Cloud4SNP: Distributed Analysis of SNP Microarray Data on the Cloud

Giuseppe Agapito
DSMC
University of Catanzaro
Italy
agapito@unicz.it

Mario Cannataro
DSMC
University of Catanzaro
ICAR-CNR
Italy
cannataro@unicz.it

Pietro Hiram Guzzi
DSMC
University of Catanzaro
Italy
hguzzi@unicz.it

Fabrizio Marozzo
DIMES
University of Calabria
Italy
fmarozzo@dimes.unical.it

Domenico Talia
DIMES
University of Calabria
ICAR-CNR
Italy
talia@dimes.unical.it

Paolo Trunfio
DIMES
University of Calabria
Italy
trunfio@dimes.unical.it

ABSTRACT

Pharmacogenomics studies the impact of genetic variation of patients on drug responses and searches for correlations between gene expression or Single Nucleotide Polymorphisms (SNPs) of patient's genome and the toxicity or efficacy of a drug. SNPs data, produced by microarray platforms, need to be preprocessed and analyzed in order to find correlation between the presence/absence of SNPs and the toxicity or efficacy of a drug. Due to the large number of samples and the high resolution of instruments, the data to be analyzed can be very huge, requiring high performance computing. The paper presents the design and experimentation of Cloud4SNP, a novel Cloud-based bioinformatics tool for the parallel preprocessing and statistical analysis of pharmacogenomics SNP microarray data. Experimental evaluation shows good speed-up and scalability. Moreover, the availability on the Cloud platform allows to face in an elastic way the requirements of small as well as very large pharmacogenomics studies.

Keywords

Pharmacogenomics, Single Nucleotide Polymorphisms, Statistical Analysis, Cloud Computing

1. INTRODUCTION

Among the main *omics* disciplines, genomics studies the activity of genes, proteomics studies the activity of proteins, and interactomics concerns the study of protein interactions inside a cell [3]. In a typical case-control study, microarray technology is able to measure the expression level of genes present in biological case-control samples, that results in a matrix of real numbers where the (i, j) value represents the expression of gene i on sample j. More recently, genotyping microarrays allow to detect the genetic variant among samples, i.e. detect nucleotide variations with respect to a reference population[1]. A nucleotide variation or Single Nucleotide Polymorphism (SNP) is defined as a stable substitution of a single base of DNA[2] occurring with a frequency of more than 1% in at least one population. For instance, in the short sequences ATGT and ACGT a base change occurs at position 2. Each individual has a unique sequence of DNA that determines his/her characteristics and differences can be measured in terms of substitutions of bases in the same position. In a case-control genotyping study, microarray technology produces a matrix of SNPs where the (i, j) vaue represents the SNP found on the DNA sequence or gene i on sample j.

Pharmacogenomics is an important branch of genomics that studies the impact of individual's genetic variations on drug response and is at the basis of the so-called "personalized medicine". It correlates gene expression or SNPs with the toxicity or efficacy of a drug, with the aim to improve drug therapy with respect to the patients' genotype, e.g. allowing to choose drugs matching the genetic profile of each patient. [17]. Pharmacogenomics experiments involve the gene sequencing and the individuation of SNPs by using microarray technology and computational analysis. The DMET (drug metabolism enzymes and transporters) Plus Premier Pack is an Affymetrix[3] microarray platform for gene profiling designed specifically to detect in human samples the presence/absence of SNPs on 225 genes that are related with drug absorption, distribution, metabolism and excre-

[1] Genotyping or genotypization determines differences in the genetic profile of an individual by comparing the individual's DNA sequence with another individual's sequence or with a reference sequence.

[2] DNA is made up of four subunits, or bases, called adenine (A), cytosine (C), guanine (G) and thymine (T).

[3] www.affymetrix.com

tion (ADME) [2].

SNPs data produced by the DMET platform must be preprocessed and analyzed in order to find correlation between the presence/absence of SNPs and the condition of samples (e.g. type of drug treatment or response to a drug). Main steps include: (i) preprocessing of binary microarray data (e.g. a .CEL Affymetrix raw data file for each sample); (ii) aggregation of data coming from all samples of a dataset to form a single table of alleles[4]; (iii) statistical analysis of alleles. For instance, the `apt-dmet-genotype` command line software of the Affymetrix Power Tools suite, or the DMET Console platform [2], generally allow the sequential preprocessing of binary data, but do not allow to test the association of the presence of SNPs with the response to drugs. Consequently, researchers have to export and manually process SNPs tables produced by the DMET Console. Main issues in analyzing DMET SNPs data are: (i) the efficient management of huge data due to the high number of samples and genes investigated in pharmacogenomics studies; (ii) the issues in analyzing SNP symbolic data that need to undergo different preprocessing compared to gene expression numeric data; (iii) the need to provide a result that is biologically interpretable.

Recently we introduced DMET-Analyzer [10], a sequential software tool[5] that supports the statistical analysis of pharmacogenomics data in an automatic way, by providing the list of SNPs that discriminate among two classes of samples according to the Fisher test. This tool has been validated in some pharmacogenomics studies [7, 8], but it is slow when huge datasets have to be analyzed. To solve this limit we designed Cloud4SNP, a novel parallel version of DMET-Analyzer that carries out data analysis on a Cloud platform. This paper presents Cloud4SNP and presents some experiments run on it.

Compared to the sequential version, Cloud4SNP is able to perform statistical tests in parallel, by partitioning the input data set and using the virtual servers made available by the Cloud, thus supporting data parallelism. Moreover, different statistical corrections such as Bonferroni, False Discovery Rate, or none correction, can be applied in parallel on the Cloud, allowing the user to choice among different statistical models, implementing a sort of parameter sweep computation.

The rest of the paper is structured as follows. Section 2 recalls the main software platforms for analyzing SNP genotyping data for pharmacogenomics. Section 3 presents the main functionalities of Cloud4SNP. Section 4 describes the main steps needed to implement Cloud4SNP on the Data Mining Cloud Framework [15]. Section 5 presents the performance evaluation of Cloud4SNP that reports near linear speed-up and scale-up. Finally, Section 6 concludes the paper and outlines future work.

2. RELATED WORK

Microarray technology comprises two main categories of microarray chips: (i) expression microarrays that aim to investigate the activity of genes in different conditions, and (ii) the so called genomic microarrays (such as DMET arrays) that aim to study the variations on sequence of genomes.

The typical dimension of a microarray dataset is growing for two main factors: (i) the increasing of dimension of files encoding a single chip, and (ii) the growing number of samples and then arrays that are usually produced in a single experiment. Let us consider, for instance, two common Affymetrix microarray files (also known as CEL files): the older Human 133 Chip CEL file that has a dimension of 5MB and contains $20,000$ different genes and the newer Human Gene 1.0 st that has a typical dimension of 10 MB and contains $33,000$ genes. On the other hand, a single array of the Exon family (e.g. Human Exon or Mouse Exon) can have up to 100 MB of size. Finally, a recent trend in genomics is to perform microarray experiment considering a large number of patients.

In this scenario arises the need for the introduction of tools and technologies to process such huge volume of data in an efficient way. A way for developing efficient preprocessing of microarray data can be implemented by the parallelization of existing algorithms on multicore architectures. In such a scenario the whole computation is distributed onto different processors, that perform computations on smaller sets of data and results are finally integrated. This requires the design of new algorithms for summarisation and normalisation that take advantage of the underlying parallel architectures. Nevertheless a first step in this direction can be represented by the replication on different nodes of existing preprocessing techniques that run on smaller datasets. Despite its relevance, the parallel processing of microarray data is a relatively new field. In fact, several projects are currently in their initial stage.

One of the main research works is affyPara [18] that is a Bioconductor[6] package for parallel preprocessing of Affymetrix microarray data. It is freely available from the Bioconductor project. Similarly, the micro-CS project [11] presents a framework for the analysis of microarray data based on a distributed architecture made of different web-services internally parallel for the annotation and preprocessing of data. Compared to affyPara, such an approach presents three main differences: (i) the possibility to realize more summarisation schemes such as Plier, (ii) the easily extension to newer SNP arrays, (iii) it does not require the installation of the Bioconductor platform.

EMAAS (Extensible MicroArray Analysis System) [1] is a web-application based on the Rich Internet Application (RIA) paradigm providing to the user management and analysis of Affymetrix arrays. It is based on a high performance computing architecture that uses Grid technology to provide computing resource for the analysis of large datasets. Among its main characteristics, EMAAS (i) is able to process only a subset of Affymetrix Expression arrays (3' and Exon Arrays); (ii) supports collaboration among users; and (iii) requires the upload of data onto the web server, so it

[4] In the rest of the paper the terms SNPs and alleles are used interchangeably.

[5] DMET-Analyzer is available at: http://sourceforge.net/projects/dmetanalyzer/files.

[6] http://www.bioconductor.org/

may require large upload time when Internet speed is not sufficient or it may cause legal problems for SNP data in some countries.

Since the majority of biology laboratories or clinical centres using microarray are not equipped with parallel computers, the parallelization of the microarray data analysis pipeline is not enough. The use of Cloud computing systems can offer also to small research groups the possibility to exploit parallel bioinformatics tools.

The use of Cloud systems as parallel and distributed computing platforms, and the support to visual workflows for high-level programming, are at the basis of the Data Mining Cloud Framework[15][16], the data analysis platform used to implement our Cloud4SNP application. There are other workflow-based systems for scientific applications running on clusters or Grids, including Kepler [13], Pegasus [6], Taverna [12] and Triana [5].

Kepler [13], developed by a team of the University of California, provides a graphical user interface and a run-time engine that can execute workflows either from within the graphical interface or from a command line. It works based on the concept of *directors*, which dictate the models of execution used within a workflow. Kepler is a java-based application that is maintained for the Windows, OSX, and Linux operating systems.

The Pegasus [6] project, developed at the University of Southern California, encompasses a set of technologies to execute workflow-based applications in a number of different environments, i.e., desktops, campus clusters and grids, and clouds. The worfklow management system of Pegasus can manage the execution of complex workflows on distributed resources and it is provided with a sophisticated error recovery system.

Taverna [12] is an open source tool for designing and executing workflows, developed at the University of Manchester. Its own workflow definition language is characterized by an implicit iteration mechanism (single node implicit parallelism). The Taverna team has primarily focused on supporting the Life Sciences community (biology, chemistry and medical imaging) although does not provide any analytical or data services itself.

Triana [5] is a problem solving environment, developed at the Cardiff University, which combines a visual interface with data analysis tools. It can connect heterogeneous tools (e.g. Web services, Java units, JXTA services) on one workflow. Triana uses its own custom workflow language, although it can use other external workflow language representations, which are available through *pluggable* language readers and writers.

Differently from other frameworks that are oriented to general-purpose scientific workflows, the Data Mining Cloud Framework focuses on data mining applications by employing a workflow language specific for this domain. For instance, it provides some data-mining-specific workflow formalisms (Data and Tool arrays) that significantly ease the design of parallel and distributed knowledge discovery applications.

In addition, DMCF has been designed as a Software-as-a-Service (SaaS). This means that no installation is required on the user's machine: the DMCF visual user interface works in any modern Web browser, and so it can be run from most devices, including desktop PCs, laptops, and tablets. This is a key feature for users and who need ubiquitous and seamless access to high-performance data mining services, without having to cope with installation and system management issues.

Recently, two cloud-based platforms for bioinformatics and biomedical applications have been deployed: Galaxy and Globus Genomics.

Galaxy [9] makes available to the user a set of functions for analyzing biomedical data with a special focus on the analysis of genomics sequences. Although the effectiveness of Galaxy, a main limitation on the use of this public server is the fact that data is not encrypted neither during data transfer nor during data storage, while DMET SNP data may be protected by laws in many country for privacy reasons. Moreover, the Data Mining Cloud Framework offers more mining algorithms with respect to Galaxy.

Globus Genomics[7] is based on Amazon Elastic Computing and Galaxy workflows, thus it presents the same limitations mentioned for Galaxy.

3. CLOUD4SNP FUNCTIONALITIES

The goal of Cloud4SNP is to allow to statistically test the significance of the presence of SNPs in two classes of samples using the well known Fisher test. The Fisher test allows to test if two distributions are significantly different, i.e. the eventual difference is not due by chance.

The main steps that Cloud4SNP supports are the following:

1. *Loading of the input dataset and sample class assignment.* The input dataset text file (*SNPs Table*) is a table containing for each sample and for each probe the detected SNPs, as produced by the DMET Console. The SNPs Table is a $np \times ns$ matrix of alleles, where np is the number of probes ($np = 1936$ for current DMET chips) and ns is the number of samples. Table 1 depicts a portion of an example dataset containing the SNPs detected in 8 samples ($S_1, \ldots, S_8$) on 2 probes. Samples are assigned to class A or B by mouse selection or by providing a list of samples for class B. In the example, samples ($S_1, \ldots, S_4$) belong to class A and samples ($S_5, \ldots, S_8$) belong to class B.

2. *Execution of statistical tests (Fisher tests).* The Fisher test is applied to couples of SNPs, e.g. SNP_h vs SNP_k, occurring on a probe $Probe_i$ on classes A and B. The algorithm uses the occurrences of the two alleles SNP_h and SNP_k on each class A and B that are reported in a 2×2 contingency table used to compute the Fisher test. Thus, to perform the Fisher test, Cloud4SNP needs to count the occurrences of the

[7] http://www.globus.org/genomics/solution

SNPs for each probe and class. Tables 2 and 3 contain the occurrences of SNPs in class A and B respectively for dataset of Table 1. For simplicity those tables do not show the occurrences of alleles not present in any of the two classes: such occurrences are of course zero. As an example, Table 4, is the contingency matrix to perform the Fisher test on SNPs A/A vs T/T on probe $Probe_1$, while Table 5 is the contingency matrix to perform the Fisher test on SNPs A/A vs C/T on probe $Probe_2$. Such tests present respectively *p-value* = 0.0286 and *p-value* = 0.3333. The *Fisher Filter threshold* (Ft) is a Cloud4SNP parameter used to accept or not the performed Fisher tests, i.e. Fisher tests results with *p-value* $> Ft$ are discarded and not visualized to the user.

3. *Statistical correction of p-values.* Fisher tests results are ordered by *p-values* and SNPs are annotated with URLs to dbSNP and PharmaGKB. If none statistical correction is selected the results are displayed to the user, otherwise *p-values* are first corrected by applying the proper statistical correction (Bonferroni or FDR) and then displayed to the user.

Table 1: Example SNPs dataset.

SNPs Table								
Probes	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
$Probe_1$	a/a	a/a	a/a	c/t	t/t	t/t	t/t	t/t
$Probe_2$	a/a	a/c	a/a	t/t	a/c	a/c	c/t	t/t

Table 2: Class A SNPs Table.

Class A				
Probes	S_1	S_2	S_3	S_4
$Probe_1$	a/a	a/a	a/a	c/t
$Probe_2$	a/a	a/c	a/a	t/t

Table 3: Class B SNPs Table.

Class B				
Probes	S_5	S_6	S_7	S_8
$Probe_1$	t/t	t/t	t/t	t/t
$Probe_2$	a/c	a/c	c/t	t/t

Table 4: $Probe_1$ A/A and T/T SNPs Occurrences for Fisher Test.

	Class A	Class B
a/a	3	0
t/t	0	4

Table 5: $Probe_2$ A/A and C/T SNPs Occurrences for Fisher Test.

	Class A	Class B
a/a	2	0
c/t	0	1

Cloud4SNP employs an optimization technique to avoid the execution of useless Fisher tests, through the filtering of probes with similar SNPs distributions. In fact, discarding a probe means that the Fisher tests involving the alleles detected on that probe are not computed, thus reducing the computational load of the system. The *Fisher Significance threshold* (Fs) parameter is used to find probes whose SNPs distributions among the two classes A and B are very close. A probe i is discarded if for each allele j, $\|FDT[i,j]\| \leq Fs$. FDT is the *Frequency Difference Table* where the (i,j) element contains the difference among the frequencies of the $j-th$ allele detected on the $i-th$ probe, respectively in class B and in class A. Figure **??**, Table f), shows a portion of the Frequency Difference Table for the dataset of Table a).

Figure 1 summarizes the overall workflow of activities managed by Cloud4SNP:

- *Data Loading and Results Visualization.* The Graphical User Interface manages the interaction with the user and allows the user to provide the input data file name, the assignment of samples to classes (A or B), the *Fisher Significance threshold* (Fs), the *Fisher Filter threshold* (Ft), and the type of statistical correction parameters. Before executing Fisher tests, the user sets the Fs and Ft parameters and selects one of the available multiple test correction, i.e. none, Bonferroni or False Discovery rate (FDR) correction. Multiple tests corrections adjust p-values derived from multiple statistical tests to correct for occurrence of false positives. Details on test corrections may be found in [10]. Then the Graphical User Interface invokes the Cloud4SNP Statistical Tests and the Cloud4SNP Statistical Corrections modules (see Figure 1) to perform statistical tests, and finally it visualizes results.

- *Execution of Statistical Tests.* The Cloud4SNP Statistical Tests module preprocesses input data, eventually filters probes having the same or similar distributions in class A and B by using the Fs parameter, computes statistical tests (Fisher tests), discards tests not statistically significant according to Ft (i.e. the tests whose p-value is greater than Ft), and annotates results with URLs to dbSNP[8] and to PharmaGKB[9].

- *Execution of Statistical Corrections.* The Cloud4SNP Statistical Corrections module performs one of the available multiple tests corrections (none, Bonferroni or False Discovery Rate) that adjust p-values derived from multiple statistical tests to correct for occurrence of false positives. Annotated and eventually corrected results are finally displayed to the user through the Graphical User Interface. In summary, Cloud4SNP Statistical Tests and Cloud4SNP Statistical Corrections are the core modules of Cloud4SNP and provide the analysis of SNP data.

4. CLOUD4SNP IMPLEMENTATION

The implementation of Cloud4SNP starting from DMET-Analyzer has been done using the Data Mining Cloud Framework [15][16], a software environment that allow users to design and execute data mining and knowledge discovery

[8] http://www.ncbi.nlm.nih.gov/projects/SNP

[9] http://www.pharmgkb.org

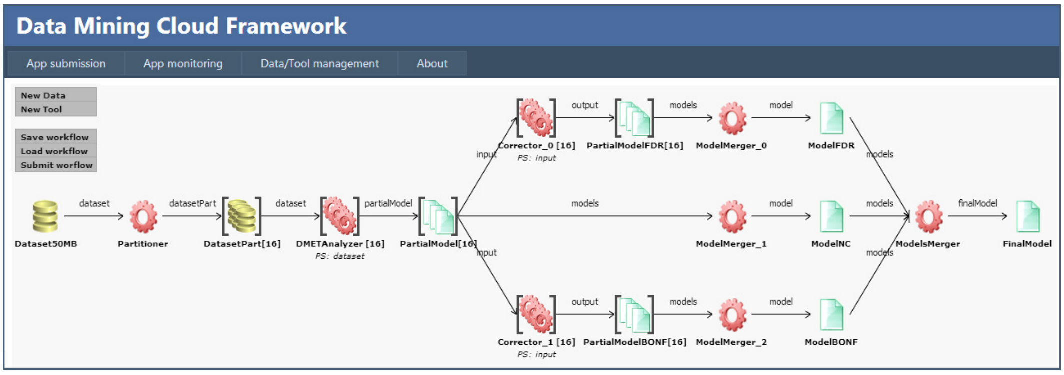


Figure 2: Cloud4SNP workflow in the Data Mining Cloud Framework.

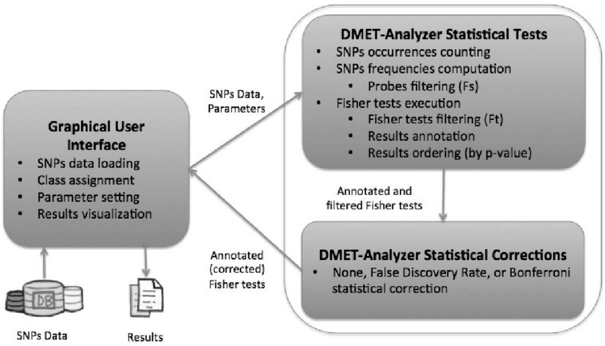

Graphical User Interface
SNPs data loading
Class assignment
Parameter setting
Results visualization
SNPs Data, Parameters
DMET-Analyzer Statistical Tests
SNPs occurrences counting
SNPs frequencies computation
Probes filtering (Fs)
Fisher tests execution
Fisher tests filtering (Ft)
Results annotation
Results ordering (by p-value)
Annotated and filtered Fisher tests
Annotated (corrected) Fisher tests
DMET-Analyzer Statistical Corrections
None, False Discovery Rate, or Bonferroni statistical correction
DB
SNPs Data
Results

Figure 1: Workflow of Cloud4SNP activities.

workflows on the Cloud. In the following we outline the Data Mining Cloud Framework and explain how Cloud4SNP has been implemented on it.

4.1 Data Mining Cloud Framework

Following the approach proposed in [19] and [4], the Data Mining Cloud Framework models knowledge discovery workflows as graphs whose nodes represent resources (datasets, data mining tools, data mining models) and whose edges represent dependencies between resources. The framework includes a Website to compose workflows and to submit their execution to the Cloud, following a Software-as-a-Service approach.

Figure 3 shows the architecture of the Data Mining Cloud Framework, which includes the following components:

- A set of binary and text data containers used to store data to be mined (*Input datasets*) and the results of data mining tasks (*Data mining models*).
- A *Task Queue* that contains the workflow tasks to be executed.
- A *Task Table* and a *Tool Table* that keep information about current tasks and available tools.
- A pool of k *Workers*, where k is the number of virtual servers available, in charge of executing the workflow tasks.

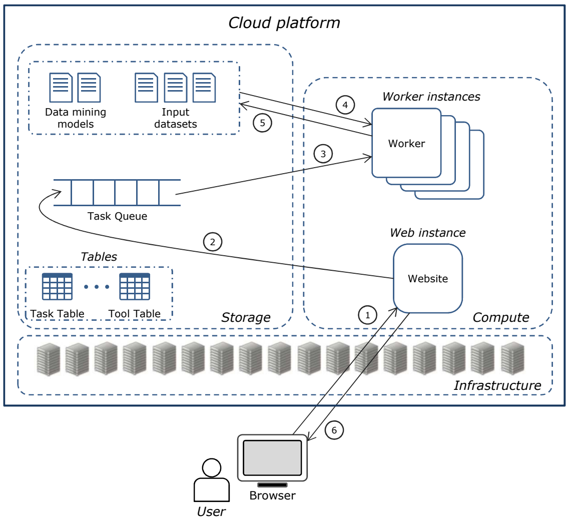


Figure 3: Architecture of the Data Mining Cloud Framework.

- A *Website* that allows users to submit, monitor the execution, and access the results of knowledge discovery workflows.

The following steps are performed to develop and execute a knowledge discovery application through the Data Mining Cloud Framework (see Figure 3)[14]:

1. A user accesses the Website and develops her/his application as a workflow through an HTML-5 interface. A service catalog provides the user with the available data and tools that can be used for application design.
2. After application submission, a set of tasks that compose the workflow are created and inserted into the Task Queue.
3. Each idle Worker picks a task from the Task Queue, and starts its execution on a virtual server.
4. Each Worker gets the input dataset from its original location.

5. After task completion, each Worker puts the result on a data storage element.

6. The Website notifies the user as soon as her/his task(s) have completed, and allows her/him to access the results.

The Data Mining Cloud Framework has been designed to be implemented on different Cloud systems. The current implementation is based on Windows Azure [10].

4.2 Cloud4SNP Tools and Workflow

Since the sequential DMET-Analyzer software was designed as a stand-alone application, we had to extract its main modules and export them as individual tools to the Data Mining Cloud Framework. In particular, starting from the DMET-Analyzer Statistical Tests module we created a *DMETAnalyzer* tool, while the DMET-Analyzer Statistical Corrections module was used to implement a *Corrector* tool. For what concerns the DMET-Analyzer GUI module, it is not used for Cloud4SNP implementation as the user interface is provided by the Data Mining Cloud Framework Website.

In addition to the tools that have been derived from the sequential DMET-Analyzer software, we introduced three new tools in the Data Mining Cloud Framework to support parallel processing of SNP input data: (i) a *Partitioner* tool, which creates a set of partitions from a single SNP dataset; (ii) a *ModelMerger* tool, which merges into a single model the partial models generated by the *DMETAnalyzer*, either corrected or not by a *Corrector*; (iii) a *ModelsMerger* tool, which takes in input three single models (with FDR, Bonferroni or none correction) and produces a single HTML file.

Using the Data Mining Cloud Framework interface, the tools described above have been composed into the Cloud4SNP workflow shown in Figure 2.

The workflow performs the following steps. The initial dataset is partitioned into n parts using the *Partitioner* tool, where n is equal to the number of available Workers (16 in this case). Each part *DatasetPart*$[i]$, $i = 1, ..., n$ is analyzed by an instance of the *DMETAnalyzer* tool (*DMETAnalyzer[i]*) producing a partial result *PartialModel[i]*, which contains the p-values of each probe. The partial models *PartialModel[n]* are corrected using two different instances of the *Corrector* tool: *Corrector_0* that uses an *FDR* correction, and *Corrector_1* that uses a *Bonferroni* correction. Then, three instances of the *ModelMerger* tool are used to create three models, *ModelNC*, *ModelFDR* and *ModelBONF*, which are respectively the model with no corrections (composition of *PartialModel[n]*), the model with FDR correction (composition of *PartialModelFDR[n]*), and the model with Bonferroni correction (composition of *PartialModelBONF[n]*). Finally, the *ModelsMerger* tool combines *ModelNC*, *ModelFDR* and *ModelBONF* to produce a single HTLM file with all the output results. Figure 4 shows the Cloud4SNP workflow at the end of the execution, with the visualization of the final result.

5. PERFORMANCE EVALUATION

This section presents an experimental evaluation of the Cloud4SNP application executed on the Windows Azure platform. The Cloud setting included 1 virtual server to run the Data Mining Cloud Framework Website, and up to 16 virtual servers for the Workers. Each virtual server was equipped with a single-core 1.66 GHz CPU, 1.75 GB of memory, and 225 GB of disk space. Each test has been executed by varying both the size of the input dataset and the number of virtual servers used to run the application.

As performance indicators, we used the *turnaround time* and the achieved *speedup*. To evaluate the scalability with increasing workloads, we analyzed three SNP datasets with size of 12.5 MB, 25 MB and 50 MB. These datasets have a constant number of samples (28), with an increasing number of probes: around 10,000 probes for the dataset of 12.5 MB, 20,000 for the 25 MB dataset, and 40,000 for the 50 MB dataset.

Figure 5 shows the turnaround times of the application for the three datasets using 1 to 16 virtual servers. For the 12.5 MB dataset the turnaround time decreases from around 35 minutes obtained with a single server, to about 2.6 minutes using 16 servers. For the 25 MB dataset the turnaround time passes from 1.2 hours to 5 minutes. With the 50 MB dataset, the turnaround time ranges from about 2.5 hours to about 10 minutes.

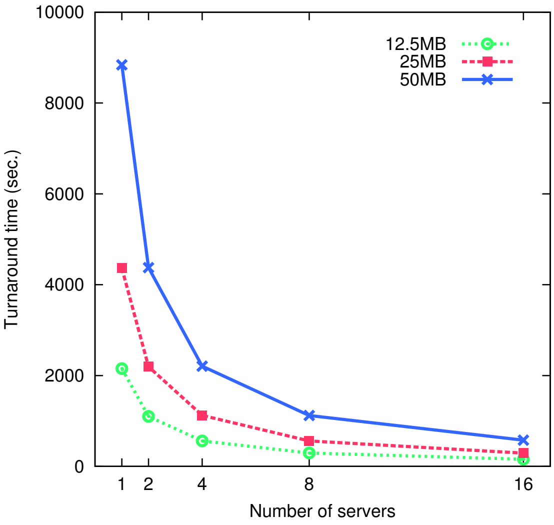
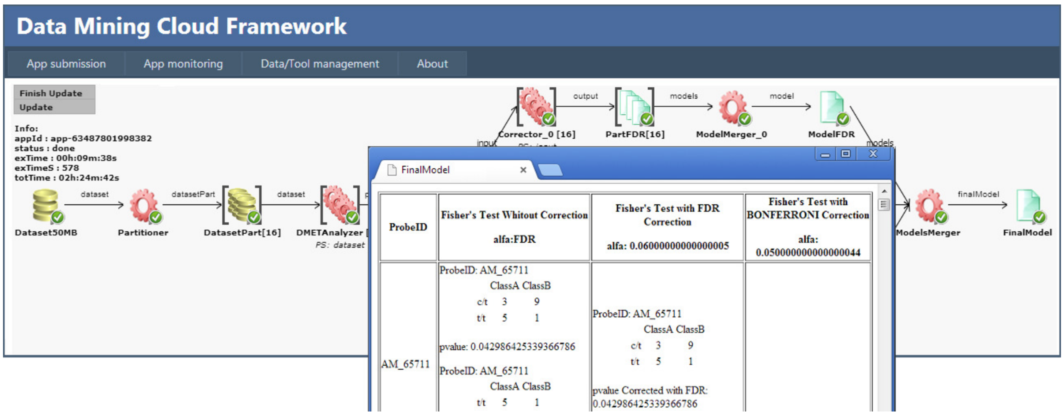


Figure 5: Turnaround times of the Cloud4SNP application.

The corresponding execution speedup values are shown in Figure 6. The speedup is almost linear with all three datasets. In particular, for the 12.5 MB dataset, the speedup passes from 1.9 using 2 servers to 13.7 using 16 servers. For the 25 MB dataset, the speedup ranges from 2.0 to 14.9. Finally, with the 50 MB dataset, the speedup ranged from 2.0 to 15.2.

To better highlight the scalability that can be achieved using the Cloud4SNP approach, Figure 7 measures the system scale-up by showing the turnaround times obtained by

[10] http://www.microsoft.com/windowsazure



Info:
appId : app-63487801998382
status : done
exTime : 00h:09m:38s
exTimeS : 578
totTime : 02h:24m:42s

ProbeID	Fisher's Test Whitout Correction alfa:FDR	Fisher's Test with FDR Correction alfa: 0.06000000000000005	Fisher's Test with BONFERRONI Correction alfa: 0.050000000000000044
AM_65711	ProbeID: AM_65711 ClassA ClassB c/t 3 9 t/t 5 1 pvalue: 0.042986425339366786 ProbeID: AM_65711 ClassA ClassB t/t 5 1	ProbeID: AM_65711 ClassA ClassB c/t 3 9 t/t 5 1 pvalue Corrected with FDR: 0.042986425339366786	

Figure 4: Cloud4SNP workflow at the end of the execution, with visualization of the final result.

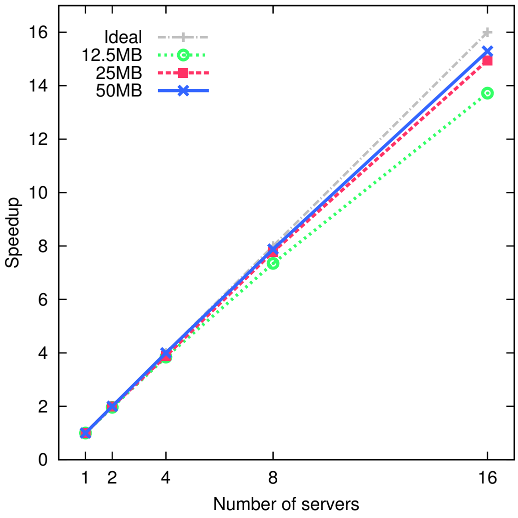


Figure 6: Speedup values of the Cloud4SNP application.

Cloud4SNP when the size of the input dataset increases proportionally to the number of virtual servers used (i.e., 6.25 MB on 1 server, 12.5 MB on 2 servers, and so on). The results show that the turnaround time is almost constant, hovering around 1,100 seconds in all cases. This demonstrates that the amount of data that can be analyzed in a given amount of time increases almost linearly with the number of computing resources available.

6. CONCLUSION

This paper discussed the use of a Cloud-based computing infrastructure for the analysis of SNP microarray data. We presented the design, implementation, and evaluation of Cloud4SNP, a novel Cloud-based bioinformatics tool for the parallel preprocessing and statistical analysis of pharmacogenomics SNP microarray data. Experimental evaluation shows efficient execution times and very good scalability. Moreover, the system implementation shows how the exploitation of a Cloud platform allows researchers and professionals to face in an elastic way the requirements of small as well as very large pharmacogenomics studies.

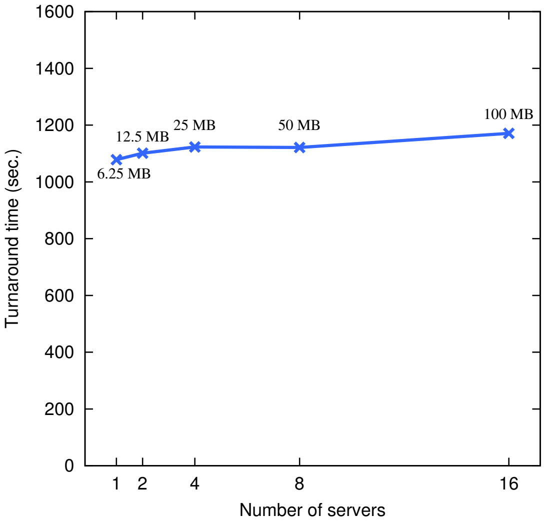


Figure 7: Turnaround times to evaluate the scale-up of Cloud4SNP application.

7. REFERENCES

[1] Barton, G., Abbott, J., Chiba, N., Huang, D., Huang, Y., Krznaric, M., Mack-Smith, J., Saleem, A., Sherman, B., Tiwari, B., Tomlinson, C., Aitman, T., Darlington, J., Game, L., Sternberg, M., and Butcher, S. Emaas: An extensible grid-based rich internet application for microarray data analysis and management. *BMC Bioinformatics 9*, 1 (2008), 493.

[2] Burmester, J. K., Sedova, M., Shapero, M. H., and Mansfield, E. Dmet microarray technology for pharmacogenomics-based personalized medicine. *Microarray Methods for Drug Discovery, Methods in Molecular Biology 632* (2010), 99–124.

[3] Cannataro, M., Guzzi, P. H., and Veltri, P. Protein-to-protein interactions: Technologies, databases, and algorithms. *ACM Comput. Surv. 43*, 1 (2010), 1:1–1:36.

[4] Cesario, E., Lackovic, M., Talia, D., and Trunfio, P. Programming knowledge discovery workflows in service-oriented distributed systems.

Concurrency and Computation: Practice and Experience 25, 10 (July 2013), 1482–1504.

[5] Churches, D., Gombás, G., Harrison, A., Maassen, J., Robinson, C., Shields, M. S., Taylor, I. J., and Wang, I. Programming scientific and distributed workflow with Triana services. *Concurrency and Computation: Practice and Experience 18*, 10 (2006), 1021–1037.

[6] Deelman, E., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Patil, S., Su, M.-H., Vahi, K., and Livny, M. Pegasus: Mapping Scientific Workflows onto the Grid. In *Grid Computing*, M. Dikaiakos, Ed., vol. 3165 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2004, ch. 2, pp. 131–140.

[7] Di Martino, M. T., Arbitrio, M., Guzzi, P. H., Leone, E., Baudi, F., Piro, E., Prantera, T., Cucinotto, I., Calimeri, T., Rossi, M., Veltri, P., Cannataro, M., Tagliaferri, P., and Tassone, P. A peroxisome proliferator-activated receptor gamma (pparg) polymorphism is associated with zoledronic acid-related osteonecrosis of the jaw in multiple myeloma patients: analysis by dmet microarray profiling. *British Journal of Haematology* (2011), 529–533.

[8] Di Martino, M. T., Arbitrio, M., Leone, E., Guzzi, P. H., Saveria Rotundo, M., Ciliberto, D., Tomaino, V., Fabiani, F., Talarico, D., Sperlongano, P., Doldo, P., Cannataro, M., Caraglia, M., Tassone, P., and Tagliaferri, P. Single nucleotide polymorphisms of ABCC5 and ABCG1 transporter genes correlate to irinotecan-associated gastrointestinal toxicity in colorectal cancer patients: A DMET microarray profiling study. *Cancer Biology and Therapy 12*, 9 (November 1 2011), 780–787.

[9] Goecks, J., Nekrutenko, A., Taylor, J., and Team, T. G. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology 11*, 8 (Aug. 2010), R86+.

[10] Guzzi, P. H., Agapito, G., Di Martino, M. T., Arbitrio, M., Tagliaferrri, P., Tassone, P., and Cannataro, M. DMET-analyzer: automatic analysis of affymetrix DMET data. *BMC Bioinformatics 13:258* (Oct. 2012), 258+.

[11] Guzzi, P. H., and Cannataro, M. mu-cs: An extension of the tm4 platform to manage affymetrix binary data. *BMC Bioinformatics 11* (2010), 315.

[12] Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M. R., Li, P., and Oinn, T. Taverna: a tool for building and running workflows of services. *Nucleic Acids Research 34*, suppl 2 (July 2006), 729–732.

[13] Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E. A., Tao, J., and Zhao, Y. Scientific workflow management and the kepler system: Research articles. *Concurr. Comput. : Pract. Exper. 18*, 10 (Aug. 2006), 1039–1065.

[14] Marozzo, F., Talia, D., and Trunfio, P. A cloud framework for parameter sweeping data mining applications. In *Proc. of the 3rd IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2011)* (Athens, Greece, 1 December 2011), pp. 367–374.

[15] Marozzo, F., Talia, D., and Trunfio, P. A cloud framework for big data analytics workflows on azure. In *Proc. of the 2012 High Performance Computing Workshop, HPC 2012*. 2012.

[16] Marozzo, F., Talia, D., and Trunfio, P. Using clouds for scalable knowledge discovery applications. In *Euro-Par Workshops* (Rhodes Island, Greece, August 2012), pp. 220–227.

[17] Phillips, C. SNP Databases. In *Single Nucleotide Polymorphisms*, A. A. Komar, Ed., vol. 578. Humana Press, Totowa, NJ, 2009, ch. 3, pp. 43–71.

[18] Schmidberger M, Vicedo E, M. U. affypara-a bioconductor package for parallelized preprocessing algorithms of affymetrix microarray data. *Bioinform Biol Insights 30*, 22 (2009), 83–7.

[19] Talia, D., and Trunfio, P. *Service-oriented distributed knowledge discovery*. Chapman and Hall/CRC, October 2012.