HandsFree - Next Generation Sequence Processing, Mapping and Analysis Made Easy

Phillipe Loher[1], Nikos Vasilakis[2], John Malamon[1], Huang-Wen Chen[3] and Isidore Rigoutsos[1,*]

[1] Computational Medicine Center, Thomas Jefferson University, Philadelphia, PA 19107
{Phillipe.Loher, John.Malamon, Isidore.Rigoutsos}@jefferson.edu
[2] Distributed Systems Lab, University of Pennsylvania, Philadelphia, PA 19104
nvas@seas.upenn.edu
[3] Current address: Bloomberg L.P., New York, NY 10022
hwchen@cs.nyu.edu

Abstract. As the speed and throughput of next-generation sequencing improve there is a continuously increasing demand for speedy post-sequencing data analysis. HandsFree is a cloud-based service designed for non-Bioinformaticians that allows users to upload, map and process sequenced RNA-seq, DNA-seq, and ChiP-seq data. The system offers the user a pre-built pipeline that addresses the early, procedural steps of the logical progression from data generation to knowledge generation. HandsFree alleviates the configuration overhead and the need of the identification and installation of analytical tools on a local or rented computing infrastructure while offering an intuitive user-interface that assumes little to no technical expertise and requires minimal input by the user. URL: http://cm.jefferson.edu/HandsFree

Keywords: DNA analysis, RNA analysis, genomic pipeline, mapping service

1 Introduction

Although Moore's Law has been fueling a consistent increase of hardware resources within the 'silicon world', doubling every 18 to 24 months [8], advances in DNA sequencing have been delivering similar increases approximately every 5 months [12]. With the rise of personalized medicine, the focus is quickly shifting from sequencing and mapping to data mining of and reasoning about sequenced datasets and the application of the derived knowledge to prognosis, diagnosis, and therapy.

Basic researchers have to overcome, among other obstacles, the technical barrier of working with big data. However, the speed at which next-generation sequencing burst on the stage means that very few laboratories have the capability required to address this problem. Investigators quickly realize that it is not trivial to set up the necessary computational, analytical, and storage capacity, which in turn forces them to consider core-facilities, if available, or cloud-based solutions [13, 9].

We believe that a considerable portion of those tasks can be easily avoided. Especially considering that the earlier and reasonably-mature-by-now stages of the typical

underlying process namely sequence processing, read mapping and a number of genome-wide analyses can be automated. HandsFree comes to address precisely these considerations by offering streamlined, fast and automated processing that leverages distributed computing capabilities. It offers increased throughput and flexibility, and, by lowering the adoption barrier makes such capabilities easily accessible to practitioners.

HandsFree allows researchers to harvest the computing power of server clusters for post-sequencing processing and analysis by taking advantage of publicly available tool-chains that exist for base-quality-based trimming and mapping (long and short RNA-seq, DNA-seq, ChIP-seq, etc.) to the genomes of model organisms. HandsFree is able to handle sequence data from various platforms such as IonTorrent, Illumina/Solexa, Solid3/4, and Solid 5500xl. Among other features, HandsFree scales up dynamically based on the needs for processing, advises on customization, offers fine-grained process book-keeping and real-time feedback to the administrator of submitted, queued and active jobs, and provides scheduling management. Lastly, we have taken extensive care in making the interface as user-friendly as possible, enabling the users to easily submit, configure, and monitor their jobs.

In what follows we introduce and describe HandsFree, our current implementation, and the system's attributes and capabilities. We also identify a number of challenges, propose a variety of approaches, and outline possible future direction.

2 Related Work

There have been several efforts in the literature to leverage large-scale computing power for genome analysis. For example, Crossbow [5], a software pipeline for sequence analysis, takes advantage of the Apache Hadoop framework, to launch many copies of a short-read aligner in parallel. Other efforts used MapReduce as a programming model [10, 14] and, although all these systems offer comparative throughput at a very low cost, they require considerable technical expertise to set up and use. Galaxy [3], a web-based platform for genome analysis that provides interactive dispatch to external sources, is a first and very significant step towards a friendlier environment. However, Galaxy requires the end-user to understand and create their own pipeline. Similarly, DNAnexus (https://dnanexus.com/) is a cloud-based genomic analysis pipeline as well that is currently designed for use by Bioinformaticians. Unlike the above efforts, HandsFree makes available to the end-user a pre-built pipeline that relies on tools that have been developed, published, and extensively vetted by several research teams.

Another family of initiatives [1, 4] tries to address heterogeneity requirements by introducing virtual machines; however, setting up a dedicated Linux VM introduces a new class of usability challenges for researchers. Pre-configured, hardware-agnostic solutions like VMs incur virtualization penalties and relatively low degree of tuning, offering lower throughput. These solutions require significant computer science knowledge and computer engineering skills on the part of the scientist team that aims to make use of them.

3 System Overview

Figure 1 presents an overview of HandsFree. Broadly speaking, the system comprises three components: user interface, dispatcher/scheduler, and the processing pipeline. The system is built using a compendium of tools implemented in different languages, and glued together using UNIX primitives that handle text streams (e.g., filters, pipes, redirection). The workhorses among these tools are "open source." A running instance of the pipeline is composed of a series of jobs that preprocess, map, and analyze sequencing data. The user-interface, exposed as a set of web services, interacts with the back-end tools using a domain-specific language (DSL) that verifies the correctness of the configuration. Finally, the dispatcher is responsible for scheduling new jobs to the cloud and ensuring fair utilization of resources.

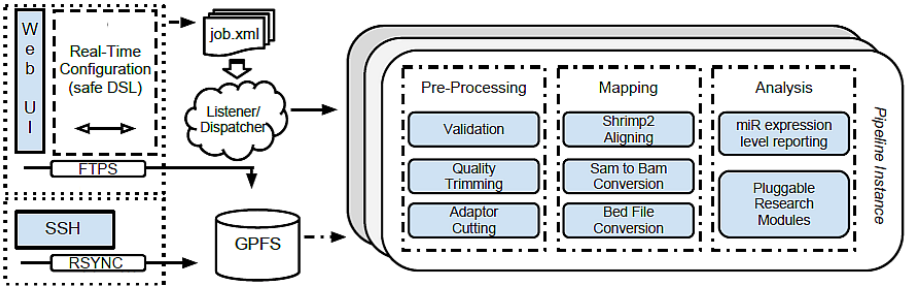

Fig. 1. System Overview

An example usage scenario will help illustrate how the system operates. A first-time user can register and immediately start demoing the capabilities of HandsFree. Once given write-access privileges, the user uploads his/her sequencing data to HandsFree using FTPS, a secure file transport service. Following that, the user through the web interface can create a new job by selecting the uploaded sequencing data and specifying the desired options. A unique sequence of steps is executed, based on the selected options and provided sequencing data. Options include the target organism (human, mouse, etc.), sample type (RNA/DNA), custom adaptor sequences, whether to report SNP's for DNA inputs, etc.. Following that, the service will automatically instantiate the pipeline whereas the resource manager will allocate the required computational resources: if resources are not available, the job will be put in a queue and the user notified accordingly. Software running on the computing nodes updates the logs and user-interface in real time providing the user with immediate progress information. Upon job completion the output is deposited in multiple formats (e.g., Excel, SAM tools, UCSC browser) for download via FTPS and the user is notified by email.

4 HandsFree Architecture

Pipeline: The pipeline consists of a series of chained modules (Figure 1). A pipeline instance is responsible for pre-processing, mapping and analyzing sequenced data, using open-source tools. The pipeline observes interdependencies, with jobs being run only after all dependent tasks have completed. The jobs are modular and can be easily added, removed, or patched. The pipeline can be managed both via web services or the command line, the latter even bypassing the scheduler or assigning static priority, if such an operation mode is needed. Logs are continuously updated, filtered and redirected to various points of interest.

User Interface: HandsFree offers an intuitive and powerful web interface and, thus, cross-device compatibility. Employing asynchronous calls and a DSL, it notifies users in real time and allows them to control their personal workspace and sequence files, to configure and submit new jobs to the system, and to monitor their progress. The interface dynamically generates controls based on the selected options, by pre-running the jobs for a specific configuration: selected options at a given level decide how many and which options become available at the next level. Essentially, a pipeline is run asynchronously that generates the DSL, which is in turn parsed to generate user controls.

Dispatcher: The dispatcher script is the liaison between the user interface and the pipeline, enabling back-end scripts to be independent of the web application. Essentially a daemon process, the dispatcher checks for a job creation or completion, parses relevant metadata and acts accordingly. Upon job creation, it initiates a new pipeline instance, allocates required hardware resources, and schedules the job accordingly. Upon completion, it takes care of garbage collection and bookkeeping, moving around logs, input, results, and accounting information. It is also responsible for notifying interested parties, including administrators and end-users, upon completion.

5 Discussion

A production-mode HandsFree installation has been deployed and is now available to biologists, biochemists, and medical doctors in the local community (Thomas Jefferson University and Hospital, Kimmel Cancer Center, and other Delaware Valley institutions) and beyond. Several analysis modules (e.g. normalized expression estimation for messenger RNAs, microRNAs, pyknons, single-nucleotide polymorphism analysis of IonTorrent data, etc.) are already available to the users. Planned features include, among other things, improved visualization aimed at offering a visual rendering of the current textual output to aid further analyses; enabling increased heterogeneity of the available computational resources by including custom GPU servers, on-demand allocation from academic or commercial cloud pools, and other. In parallel, we will be experimenting on striking a good balance among closely-coupled processing (e.g. MapReduce [2]), loosely-coupled long-running streams executing in parallel (e.g. [7]), or totally independent batch jobs (e.g. [6, 11]) in order to speed-up processing while maintaining a resource agnostic, unified user interface.

References

1. Angiuoli, S.V., Matalka, M., Gussman, A., Galens, K., Vangala, M., Riley, D.R., Arze, C., White, J.R., White, O., Fricke, W.F.: Clovr: A virtual machine for automated and portable sequence analysis from the desktop using cloud computing. BMC bioinformatics 12(1), 356 (2011)
2. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Communications of the ACM 51(1), 107–113 (2008)
3. Giardine, B., Riemer, C., Hardison, R.C., Burhans, R., Elnitski, L., Shah, P., Zhang, Y., Blankenberg, D., Albert, I., Taylor, J., et al.: Galaxy: a platform for interactive large-scale genome analysis. Genome research 15(10), 1451–1455 (2005)
4. Krampis, K., Booth, T., Chapman, B., Tiwari, B., Bicak, M., Field, D., Nelson, K.E.: Cloud biolinux: pre-configured and on-demand bioinformatics computing for the genomics community. BMC bioinformatics 13(1), 42 (2012)
5. Langmead, B., Schatz, M.C., Lin, J., Pop, M., Salzberg, S.L.: Searching for snps with cloud computing. Genome Biol 10(11), R134 (2009)
6. Litzkow, M.J., Livny, M., Mutka, M.W.: Condor-a hunter of idle workstations. In: Distributed Computing Systems, 1988., 8th International Conference on. pp. 104–111. IEEE (1988)
7. Marz, N.: Storm: Distributed and fault-tolerant realtime computation (2012)
8. Moore, G.E.: Cramming more components onto integrated circuits. Proceedings of the IEEE 86(1), 82–85 (1998)
9. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L.,Zagorodnov, D.: The eucalyptus open-source cloud-computing system. In: Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on. pp. 124–131. IEEE (2009)
10. Schatz, M.C.: Cloudburst: highly sensitive read mapping with mapreduce. Bioinformatics 25(11), 1363–1369 (2009)
11. Staples, G.: Torque resource manager. In: Proceedings of the 2006 ACM/IEEE conference on Supercomputing. p. 8. ACM (2006)
12. Stein, L.D.: The case for cloud computing in genome informatics. Genome Biol 11(5), 207 (2010)
13. Walker, E.: Benchmarking amazon ec2 for high-performance scientific computing. Usenix Login 33(5), 18–23 (2008)
14. Wall, D.P., Kudtarkar, P., Fusaro, V.A., Pivovarov, R., Patil, P., Tonellato, P.J.: Cloud computing for comparative genomics. BMC bioinformatics 11(1), 259 (2010)