An Interactive Workflow Generator to Support Bioinformatics Analysis through GPU Acceleration

Anuradha Welivita, Indika Perera, Dulani Meedeniya
Department of Computer Science and Engineering
University of Moratuwa, Sri Lanka
Email: {anuradha, indika, dulanim}@cse.mrt.ac.lk

***Abstract*—Next Generation Sequencing has introduced novel means of sequencing millions of DNA molecules simultaneously and has opened up new avenues in the field of bioinformatics that requires high performance computing technologies. Bioinformatics pipelines are constructed to carry out bioinformatics analyses in a fast and efficient manner. Workflow systems are developed to simplify the construction of pipelines and automate analyses. Still, with the availability of large amounts of sequence data, it has become challenging to have results within a reasonable amount of time. The research proposes a GPU accelerated generic software system to construct bioinformatics workflows. The system allows performing analyses through dedicated GPU computing resources, while incorporating novel web technologies to support specific requirements of bioinformatics software. The results indicate a speedup of x3.11 when a workflow is run on the GPU accelerated system than on a CPU. System usability scale score of 77.5 suggests good usability for the system.**

***Keywords*—Bioinformatics software; biological data analysis; GPU acceleration; cloud computing; Amazon EC2**

I. Introduction

The completion of the Human Genome Project [1] and the introduction of next-generation sequencing (NGS) techniques have geared biologists and bioinformaticians to engage in biological analyses involving large biological datasets. Biological analyses involve multi-step procedures requiring many software tools. It requires feeding the output of a given tool as the input for another. This has intensified the need for creating robust pipelines and workflow generation systems [2].

Bioinformatics pipelines combine multiple software tools sequentially to produce output for a given input(s). With the availability of increasing amounts of biological data, biologists and bioinformaticians often face difficulties in getting results efficiently. This has lead to research on development of distributed and parallel computing infrastructures for bioinformatics analyses [3]. Conducting computer intensive calculations on a computer's Graphics Processing Unit (GPU) is one way of achieving higher degrees of parallelism. GPUs differ from CPUs by having hundreds of cores that can handle thousands of threads simultaneously, whereas CPUs having only a few cores can handle only a few threads at a time. Thus, GPU accelerated computing has become the mainstream for applications whose performance can be enhanced through parallelized computations. Additionally, there are other aspects to be considered, when developing a complete bioinformatics workflow generation system as discussed later.

This paper is structured as follows. Section II explores the literature on automating bioinformatics analyses with hardware accelerators to improve the performance. Section III presents the architectural details of the proposed GPU accelerated system to generate and execute bioinformatics workflows. The gained performance enhancement and other important aspects of bioinformatics software within the proposed system are discussed in Section 4. Section 5 discusses a comparative evaluation of the proposed system against the popular bioinformatics workflow management system, Apache Taverna [6]. Finally, Section 6 concludes the paper.

II. Literature Review

Scripting, makefiles and scientific workflow management systems are used to automate bioinformatics analyses. Scripting is a low level, a less abstract method of generating workflows. It uses basic scripting languages such as Bash, Perl, and Python to construct bioinformatics pipelines [4]. A makefile is a script having a set of rules defining a dependency tree declaratively. It defines the dependencies by linking the output of a rule with the input of another rule. Most commonly used makefiles are Make and CMake [5]. Scientific workflow management systems offer advanced features such as interactive GUI based workflow creation, sharing, and reentrancy. Scientific workflow management systems can easily be handled by both biologists and bioinformaticians with least programming experience. Thus, GUI based scientific workflow management systems such as Apache Taverna [6], Galaxy [7] and Bioconductor [8] are popular. A descriptive evaluation of these analysis techniques are explored in Table I.

Platforms such as Bioconductor [8], BioPython [16], etc. are open source software projects for analyzing genomic data. They provide interfaces to reuse existing libraries and scripts written in other low-level programming languages. This eliminates work redundancy in an analysis process. Yet, they require programming expertise to work with workflow generation, which is a challenge for those who lack programming skills.

Taverna [6] is a GUI based stand-alone desktop application that creates workflows to analyze genomic data. It enables integration of tools distributed across the Internet through web services and graphically making connections between them to construct workflows. It has a web-based platform for workflows sharing. Taverna is platform dependent.

TABLE I
EVALUATION OF BIOINFORMATICS ANALYSIS TECHNIQUES

Technique name	Advantages	Limitations	Examples
Scripting	- Simple to construct - Openness - Ability to execute from command line - Extreme flexibility to manipulate pipelines	- Not support for shared file systems - Development overhead - Hard to determine the exact point of failure - Difficult to reproduce analyses - Difficult to integrate new tools and databases	Bash, Perl, Python [4]
Makefiles	- Simple to construct - Describe the dataflow - Take care of dependency resolution - Commands can be executed in parallel - Cache results from previous runs - State dependencies among files & commands - Lazy processing (checks the modification time of target and prerequisites to avoid repetitions)	- Not flexible compared to scripting - Single wild-card per rule restriction - Cannot describe a recursive flow - Require programming or shell experience - Deceptive error messages - Not support multi-threaded/ multi-process jobs - Limitations on execution on heterogenous failure prone distributed resources	Make, CMake [12], SCans [13], Makeflow [14], Snakemake [15]
Scientific workflow management systems	- Interconnects components - Does not require programming experience - Enable reproducible data analysis - Can simply integrate with HPC systems - Allow execution on distributed resources	- Require more effort - No authority to standardize for interoperability	Galaxy [7], Taverna [6], Biocoductor [8], BioPython [16], NextflowWorkbench [17]

Galaxy [7] is an open web-based platform for genomic research. Galaxy's web-based publication framework, tags and annotations and public repository containing published Galaxy items are unique features that distinguish Galaxy from other counterparts. These features support reproducibility of results and transparency of workflow execution.

Taverna and Galaxy do not enable GPU based computations; provide support for computations on distributed environments, however. The tools used to construct workflows in Galaxy can be configured to be run on cluster nodes. Galaxy has both Amazon cloud support and local grid support to distribute the computing workload [7]. Taverna allows workflows to be executed on remote computational infrastructure [9].

Recent bioinformatics computation literature has stated the use of GPUs with cloud computing. For example, BioCloud enables virtual machines to use GPUs in cloud environments [10]. The highly parallelized nature of GPU computing has led to the use of GPUs in applications that require efficient processing and low latency. The integration of GPU-accelerated computing to the cloud enables to harness the power of GPU computation from the cloud itself and on demand.

This paper proposes a performance enhancement approach for a bioinformatics workflow generation system using Amazon EC2 P2 [11], the largest GPU powered virtual machine in the cloud. It supports to incorporate GPU accelerated tools into workflows. The proposed system uses state-of-art web application technologies to support workflow generation.

III. METHODOLOGY

Scientific workflow management systems such as Taverna [6] and Galaxy [7] are popular in the bioinformatics community. We propose a similar system with enhanced performance that can be extended to a large user base using new technologies. It is developed using JavaScript and NodeJS, as a single page web application using AngularJS. The two-way data binding feature reduces the server-side programming overhead by maintaining a single-page web application.

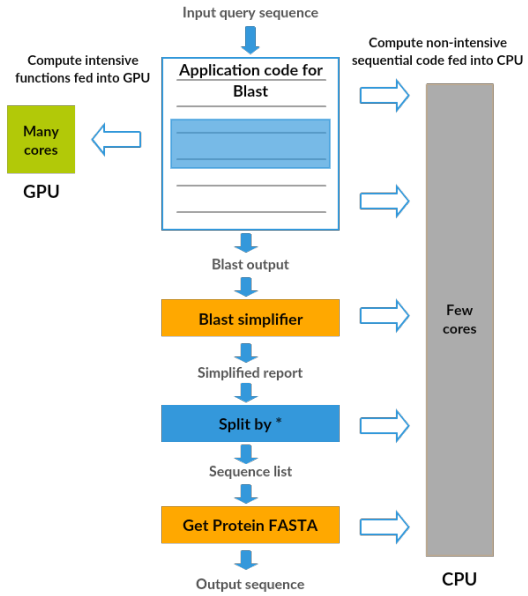

Fig. 1. GPU acceleration process in a simple Bioinformatics workflow.

The application is hosted in an Amazon EC2 P2 instance [11], a GPU accelerated cloud platform with up to 16 NVIDIA Tesla K80 GPUs. Fig. 1. shows the GPU acceleration process. It is scalable and provides parallel computing capabilities using GPUs. GPUs can be used to handle server-side computation workloads in the backend of the bioinformatics workflows. Computing workloads that require high performance and low latency can be run effectively on this instance.

Fig. 2 denotes the high-level architecture of the proposed system. The front-end of the application is developed following the Model-View-Controller (MVC) architectural pattern. Through the application frontend, a user can visually drag and drop components and create a workflow. The application frontend communicates with the jsPlumb javascript library

[20] to visually connent components in the interface and with the D3.js library [21] to visually display results of a workflow.

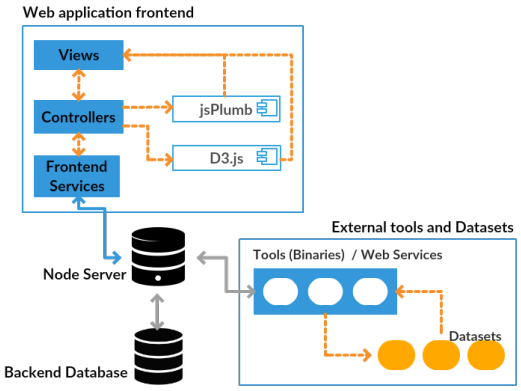

Fig. 2. The high-level architecture view of the proposed application.

The backend node server communicates with locally installed bioinformatics software tools (binaries) and web services to execute the workflows created by the user. The steps in a workflow and the results are stored in a backend SQL database, such that they can be reproduced. The system administrator can configure any number of bioinformatics software tools and web services to be used in a workflow. Their input/output definitions and descriptions can be configured by updating them in a JSON configuration file. The unique strengths of this proposed system are discussed below.

IV. Performance Enhancement

As bioinformatics pipelines run on large biological datasets, support for doing processing on distributed, parallelized computing environments is of utmost importance. GPU accelarated cloud computing supports this by executing the computer intensive application code in GPU powerhouses provided with the cloud instance and rest of the sequential code in CPU cores. The computational load can be distributed on several cloud servers meeting on demand processing requests.

Galaxy has Amazon cloud support as well as local grid support to distribute the computing workload [7]. Taverna web services are mostly distributed and cloud installations of Taverna are also available. These allow workflows to be executed on remote computational infrastructure [9]. However these systems do not enable GPU based processing, which introduces a lag in performance compared to other computation intensive applications that are GPU accelerated.

In order to address this requirement, the application can be hosted in the Amazon cloud, on an Amazon EC2 P2 virtual machine instance. Amazon EC2 is a cloud based instance that enables hosting of HPC applications. P2 is a type of Amazon EC2 cloud instance that supports computations on NVIDIA k80 GPUs. It gives up to 16 GPUs in a single instance. They can be programmed using CUDA [11]. NVIDIA k80 GPUs have been popular for remote rendering for virtual reality, deep learning and financial computations.

With the GPU accelerated cloud platform, it can automatically be scaled across a cluster of nodes depending on the demand. Amazon Elastic Load Balancing (ELB) supports this as shown in Fig. 3 [18]. Amazon ELB is a workaround to distribute incoming application traffic across multiple Amazon EC2 instances in the cloud automatically. Additionally, it integrations with auto scaling to ensure that the application has backend capacity to meet varying levels of traffic without manual intervention, hence high performance.

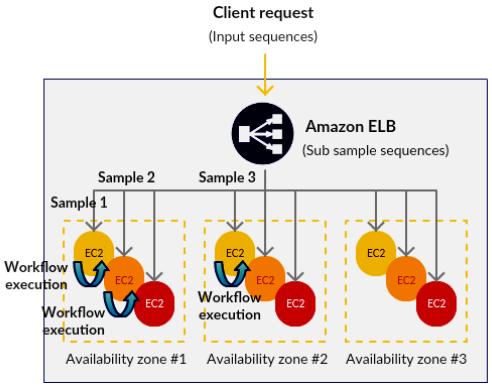

Fig. 3. Amazon Elastic Load Balancer architecture for distributing bioinformatics workload.

It also needs to have a good balance of performance and memory for HPC applications. Since the primary memory and CPU power of the virtual machine where the application is hosted may be used by other processes running in it, having a separate instance for database related operations solves the issues of insufficient memory and processing power. Having a separate Amazon Relational Database Service (RDS) [19] instance that allows easy setting up of scalable relational databases in the cloud is an efficient workaround for this.

V. Key Features for Bioinformatics Workflow Generation

Apart from enhanced performance, there are other features that need to be implemented in a system for generation of bioinformatics workflows. By analyzing different types of bioinformatics software for workflow construction, interactive and graphical workflow creation, module extensibility, reporting, reproducibility and user management could be identified as important features that should exist in a bioinformatics workflow generation system. The proposed system supports these features by incorporating novel web application development technologies. Table II compares the features of the proposed system to Galaxy and Taverna.

A. Interactive and graphical workflow creation

Interactive and graphical workflow creation is one reason behind the popularity of scientific workflow management systems such as Galaxy and Taverna over traditional scripting and makefiles. The proposed system supports this requirement by providing a web based GUI that enables users to dynamically drag and drop components on to an HTML canvas and

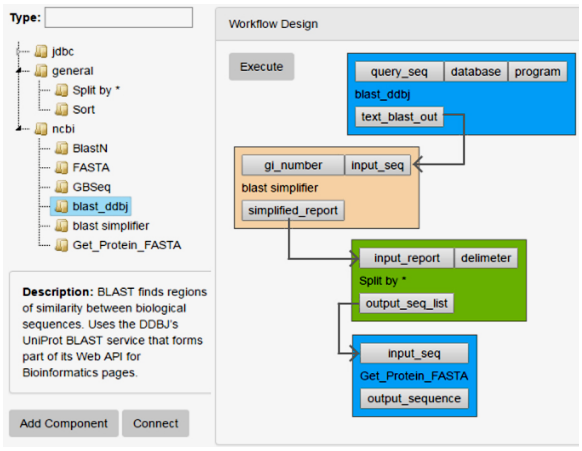

Fig. 4. The GUI of the application containing a simple workflow.

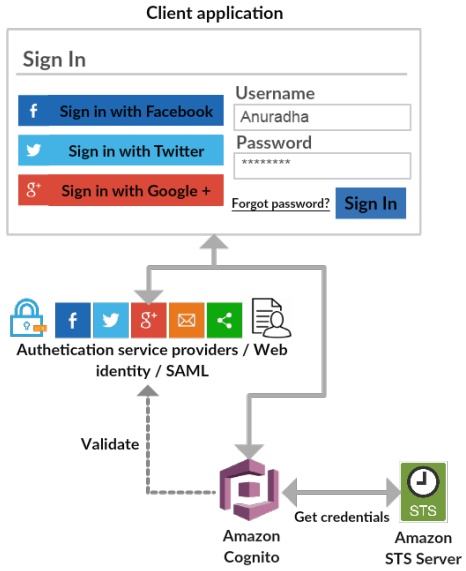

Fig. 5. Amazon Cognito user authentication used in our system

interactively build a workflow. A component is represented by a box and may refer to a locally available tool or a webservice that allows users to access already implemented functions related to bioinformatics analyses. Means for a user to visually connect elements on a web interface is achieved using jsPlumb javascript library. It is a cross browser compatible plugin, which runs on all modern web browsers.

B. Module extensibility

The capability to add or remove data processing or analysing components is known as module extensibility. It is useful to easily adapt workflows to changing research goals. It requires a lot of programming effort when this has to be done with scripting and makefiles. However scientific workflow management systems such as Taverna and Galaxy support this by providing an extensible plugin architecture with access to web services and locally installed tools.

The proposed system encompasses a plugin architecture for service addition as well as service creation. It currently uses the tools of the BLAST+ suite [22], which integrates BLAST-based sequence similarity database searches into workflows. As the proposed system is a JavaScript application based on the modern NodeJS framework, it uses blastjs [23], which is a NodeJS wrapper for BLAST+. This JavaScript and NodeJS based wrappers reduce the client side coding.

C. Reporting

Reporting helps maintaining details of the executed pipelines and summaries of analysis. Both Taverna and Galaxy support generation of reports containing details of pipeline executions and their results [6], [7]. Galaxy generates analysis specific visualizations as well representing genomic data using charts [7]. The proposed system achieves this requirement by means of automatic HTML report generation and uses PhantomJS [24] to generate analysis reports in pdf, automatically.

It uses the D3.js javascript library to generate analysis specific visualizations. D3 has the ability to bind results produced by bioinformatics analysis to a Document Object Model (DOM) and generate genomic data visualizations.

D. Reproducibility

The ability to reproduce or repeat an analysis is important to accelerate new scientific discoveries. Galaxy [7] enables automatic and unobstructive provenance tracking. Taverna [6] also allows sharing of workflows through a web based platform called myExperiment. The proposed system maintains an audit trail along with all the technical meta data including versions of particular software tools used in the analysis and date stamps. While the workflow is being constructed, a separate thread is responsible to record details such as the inputs, processing steps, versions of software tools used etc. each time the workflow is updated. This produces an overview of the entire analysis process as each step is recorded and can be accessed whenever the workflow needs to be reproduced.

E. User management

Having a proper user management and authentication system is important to track and share individual analyses of bioinformatics research scientists, keep track of user data and process quotas. The proposed system uses Amazon Cognito [25] that provides user registration, authentication and data synchronization functionalities for this purpose. It also allows user authentication through social identity providers such as Facebook, Google, Twitter and Amazon. Scaling, secure handling of user management and authentication are automatically taken care of by Amazon Cognito so that the developers can only focus on application development. Fig. 5 shows Amazon Cognito based user authentication as part of the system.

VI. Evaluation and Results

A. Performance evaluation

A performance evaluation for executing a simple workflow generated by the system was carried out. The workflow was

TABLE II
COMPARISON OF THE FEATURES OF THE PROPOSED SYSTEM TO GALAXY AND TAVERNA

Feature	Taverna	Galaxy	Proposed system
1. Performance	- Computation on distributed computing environments.	- Use of Amazon cloud and local grid support to distribute.	- Enhanced performance using GPU-accelerated Amazon cloud services.
2. Interactive graphical workflow creation	- Supported with a GUI based work-bench for composing workflows. - Poor drag & drop of workflow items.	- Supported with a web based graphical workflow editor.	- Supported with a web based GUI for workflow generation on HTML canvas.
3. Module extensibility	- Extensible service plugin architecture for service addition and creation.	- Addition of new web services and con-figuration of new tools for workflows.	- Plugin architecture supporting web services addition, configuration and re-motely installed tools.
4. Reporting	- Results are not graphically shown.	- Analysis specific visualizations.	- An HTML report with PhantomJS. - Visualizations of results using D3.js
5. Reproducibility	- Recording of technical metadata on how each task has been performed (pro-cessor type, status, start & end time).	- Automatic generation of metadata for each analysis step that ensures the re-peatablity of the analysis.	- An audit trail with technical meta data including versions of software tools used in the analysis and date stamps.
6. User management	- Lacks user authentication and manage-ment in the local installation. - Multi-user executions, session mgt. & authentication are not in Server.	- Standard username/password login to authenticate users. - Need additional configuration to sup-port external user authentication.	- Amazon Cognito for authentication, security, session mgt. & scaling. - Authentication through social identi-ties.

TABLE III
HARDWARE SPECIFICATIONS OF THE EVALUATION SYSTEM

Feature	CPU	GPU
Processor	Intel(R) Core(TM) i5-2450M CPU	Nvidia GeForce GT 525M
Processor Frequency	2.50 GHz	1.48 GHz
Cores	2 cores	96 CUDA Cores
Memory	4 GB RAM	1 GB RAM

TABLE IV
CPU VS GPU AVERAGE EXECUTION TIMES FOR THE TEST WORKFLOW

Length of input sequence	Time on CPU (sec.)	Time on GPU (sec.)	Speedup ratio
50	0.019	0.017	1.12
300	0.023	0.018	1.28
600	0.050	0.020	2.50
900	0.060	0.021	2.86
1200	0.073	0.024	3.04
1500	0.081	0.026	3.11

run on top of a GPU enabled Amazon EC2 Linux instance, having CPU and GPU specifications as indicated in Table III, using both remotely installed ncbi-blast [26] and GPU-Blast [27]. GPU-Blast is an accelerated version of ncbi-blast. It produces identical results as of ncbi-blast designed to accelerate gapped and ungapped protein sequence alignments, which is done through modifying the original ncbi-blast code to enable computations on GPUs.

Average execution times obtained by running each input query sequence ten times were recorded and are indicated in Table IV. The speedup ratio, which is the ratio of the CPU execution time to the GPU execution time was also calculated. It shows how much the GPU accelerated system performs faster over the CPU system for the execution of the workflow. Furthermore, these results are depicted graphically in Fig. 6 for ease of comparison. The speedup ratios were also plotted and are indicated in Fig. 7.

From the results obtained, it can be observed that when the input query length is increasing, the speedup ratio is also

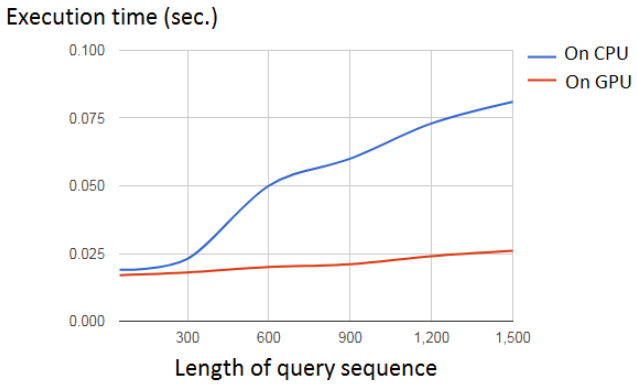

Fig. 6. Comparison between executions times on CPU vs GPU.

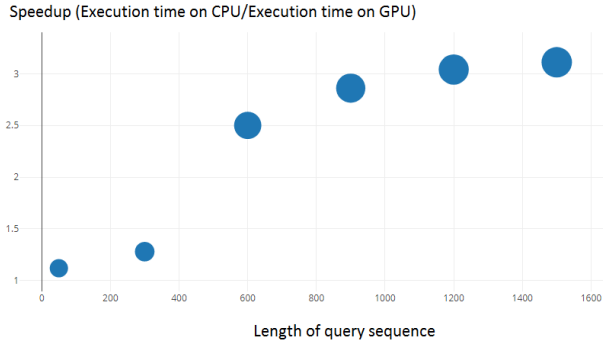

Fig. 7. Average GPU speedup for different lengths of input query sequence.

increasing. When the query length is very small, the time taken only to process the input and give an output may be significantly low compared to the time taken to start the GPU and transfer data between the main memory and the GPU memory. Therefore, significant increase in performance cannot be observed when the query length is small. However for long input queries, about 32% decrease in execution time is gained with GPU acceleration compared to the CPU.

B. Usability evaluation

The proposed web application was evaluated using the System Usability Scale (SUS) [28] and compared against Taverna. Ten subjects in the age group 20-30 having basic knowledge in computing and bioinformatics were asked to create a simple workflow using both Taverna standalone application and the proposed web application and answer the questionnaire based on their experience with the two systems. In order to remove biasness, they were asked to use both systems interchangeably. An average SUS score of 77.5 out of 100 was observed for the proposed system and an average SUS score of 72.5 out of 100 was observed for Taverna. It can be seen that the proposed workflow generation web based system scores relatively better usability than Taverna.

In the open-ended interview, many users indicated the inability to drag individual components in Taverna makes it inflexible to visualize the workflow. As Taverna is desktop based, it has certain dependencies to be pre-installed in the user's local machine. This can be another reason for the preference of the proposed web based platform than Taverna. The comparative evaluation would have been more useful if Galaxy is also compared as it is web based. However, there are access restrictions in the Galaxy's main public server.

VII. Discussion and Conclusion

Bioinformatics often require to analyse massive amounts of biological data, which requires higher computing power and memory. Using Amazon EC2 GPU accelerated cloud platform, this has been addressed in the proposed application gaining significant increase in speed of execution of a workflow.

Using a GPU accelerated cloud platform the system can get instant access to massively parallel computational power. It is cost effective, as it provides flexible means of scaling on demand. With GPUs, the application gains power to process massive datasets, while gaining comparable performance improvements with respect to CPU based systems.

With novel web technologies and the feature rich user interface the system can attract biologists and bioinformaticians. Its intuitive interface does not require programming, hence needs minimum training.

Future research includes exploration of the applicability of Amazon EC2 FPGA based computing instances that can be used to create custom hardware accelerations for the application and whether it gains eve better performance. This work can further be extended by inclusion of usability features such as enabling sharing of user workflows among the community, pipeline comparison and citation support.

References

[1] M. P. Sawicki, G. Samara, M. Hurwitz and E. Passaro, "Human genome project", *The American journal of surgery*, vol. 165, no. 2, pp. 258-264, 1993.

[2] J. Leipzig, "A review of bioinformatic pipeline frameworks", *Briefings in bioinformatics*, pp. bbw020, 2016.

[3] A. Manconi, M. Moscatelli, M. Gnocchi, G. Armano and L. Milanesi, "A GPU-based high performance computing infrastructure for specialized NGS analyses", PeerJ Preprints, 2017. [Online]. Available: https://doi.org/10.7287/peerj.preprints.2175v1. [Accessed: 09- Aug- 2017].

[4] O. Spjuth, E. Bongcam-Rudloff, G. C. Hernndez, L. Forer, M. Giovacchini, R. V. Guimera, A. Kallio, E. Korpelainen, M. M. Kadua, M. Krachunov and D. P. Kreil, "Experiences with workflows for automating data-intensive bioinformatics", *Biology direct*, vol. 10, no. 1, pp. 43, 2015.

[5] B. Cmelik, "Concurrent Make: The Design and Implementation of a Distributed Program in Concurrent C", *Concurrent C Project*, 1986.

[6] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, , M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat and P. Li, "Taverna: a tool for the composition and enactment of bioinformatics workflows", *Bioinformatics*, vol. 20, no. 17, pp. 3045-3054, 2004.

[7] J. Goecks, A. Nekrutenko and J. Taylor, "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences", *Genome biology*, vol. 11, no. 8, pp. R86, 2010.

[8] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry and K. Hornik, "Bioconductor: open software development for computational biology and bioinformatics", *Genome biology*, vol. 5, no. 10, pp. R80, 2004.

[9] K. Wolstencroft, R. Haines, D. Fellows, A. Williams, D. Withers, S. Owen, S. Soiland-Reyes, I. Dunlop, A. Nenadic, P. Fisher and J. Bhagat, "The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud", *Nucleic acids research*, pp. gkt328, 2013.

[10] H. Jo, J. Jeong, M. Lee and D. Choi, "Exploiting GPUs in Virtual Machine for BioCloud", *BioMed Research International*, pp. 1-11, 2013.

[11] "Introducing Amazon EC2 P2 Instances, the largest GPU-Powered virtual machine in the cloud", Amazon Web Services, Inc., 2017. [Online]. Available: https://aws.amazon.com/about-aws/whats-new/2016/09/ [Accessed: Aug- 2017].

[12] B. Cmelik, "Concurrent Make: The Design and Implementation of a Distributed Program in Concurrent C", *Concurrent C Project*, 1986.

[13] W. Deegan, "SCons: A software construction tool", SCons, 2017. [Online]. Available: http://www.scons.org. [Accessed: 13- Jun- 2017].

[14] M. Albrecht et al., "Makeflow: A portable abstraction for data intensive computing on clusters, clouds, and grids", in *Proceedings of the 1st ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies*, pp. 1, ACM.

[15] J. Kster and S. Rahmann, "Snakemakea scalable bioinformatics workflow engine", *Bioinformatics*, vol. 28, no. 19, pp. 2520-2522, 2012.

[16] P.J. Cock et al., "Biopython: freely available Python tools for computational molecular biology and bioinformatics", *Bioinformatics*, vol. 25, no. 11, pp. 1422-1423, 2009.

[17] J. P. Kurs, M. Simi and F. Campagne, "NextflowWorkbench: Reproducible and Reusable Workflows for Beginners and Experts", *bioRxiv*, pp. 041236, 2016.

[18] "AWS — Elastic Load Balancing - Cloud Network Load Balancer", Amazon Web Services, Inc., 2017. [Online]. Available: https://aws.amazon.com/elasticloadbalancing. [Accessed: 09- Aug- 2017].

[19] "Amazon Relational Database Service (RDS) AWS", Amazon Web Services, Inc., 2017. [Online]. Available: https://aws.amazon.com/rds. [Accessed: 09- Aug- 2017].

[20] "jsplumb/jsPlumb", GitHub, 2017. [Online]. Available: http://github.com/jsplumb/jsplumb. [Accessed: 09- Aug- 2017].

[21] M. Bostock, "D3.js - Data-Driven Documents", D3js.org, 2017. [Online]. Available: https://d3js.org. [Accessed: 09- Aug- 2017].

[22] M. Page, D. MacLean and C. Schudoma, "blastjs: a BLAST+ wrapper for Node. js", *BMC research notes*, vol. 9, no. 1, p.130, 2016.

[23] "blastjs", npm, 2017. [Online]. Available: http://www.npmjs.com/package/blastjs. [Accessed: 09- Aug- 2017].

[24] "Automate report generation using PhantomJS - amCharts", amCharts, 2017. [Online]. Available: http://www.amcharts.com/kbase/automate-report-generation-using-phantomjs. [Accessed: 09- Aug- 2017].

[25] "AWS User Authentication and Mobile Data Service — Amazon Cognito", Amazon Web Services, Inc., 2017. [Online]. Available: http://aws.amazon.com/cognito. [Accessed: 09- Aug- 2017].

[26] "BLAST: Basic Local Alignment Search Tool", Blast.ncbi.nlm.nih.gov, 2017. [Online]. Available: https://blast.ncbi.nlm.nih.gov/Blast.cgi. [Accessed: 09- Aug- 2017].

[27] "GPU-Blast — Sahinidis", Archimedes.cheme.cmu.edu, 2017. [Online]. Available: http://archimedes.cheme.cmu.edu/?q=gpublast. [Accessed: 09- Aug- 2017].

[28] A. Affairs, "System Usability Scale (SUS) — Usability.gov", Usability.gov, 2017. [Online]. Available: https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html. [Accessed: 09- Aug- 2017].