Sains Malaysiana 46(9)(2017): 1449–1455
http://dx.doi.org/10.17576/jsm-2017-4609-13

Parallelization of Logic Regression Analysis on SNP-SNP Interactions of a Crohn's Disease Dataset Model

(Analisis Regresi Logik Keselarian pada Interaksi SNP-SNP Model Dataset Penyakit Crohn)

UNITSA SANGKET*, SURAKAMETH MAHASIRIMONGKOL, PICHAYA TANDAYYA, SURASAK SANGKHATHAT, WASUN CHANTRATITA, QI LIU & YUTAKA YASUI

ABSTRACT

SNP-SNP interactions have been recognized to be basically important for understanding genetic causes of complex disease traits. Logic regression is an effective methods for identifying SNP-SNP interactions associated with risk of complex disease. However, identifying SNP-SNP interactions are computationally challenging and may take hours, weeks and months to complete. Although parallel computing is a powerful method to accelerate computing time, it is arduous for users to apply this method to logic regression analyses of SNP-SNP interactions because it requires advanced programming skills to correctly partition and distribute data, control and monitor tasks across multi-core CPUs or several computers, and merge output files. In this paper, we present a novel R-library called SNPInt to automatically speed up analyses of SNP-SNP interactions of genome-wide association (GWA) studies using parallel computing without the advanced programming skills. The Crohn's disease GWA studies dataset from the Wellcome Trust Case Control Consortium (WTCCC) that includes 4,680 individuals with 500,000 SNPs' genotypes was analyzed using logic regression on a computer cluster to evaluate SNPInt performance. The results from SNPInt with any number of CPUs are the same as the results from non-parallel approach, and SNPInt library quite accelerated the logic regression analysis. For instance, with two hundred genes and twenty permutation rounds, the computing time was continuously decreased from 7.3 days to only 0.9 day when SNPInt applied eight CPUs. Executing analyses of SNP-SNP interactions using the SNPInt library is an effective way to boost performance, and simplify the parallelization of analyses of SNP-SNP interactions.

Keywords: Crohn's disease GWA studies; logic regression; parallel computing; R; SNP-SNP interactions

ABSTRAK

Interaksi SNP-SNP telah diiktiraf penting pada dasarnya untuk memahami punca genetik sifat penyakit kompleks. Regresi logik adalah satu kaedah yang berkesan untuk mengenal pasti interaksi SNP-SNP yang dikaitkan dengan risiko penyakit kompleks. Walau bagaimanapun, mengenal pasti interaksi SNP-SNP adalah mencabar secara pengkomputeran dan mungkin mengambil masa berjam, berminggu dan berbulan untuk diselesaikan. Walaupun pengkomputeran selari adalah satu kaedah berkuasa untuk mempercepatkan masa pengiraan, ia adalah sukar bagi pengguna untuk menggunakan kaedah ini dalam analisis regresi logik interaksi SNP-SNP kerana ia memerlukan kemahiran pengaturcaraan lanjutan untuk pemetakan dan pengagihan data dengan betul, mengawal dan memantau tugas pelbagai teras CPU atau beberapa komputer dan menggabungkan fail output. Dalam kertas ini, kami memberikan R-perpustakaan novel yang disebut SNPInt untuk secara automatik mempercepatkan analisis interaksi SNP-SNP kajian sekutuan genom-menyeluruh (GWA) menggunakan pengkomputeran selari tanpa kemahiran pengaturcaraan lanjutan. Kajian dataset penyakit Crohn GWA daripada Wellcome Trust Case Control Consortium (WTCCC) yang merangkumi 4,680 individu dengan 500,000 SNP genotip telah dianalisis menggunakan regresi logik pada kelompok komputer untuk menilai prestasi SNPInt. Hasil daripada SNPInt dengan apa-apa bilangan CPU adalah sama seperti hasil daripada pendekatan bebas-selari dan perpustakaan SNPInt mempercepatkan analisis regresi logik. Sebagai contoh, dengan dua ratus gen dan dua puluh pusingan permutasi, masa pengiraan berterusan menurun daripada 7.3 hari kepada 0.9 hari sahaja apabila SNPInt menggunakan lapan CPU. Analisis pelaksanaan interaksi SNP-SNP menggunakan perpustakaan SNPInt adalah merupakan satu cara yang berkesan untuk meningkatkan prestasi dan memudahkan keselarian analisis interaksi SNP-SNP.

Kata kunci: Interaksi SNP-SNP; kajian penyakit Crohn GWA; pengiraan selari; R; regresi logik

INTRODUCTION

Single nucleotide polymorphisms (SNPs) refer to genetic variations at the single nucleotide level. There are more than one million SNPs in the human genome. From a large set of SNP measurements, finding SNPs whose variations are associated with a disorder is an important analytic goal of bioinformatics. Such analyses can help researchers discover genes that predispose individuals to higher risk of the disorder. In addition, SNP analyses may assist researchers to explain possible heterogeneity in

individuals' responses to a certain medicine (http://ghr.nlm.nih.gov/handbook/genomicresearch/snp). Schwender and Ickstadt (2008) suggested that it is usually not an individual SNP that plays an imperative role in the risk of a complex disorder. Rather, it is SNP-SNP interactions (such as SNP_1 and SNP_2) that influence strongly the risk of a complex disorder. This suggests that SNP-SNP interactions may identify high risk groups (Garte 2001), to whom an intervention strategy for decreasing the risk or detecting the disorder early for treatment may be considered. Logic regression, developed by Ruczinski et al. (2003) is a flexible method of regression with Boolean combinations of binary covariates as explanatory variables. The strength of logic regression is its capacity for finding complex interactions between predictors. It has certain advantages over other analyses such as Classification and Regression Trees (CART) (Breiman 1984) and random forests (Breiman 2001), which relate only the main effects and simple (two to three-way at most) interactions between predictors. In addition, analyses of BOOST (Wan et al. 2010) can detect interaction between only two SNPs. Logic regression can be employed in many various purposes, especially to search for multi-way SNP interactions, e.g. 4-way interactions: such an analysis is often difficult with other methods including random forests, CART and Support Vector Machines (SVMs) (Guyon et al. 2002; Schwender & Ickstadt 2008). For these reasons, logic regression is a powerful methods for identifying SNP-SNP interactions associated with risk of complex disorders. R is a well-known open source statistics programming language and environment (Ihaka & Gentleman 1996). With many useful libraries such as GenABEL (Aulchenko et al. 2007) and ParallABEL (Sangket et al. 2010) for Genome-Wide Association Analyses. LogicReg is an effective R library for logic regression analyses implemented in R by Kooperberg and Ruczinski (http://cran.rproject.org/web/packages/LogicReg/index.html). LogicReg can be used to successfully find SNP-SNP interactions in Crohn's Disease (CD) GWAS data of the Welcome Trust Case Control Consortium (WTCCC 2007) and the two smaller GWASs from the Database of Genotype and Phenotype (dbGaP) (Dinu et al. 2012). Strong evidence of CD-association for 195 genes has been found including novel susceptibility genes such as *ISX*, *SLCO6A1* and *TMEM183A* (Dinu et al. 2012). However, identifying SNP-SNP interactions by LogicReg are computationally challenging and may take hours, weeks or months to complete depending on how large the number of permutations and datasets are. For instance, for the gene-level SNP analysis of the Crohn's disease dataset which include approximately 13,500 genes, more than 400,000 runs of logic regressions were needed when SNP-SNP interactions within each gene must be analyzed with thirty permutations. Moreover, the size of the dataset is also large: For example, the WTCCC Crohn's disease dataset includes 4,680 individuals with 2,000 SNPs. Accordingly, without parallel computing, the logic regression analysis requires massive computing time, hours to months, depending on the size of the dataset being analyzed and computer performance. To speed up analyses of SNP-SNP interactions or to allow a large number of permutation rounds for a large dataset such as the ones from genome-wide association (GWA) studies, users have to apply parallel computing to the analysis processes. Nevertheless, it is very difficult and complicated for users to apply parallel computing to logic regression analyses of SNP-SNP interactions because they need advanced programming skills to correctly partition an distribute data, control and monitor tasks across the computers and merge outputs. For example, the analyses will be failed, if the users mistakenly partition the large data. Another example, the outputs from the computers are usually messy and their order is hard to track. Accordingly, it is necessary to create a novel R-library (parallel version of LogicReg) that allows parallel computation of logic regression. With the novel R-library, the users can execute the novel R-library to automatically parallelize logic regression analyses studies without the advanced programming skills. Moreover, the statistical outputs from the novel R-library with any number of CPUs are the same as the statistical outputs from non-parallel approach.

In this paper, we propose a development of SNPInt, a novel R library to parallelize SNP-SNP interaction analyses of GWA studies using logic regression and parallel computing. SNPInt aims not only to accelerate the computation of SNP-SNP interaction analyses, but also simplify analysis parallelization. Moreover, with SNPInt, users do not need to be proficient with parallel programming because it will automatically partition and distribute data, control and monitor tasks across the computers and merge output files.

Materials An Methods

Logic regression analyses of GWA studies

Logic regression aims to find Boolean combinations of the predictors. We consider that all predictors $\{X_i, i = 1,\ldots,p\}$ are binary (0 or 1, yes or no), for identification of SNP associations. Specifically, the predictor $X_i = 1$ if the i^{th} SNP has a certain genotype and $X_i = 0$ otherwise. Each Boolean combination of SNPs could use three operators, $\wedge$ (AND), $\vee$ (OR) and c (NOT) to form a logic expression, $L_j, j = 1,\ldots,t$ such as:

$$L_j = (X_1 \wedge X_2) \vee X_3^c.$$

This example of Boolean logic expression means:

$$L_j = (SNP_1 \wedge SNP_2) \vee SNP_3^c.$$

Logic regression uses L's instead of X's in its linear predictor and takes the form:

$$f(E[Y]) = \beta_0 + \sum_{j=1}^{t} \beta_j L_j,$$

where Y is a response variable; f is a link function; and parameters $\beta_j, j = 0,\ldots, t$ are concurrently estimated with the search for the Boolean expressions L_j's in the above equation that minimize the scoring function related with this model type (Ruczinski et al. 2003). The SNP-SNP interactions can be used as biomarkers which refer to the risk of the disease. For example, if the p-value of SNP_1 and SNP_2 and SNP_3 is less than threshold value, people who have these SNPs will have the high risk. Moreover, the gene containing these SNPs and protein associated with disease can be found. Drug discovery and development would be possible using this information.

SOFTWARE DESIGN

The computing time of logic regression is demanding as it explores a large space for an optimal set of logics and needs a large number of permutation tests to assess signals in the data. Hence, parallel computing is very important as it decreases computing time. To parallelize a logic regression analysis, SNPInt running on the frontend-node partitions the input dataset into G subsets, where G is the number of genes to be analyzed. Each subset is included the SNPs of each gene. For instance, the first subset contained SNPs of the first gene. The advantage of this partitioning method is that the outputs of each subset will be the same as the outputs from non-parallel approach because the subsets of this method do not impact the algorithm of logic regression analyses. Since each gene has different number of SNPs, the 'task full' approach (Browning & Browning 2008) is adapted to keep load balancing when SNPInt is executing. Also, this approach is not sensitive to the number of CPUs or compute-nodes. With this approach, the frontend-node sends these subsets to idle CPUs on compute-nodes. The examples of data partitioning and distribution are shown in Figure 1. If there are three compute-nodes and each compute-node has only one CPU, SNPs of $G_1 - G_3$ ($Gene_1$ - $Gene_3$) will be executed on these compute-nodes. When the execution of the second compute-node has finished, the frontend-node will send the SNPs of next gene (G_4) to it as shown in Figure 1(a) - a cycle that proceeds until all the genes are sent. After all the compute-nodes has finished, the frontend-node will combine all the outputs as shown in Figure 1(b). Because the SNPInt has been designed based on the 'task full' approach, the users should optimize the number of CPUs suitable for computational throughput by setting the number of CPUs as many as they can but not over the number of genes. If the number of CPUs is more than the number of genes, some CPUs will be idle and wasteful.

SOFTWARE IMPLEMENTATION

SNPInt (Project home page: http://www.mbb.psu.ac.th/SNPInt/index.html) performs SNPs data in LogicReg input format, and an example of the useful input can be seen in Figure 2 (https://www.ualberta.ca/~yyasui/snpGWAS/). The sequential workflow for a logic regression analysis on a single CPU/computer is shown in Figure 3(a). The data were analyzed by the LogicReg library, working under the R program. LogicReg sequentially analyzes the raw data and produces statistical data (e.g. p-values) as outputs. Since this sequential workflow generally takes massive computing time to conduct statistical analyses, we have developed a novel parallel workflow for SNPInt to save computing time. The novel parallel workflow in a logic regression analysis is shown in Figure 3(b). A job scheduler such as the SUN Grid Engine (http://www.rocksclusters.org/roll-documentation/sge/5.4/using-sge.html) distributes the data to each compute-node on a cluster to queue jobs and reserve a set of CPUs required

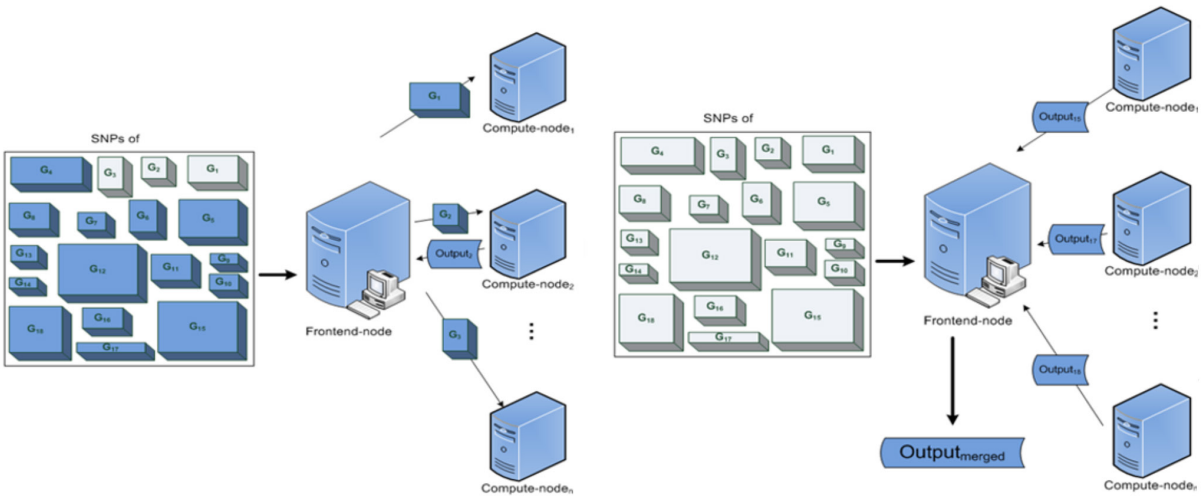

FIGURE 1. The example of data partitioning and distribution, (a) The data is partitioned into eighteen subsets, and each subset is contained SNPs of a gene. G_1 - G_3 subsets are executed on the compute-nodes and (b) The frontend-node will combine all the outputs after all the compute-nodes has finished

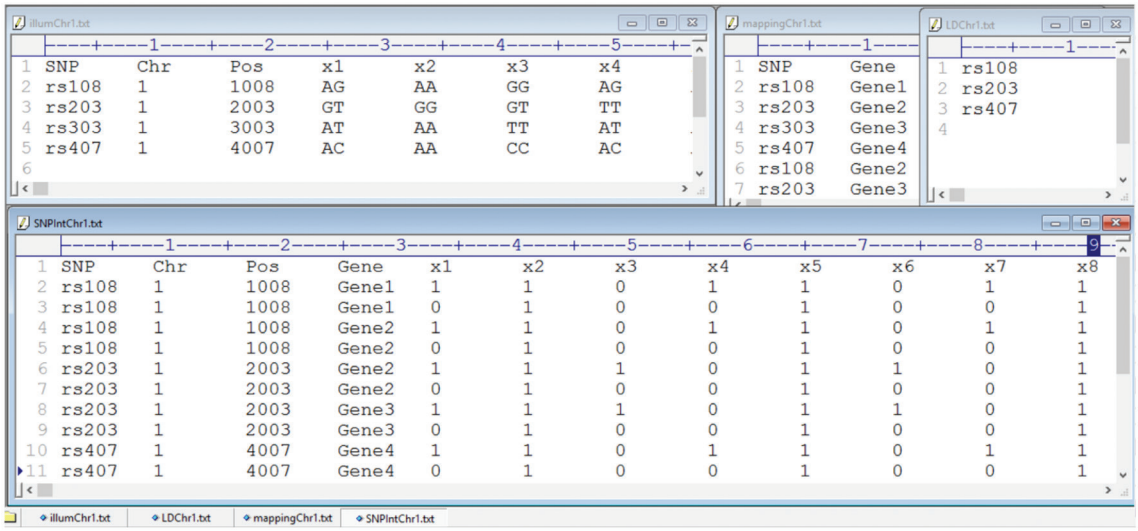

FIGURE 2. The data preparation for SNPInt. The example of SNPs data that can be executed by SNPInt is shown in "SNPIntChr1.txt" file. The valid input data of SNPInt contains SNP id, chromosome number, position of the SNP on chromosome and the SNPs codes of controls and cases (x1, x2, x3, x4, x5, x6, x7 and x8). The SNPs data file can be created from the three files including (1) the SNPs data file in Illumina format ("illumChr1.txt"), (2) the list of genes of each SNP file ("mappingChr1.txt"), and the list of SNPs that passed LD process ("LDChr1.txt") using R script, for the SNPs codes in the output file of the script, if the genotype is AA AT TT, then the two ways of coding correspond to two indicators. One indicator for AA, and the other for only one "A". For instance, if a SNP has AA AT TT, two variables for this SNP are created. Variable1 is 1 when AA, 0 otherwise; Variable2 is 1 when AA or AT, otherwise 0

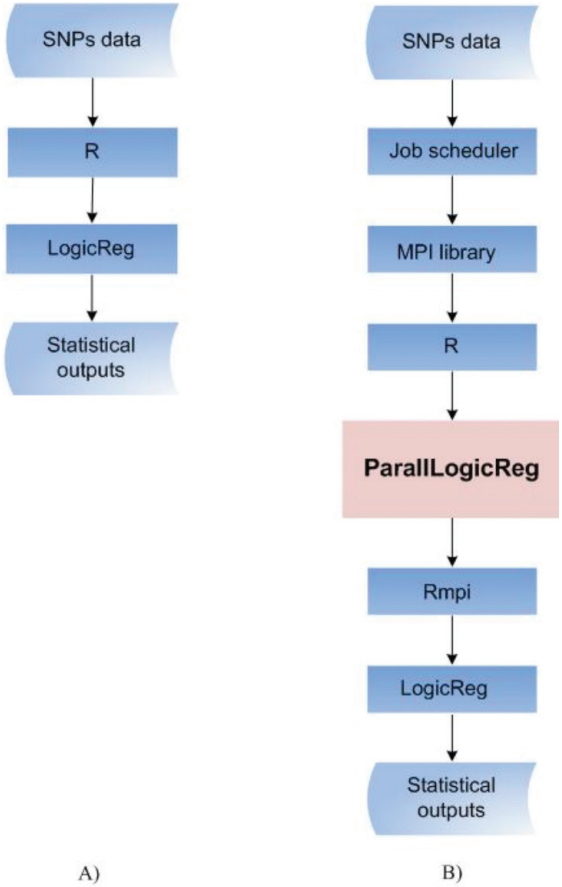

FIGURE 3. Logic regression computing workflow, (a) Sequential logic regression computing workflow runs on a single CPU or a computer and (b) Parallel logic regression computing workflow runs on a multiple CPUs or a set of computers

by a MPI (Message Passing Interface) library such as LAM/MPI (Local Area Multicomputer/Message Passing Interface, http://www.lam-mpi.org/) and Open MPI (http://www.open-mpi.org/). The MPI library has various functions called by Rmpi (an R library) to communicate the computers in a cluster using R language (http://math.acadiau.ca/ACMMaC/Rmpi/structure.html). Rmpi library is applied for SNPInt to connect the computers in a cluster analyzed logic regression using LogicReg. Also, SNPInt can be run on any Operating System supporting the components for the parallel workflow such as Linux and Solaris. In addition, SNPInt partitions a job into several smaller tasks on a frontend-node using basic R commands, and each task is contained SNPs of one gene. After that, SNPInt distributes tasks to the reserved CPUs. These CPUs are executed the tasks on compute-nodes and call the LogicReg; then, the outputs are returned to the frontend-node and combined by SNPInt. The statistical data from the parallel workflow can be approved by comparison with the statistical data from the sequential workflow. Users can use SNPInt to parallelize logic regression function easily. An executing command that parallelizes logic regression on multiple CPUs is shown in Figure 4. To run the function, the number of CPUs can be specified in sun grid engine (http://www.rocksclusters.org/roll-documentation/sge/5.4/using-sge.html). SNPInt can be run on not only a single CPU, but also on multi-core CPUs in both a single computer and a computer cluster. The Hanuman cluster has been used to evaluate the performance of SNPInt. Hanuman cluster includes five IBM servers XSeries 3362, which are comprised by a frontend-node and four compute-nodes, with two SINGLE-CORE Intel Xeon (2.8 GHz) CPUs and four GB RAM, respectively. The frontend-

```
library(SNPInt)

resp=c(rep(0,2935),rep(1,1745)  # number of controls = 2,935; number of cases = 1,745

nperm=20                # number of permutations
niter=20                # number of iterations
begin=1                 # the first gene id that will be run
end=10                  # the last gene id that will be run
in=("input.txt")        # data file that will be run

output = SNPInt (infile=in,resp=resp,begin=begin,end=end,nperm=nperm,niter=niter)
```

FIGURE 4. An example of SNPInt usage. The example of parallel execution was used to analyze Crohn's disease data when gene ids were between one and ten, and number of permutations and iterations were twenty. The data contained 1745 cases and 2935 controls. Besides, Users could set the number of CPUs in a job scheduler such as the SUN Grid Engine

node of the cluster can be connected via the Internet and can control the compute-nodes of the cluster through an Ethernet switch. Also, this cluster provides Rocks Cluster Distribution version 4.3 (http://www.rocksclusters.org/wordpress/) including the SUN Grid Engine version 4.3, LAM/MPI version 7.1.2, R program version 2.8.1, Rmpi library version 0.5-6, LogicReg version 1.4.9 and SNPInt 1.0.

RESULTS

Crohn's disease data set from WTCCC, a chronic inflammatory disease data set of the intestines (Parkes et al. 2007; WTCCC 2007) which contains 1,745 controls and 2,935 cases with approximately 500K SNPs, was used to measure the performance of SNPInt. Figure 5 shows trace results of logic regression analyses of GWA studies using SNPInt for the Crohn's disease data on Hanuman. SNPInt was executed twenty permutations and iterations for each gene in three subsets of the data in chromosome 1, which are ten, one hundred and two hundred genes. There is no concern about the statistical quality from SNPInt because there is no difference between the statistical outputs from SNPInt and original LogicReg (non-parallel approach). SNPInt saved computing time for ten, one hundred and two hundred analyses, especially with eight CPUs. For example, on a single CPU, the two hundreds gene analyses on the first chromosome took 7.3 days using non-parallel approach, but only 0.9 day with eight CPUs using SNPInt.

If the number of available CPUs is P, the computing time for P CPUs is $time_P$ and the sequential computing time for a CPU is $time_1$, thus, the speedup for P CPUs is:

$$speedup_P = time_1 / time_P.$$

The speedups of analyzing Crohn's disease data using SNPInt function applying the above equation are shown in Figure 6. It shows that the saved time by SNPInt is linearly correlated to the number of CPUs. For instance, the executing speed of the two hundreds gene analyses on eight CPUs was approximately eight times faster than that on one CPU. This suggested that executing with more CPUs,

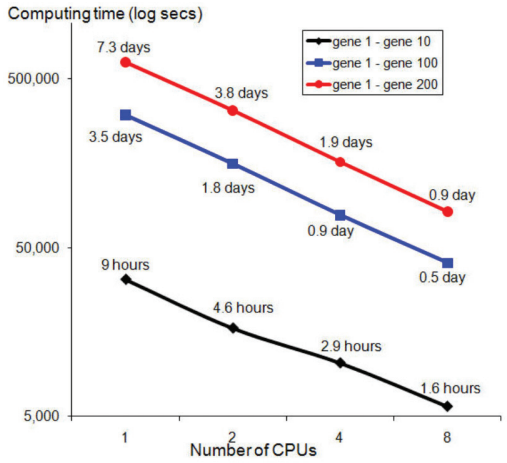

FIGURE 5. Trace results of SNPInt for Crohn's disease data on Hanuman cluster. Ten, one hundred and two hundred genes of Crohn's disease data were executed with SNPInt. These genes were run on one CPU using non-parallel approach, and two, four and eight CPUs using SNPInt. Non-parallel approach and SNPInt were executed twenty permutations and iterations for each gene

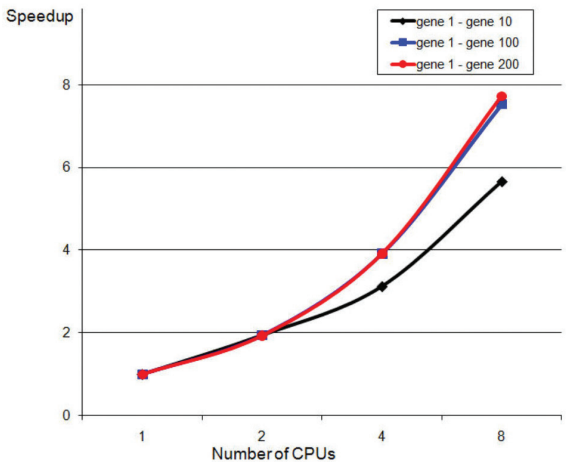

FIGURE 6. The speedups of SNPInt for Crohn's disease data. The speedups were extrapolated from Figure 4 applying the speedup equation. The speedups showed that the more CPUs, the more speedup were increased by SNPInt

more computing time will be saved by SNPInt. With eight CPUs, SNPInt can save 6.4 days for two hundred analyses, whereas it can reduce 3 days for one hundred analyses and 7.4 h for ten analyses. This also implies that the larger the volume/size data, the more computing time will be saved by SNPInt.

Discussion

According to speedup equation, the overhead for *P* CPUs is

$$overhead_P = time_p\text{-}(time_1 / P).$$

Since SNPInt is not sensitive to the number of CPUs, it can be run on a larger cluster. The overhead of analyses with various numbers of CPUs on a large computer cluster can be predicted based on the overhead of eight CPUs as shown in Figure 6. The computing time and speedups on a large cluster for ten, one hundred and two hundred analyses extrapolated from Figure 5 applying the above overhead equation are shown in Figure 7, respectively. The time-saving rates are grown when the numbers of CPUs are increased until the numbers of CPUs are greater than number of genes. Thus, with larger datasets, the time-saving rates will be larger in a large computer cluster. Nonetheless, the user should optimize the number of CPUs suitable for computational throughput.

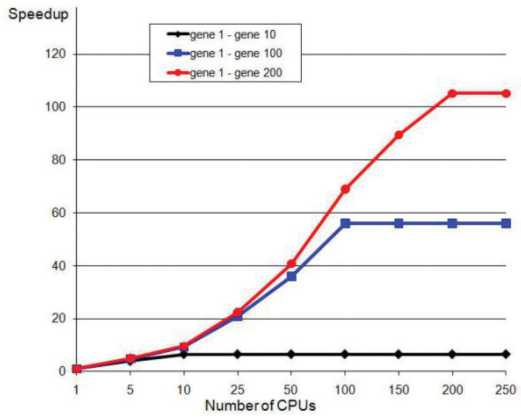

FIGURE 7. The speedups on a large computer cluster. The speedups on a large cluster for ten, one hundred and two hundred genes analyses were extrapolated from Figure 5 applying the overhead equation and speedup equation

The results showed that SNPInt using parallel computing can save computing time for analyzing massive data. The statistical outputs from SNPInt with number of CPUs is the same as the statistical outputs from non-parallel method because SNPInt partitions data into small subsets which have no effect for logic regression analyses. Users can set the number of CPUs in SNPInt to execute data, which will reduce computing time that growingly correlated to the number of CPUs. Also, if users apply more CPUs, more computing time will be saved by SNPInt. However, the number of CPUs they assigned should be less than, or equal to, the number of genes to avoid idle CPUs. Due to a benefit of MPI, SNPInt can be run not only on distributed memory architecture like an architecture on Hanuman but also on a shared memory architecture. Nevertheless, distributed memory architecture gives more overhead than shared memory architecture.

Conclusion

We have developed a novel R-library called SNPInt to speed up analyses SNP-SNP interactions using parallel computing components, which consist of a job scheduler, a MPI library, an Rmpi library and a LogicReg library. SNPInt has been designed to be a user-friendly library. Identification of SNPs associated with Crohn's disease is used to measure the performance of the SNPInt function. The SNPInt library is an effective library used to accelerate computing time of logic regression on the computer cluster in application for SNP analyses of GWA studies and other data analyses.

Acknowledgements

This research was supported by a grant from Prince of Songkla University, contract no SCI550388S and funded by Thailand Center of Excellence for Life Sciences (TCELS). We are grateful to Prof. Dr. Amornrat Phongdara and Assoc. Prof. Dr. Wilaiwan Chotigeat for supporting the PSU research group in Bioinformatics, and the Thai National Grid Center and PSUGrid Center at Prince of Songkla University for providing the Hanuman cluster used for this project. We would like to thank the National e-Science infrastructure Consortium, Thailand for supporting computing infrastructure to test the program. We also thank Conrado Franco-Villalobos for data preparation scripts.

References

Aulchenko, Y.S., Ripke, S., Isaacs, A. & van Duijn, C.M. 2007. GenABEL: An R library for genome-wide association analysis. *Bioinformatics* 23(10): 1294-1296.

Breiman, L. 2001. Random forests. *Machine Learning* 45(1): 5-32.

Breiman, L. 1984. *Classification and Regression Trees*, *The Wadsworth Statistics/Probability Series*. Belmont, Calif.: Wadsworth International Group.

Browning, B.L. & Browning, S.R. 2008. Haplotypic analysis of wellcome trust case control consortium data. *Hum. Genet.* 123(3): 273-280.

Dinu, I., Mahasirimongkol, S., Liu, Q., Yanai, H., Sharaf Eldin, N., Kreiter, E., Wu, X., Jabbari, S., Tokunaga, K. & Yasui, Y. 2012. SNP-SNP interactions discovered by logic regression explain Crohn's disease genetics. *PLoS One* 7(10): e43035.

Garte, S. 2001. Metabolic susceptibility genes as cancer risk factors: Time for a reassessment? *Cancer Epidemiol Biomarkers Prev.* 10(12): 1233-1237.

Guyon, I., Weston, J., Branhill, S. & Vapnik, V. 2002. Gene selection for cancer classification using support vector machines. *Machine Learning* 46: 389-422.

Ihaka, R. & Gentleman, R. 1996. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5(3): 299-314.

Parkes, M., Barrett, J.C., Prescott, N.J., Tremelling, M., Anderson, C.A., Fisher, S.A., Roberts, R.G., Nimmo, E.R., Cummings, F.R., Soars, D., Drummond, H., Lees, C.W., Khawaja, S.A., Bagnall, R., Burke, D.A., Todhunter, C.E., Ahmad, T., Onnie, C.M., McArdle, W., Strachan, D., Bethel, G., Bryan, C., Lewis, C.M., Deloukas, P., Forbes, A., Sanderson, J., Jewell, D.P., Satsangi, J., Mansfield, J.C., Cardon, L. & Mathew, C.G. 2007. Sequence variants in the autophagy gene IRGM and multiple other replicating loci contribute to Crohn's disease susceptibility. *Nat. Genet.* 39(7): 830-832.

Ruczinski, I., Kooperberg, C. & LeBlanc, M. 2003. Logic regression. *Journal of Computational and Graphical Statistics* 12(3): 475-511.

Sangket, U., Mahasirimongkol, S., Chantratita, W., Tandayya, P., & Aulchenko, Y.S. 2010. ParallABEL: an R library for generalized parallelization of genome-wide association studies. *BMC Bioinformatics* 11: 217.

Schwender, H. & Ickstadt, K. 2008. Identification of SNP interactions using logic regression. *Biostatistics* 9(1): 187-198.

Wan, X., Yang, C., Yang, Q., Xue, H., Fan, X., Tang, N.L. & Yu, W. 2010. BOOST: A fast approach to detecting gene-gene interactions in genome-wide case-control studies. *Am. J. Hum. Genet.* 87(3): 325-340.

WTCCC. 2007. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature* 447(7145): 661-678.

Unitsa Sangket*
Department of Molecular Biotechnology and Bioinformatics
Center for Genomics and Bioinformatics Research
Faculty of Science, Prince of Songkla University
Songkhla, 90112
Thailand

Surakameth Mahasirimongkol
Medical Genetic Section, National Institute of Health
Department of Medical Sciences
Ministry of Public Health, Nonthaburi, 11000
Thailand

Pichaya Tandayya
Department of Computer Engineering
Faculty of Engineering
Prince of Songkla University
Songkhla, 90112
Thailand

Surasak Sangkhathat
Tumor Biology Research Unit, Department of Surgery
Faculty of Medicine, Prince of Songkla University
Songkhla, 90112
Thailand

Wasun Chantratita
Department of Pathology, Faculty of Medicine
Ramathibodhi Hospital
Mahidol University, Bangkok, 10400
Thailand

Qi Liu & Yutaka Yasui
Department of Public Health Sciences
School of Public Health
University of Alberta, Edmonton, Alberta
Canada

*Corresponding author; email: unitsa.s@psu.ac.th

Received: 31 August 2016
Accepted: 17 January 2017