenotype vector to randomize the bivariate distribution of genotype and phenotype while preserving the univariate distributions.

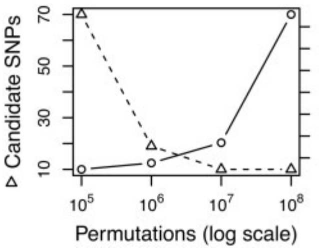

Fig. 2. In adaptive permutation, the pool of candidate SNPs decreases as *p*-value estimates become more precise. Running time increases with the number of permutations but is substantially faster than in exhaustive permutation

BlueSNP computes both exhaustive and adaptive permutation-based *p*-values. Adaptive permutation drops non-significant SNPs from subsequent rounds of permutation, vastly decreasing the number of calculations and therefore the running time compared with exhaustive permutation. The concept of adaptive permutation is well established; our MapReduce implementation of adaptive permutation is novel. Calculations are periodically rebalanced over the cluster nodes to maximize processor utilization.

On our 40-node cluster, it took 7.6 h to estimate empirical *p*-values to precision $\leq 10^{-8}$ using linear regression on the dataset described in Table 1. With adaptive permutation, a 10-fold increase in the number of permutations takes much less than a 10-fold increase in time (Fig. 2). This efficient permutation framework opens the door to novel user-defined association tests that do not rely on the assumed validity of standard distributions. Any user-defined function of genotype and phenotype can be a test statistic.

Funding: H.H., S.T. and R.J.P. funded by IBM Research.

Conflict of Interest: none declared.

REFERENCES

Guha,S. (2010) Computing environment for the statistical analysis of large and complex data. PhD Thesis, Department of Statistics, Purdue University, West Lafayette, IN, USA.

Langmead,B. *et al.* (2009) Searching for SNPs with cloud computing. *Genome Biol.*, **10**, R134.

Purcell,S. *et al.* (2007) Plink: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.*, **81**, 559–575.

R Core Team. (2011) *R: a language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria.

Schadt,E.E. *et al.* (2003) Genetics of gene expression surveyed in maize, mouse and man. *Nature*, **422**, 297–302.

Schadt,E.E. *et al.* (2010) Computational solutions to large-scale data management and analysis. *Nat. Rev. Genet.*, **11**, 647–57.