BIOINFORMATICS APPLICATIONS NOTE Vol. 28 no. 2 2012, pages 294–295
doi:10.1093/bioinformatics/btr630

Systems biology Advance Access publication November 13, 2011

Gene set analysis in the cloud

Lu Zhang[1], Shengchang Gu[2], Yuan Liu[2], Bingqiang Wang[2] and Francisco Azuaje[1,*]
[1]Laboratory of Cardiovascular research, CRP-Santé, Luxembourg L-1150, Luxembourg and [2]Bioinformatics Center, BGI, Shenzhen 518083, China

Associate Editor: Trey Ideker

ABSTRACT

Summary: Cloud computing offers low cost and highly flexible opportunities in bioinformatics. Its potential has already been demonstrated in high-throughput sequence data analysis. Pathway-based or gene set analysis of expression data has received relatively less attention. We developed a gene set analysis algorithm for biomarker identification in the cloud. The resulting tool, *YunBe*, is ready to use on Amazon Web Services. Moreover, here we compare its performance to those obtained with desktop and computing cluster solutions.

Availability and implementation: *YunBe* is open-source and freely accessible within the Amazon Elastic MapReduce service at s3n://lrcv-crp-sante/app/yunbe.jar. Source code and user's guidelines can be downloaded from http://tinyurl.com/yunbedownload.

Contact: francisco.azuaje@crp-sante.lu

Received and revised on September 22, 2011; accepted on November 7, 2011

1 INTRODUCTION

The complexity and cost associated with recent advances in high-throughput 'omic' technologies have made cloud computing a cost-effective and powerful resource for bioinformatics. Existing platforms, such as those offered by Amazon Web Services (AWS), provide the computing environment, including CPUs, storage, processing memory, networking and operating systems, required to deploy computationally expensive algorithms and applications. Such environments allow users to configure and exploit resources on a 'pay as you use' basis (Fusaro *et al*., 2011). Although cloud computing applications are increasingly being made available for high-throughput DNA sequencing data, there is a need for publicly available algorithms that can enable other translational biomedical research applications, such as large-scale gene set analysis of expression data (Dudley *et al*., 2010). In this context, expression data of thousands of genes are mapped to biologically relevant sets of genes, e.g. curated biological pathways, and differential expression of such gene sets is estimated across phenotypes. The objective of our research is 2-fold: (i) to develop a cloud compute version of a published gene set analysis algorithm (Azuaje *et al*. 2010); and (ii) to perform a comparative analysis of performance across different computing platforms.

2 ALGORITHM

In our original gene set analysis algorithm, *kipuMarkers*, there are two main processing steps (Fig. 1A): expression data overlay and pathway scoring. In the overlay task the aim is to map expression measurements from samples (e.g. patients) onto gene sets (e.g. molecular pathways) provided by the user. In this step 'activity levels' for each sample-specific pathway are calculated. In the case considered here, an activity level refers to the mean expression value observed in a pathway. This is followed by the computation of a 'perturbation score' for each gene set, which is based on the comparison of differential activity levels across two sample groups defined in the data (e.g. disease classification).

We developed a cloud compute version of this method, *YunBe*, which is written in Java using the *MapReduce* framework (Fig. 1B). The overlay step corresponds to a matrix multiplication task: gene expression data matrix (M) is multiplied by a pathway matrix (N) to produce a new matrix (K) with samples matched to the pathways available. Matrix M is uploaded to the *Hadoop* Distributed File System (HDFS), while the matrix N is uploaded to a distributed cache system upon execution. Thus, in each processing iteration, sample data are 'multiplied' by (i.e. mapped to) the entire pathway collection, one by one. Hence, a mapping task processes two inputs: samples (from matrix M) and pathways (from matrix N) to produce a *key/value* pair, where *key* is a pathway ID and *value* is an 'activity value'. The 'reduce' phase connects all *values* associated with the same *key* (pathway ID), followed by the calculation of 'perturbation scores' and *P*-values for each pathway as reported in (Azuaje *et al*., 2010).

3 DATA AND IMPLEMENTATION

To test *YunBe* we analyzed published and simulated gene expression datasets: a human liver gene expression dataset (Schadt *et al*., 2008) and synthetic datasets of varying sizes. The liver dataset includes 466 samples from 31 842 transcripts and grouped by gender, i.e. two-class phenotype. To generate the simulated data, we first selected transcript names lists from the Agilent Whole Human Genome Oligo Microarray (i.e. 19 634 transcript names). We then randomly grouped 1000 samples and computed (normally distributed) values for each transcript in the sample. As gene sets, we used a canonical pathway list with 880 gene sets from the Molecular Signatures DataBase (MSigDB) (Subramanian *et al*., 2005).

Gene expression data, pathways and *YunBe* Jar files were uploaded to an Amazon S3 bucket. A job flow was created with Amazon Elastic MapReduce service (Fig. 1B) with m1.large instance type. A m1.large instance represented a 64-bit platform with two virtual cores. Each virtual core has two EC2 Compute Units that individually equals the CPU capacity of a 1.0-1.2 GHz 2007 Xeon processor. We compared *YunBe*'s execution speeds with a program version running on a computing cluster, which consisted of dual socket quad-core Intel E5430 Harpertown CPUs. In this analysis,

*To whom correspondence should be addressed.

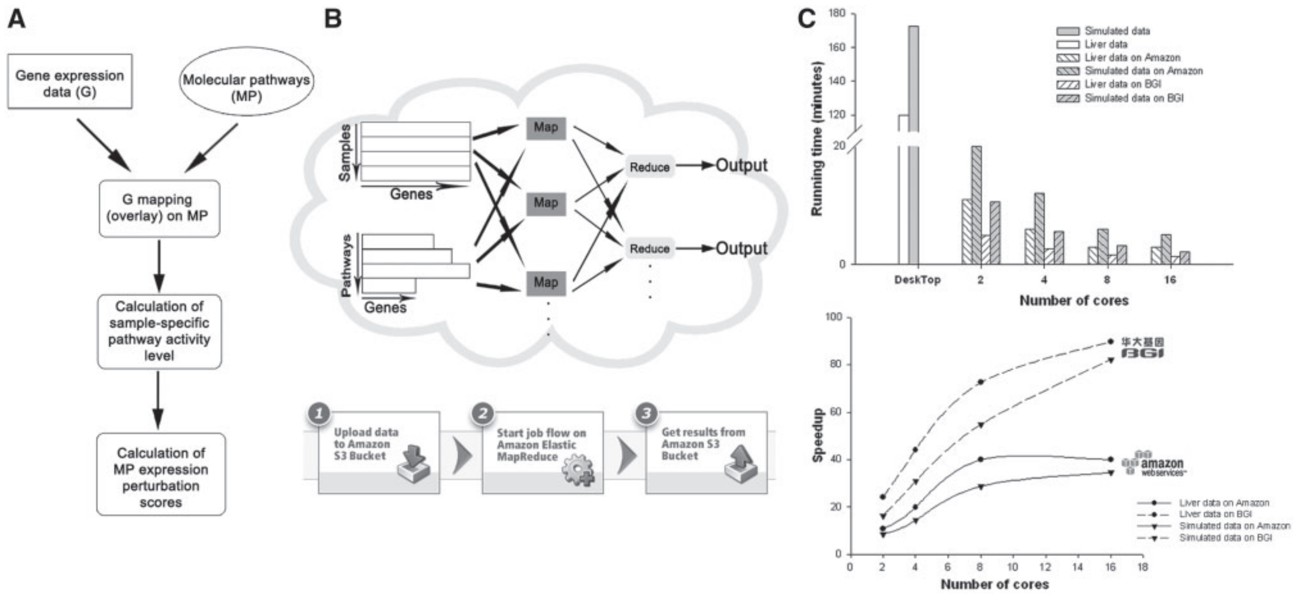

Fig. 1. Gene set analysis in the cloud. (**A**) Analysis pipeline. (**B**) *MapReduce* implementation on Amazon EC2. (**C**) Performance of desktop program, Amazon EC2-based *YunBe* and BGI cluster on liver and simulated data.

1, 2, 4 and 8 EC2 m1.large instances were compared with 2, 4, 8 and 16 cluster cores running at BGI. We also made comparisons with a desktop program running on a duo-core Intel E7500 Wolfdale CPU.

4 COMPARATIVE ANALYSIS RESULTS

The desktop computer version of our algorithm required 120 and 173 min (wall clock time) to perform gene set analysis of the liver and simulated datasets, respectively. In both datasets, *YunBe* reduced the execution time from hours to minutes (Fig. 1C). The execution time was significantly reduced even when using only two cores. In the case of the liver dataset and in relation to the desktop implementation, speedups of 10.9 and 24.1 times were obtained with the Amazon EC2 and BGI cluster, respectively. Major execution time improvements were also observed on the simulated dataset: 8.6 and 16.4 faster with Amazon EC2 and BGI cluster, respectively. Overall, the BGI platform produced faster results than AWS. This result may be expected due to overheads incurred by the cloud's virtualization layer. Another factor to take into account is that speed also depends on the specific hardware utilized for execution. Note that in our analyses, we equalized the bit size of computer architecture (64-bit) and the number of cores between Amazon EC2 and BGI cluster. Nevertheless, other factors, such as memory and I/O performance, may have influenced our comparison. Moreover, differences in networking hardware, inter-node communication and geographical distance should be considered when interpreting observed differences in speed.

YunBe's running time scales with nearly linear speedup over the desktop program performance as the number of cores increases (Fig. 1C). For instance, on Amazon EC2 and liver data, we obtained a speedup of 11 and 20 for 2 and 4 virtual cores, respectively. For the simulated data, the speedup was of 8 and 14 for 2 and 4 cores, respectively. However, such proportional increases were not observed above eight cores, more significantly on Amazon EC2. In theory, *MapReduce* computations are independent, and therefore the (wall clock) running time should scale linearly with the number of processor cores available. In practice, the speedup of a program running on multiple processors may be limited by serial processing overheads, as described by Amdahl's law. *YunBe* analyses on the AWS are relatively inexpensive. For example, a full analysis of the liver dataset requiring eight virtual cores was completed for about US$ 1.7 (~EUR 1.2).

In conclusion, we offer *YunBe*, a new open-source gene set analysis tool for the cloud. *YunBe* is freely available and ready to run on AWS. We showed how, in comparison to a desktop implementation, *YunBe* significantly improves execution times. *YunBe* can accelerate pathway-based biomarker identification through inexpensive and secure distributed computing.

ACKNOWLEDGEMENTS

We thank Y. Devaux, D. Wagner, J.C. Schmit, L. Fang and G. Cao for supporting this Luxembourg-China partnership.

Funding: National Research Fund (FNR) of Luxembourg (AM2c Programme).

Conflict of Interest: none declared.

REFERENCES

Azuaje,F. *et al.* (2010) Integrative pathway-centric modeling of ventricular dysfunction after myocardial infarction. *PLoS One*, **5**, e9661.

Dudley,J.T. *et al.* (2010) Translational bioinformatics in the cloud: an affordable alternative. *Genome Med.*, **2**, 51.

Fusaro,V.A. *et al.* (2011) Biomedical cloud computing with Amazon web services. *PLoS Comput. Biol.*, **7**, e1002147.

Schadt,E.E. *et al.* (2008) Mapping the genetic architecture of gene expression in human liver. *PLoS Biol.*, **6**, e107.

Subramanian,A. *et al.* (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl Acad. Sci. USA*, **102**, 15545–15550.