

# Optimizing High Performance Big Data Cancer Workflows

Ivan Jimenez-Ruiz

University of Puerto Rico, Río Piedras  
Campus  
P.O. Box 70377  
San Juan, PR  
ivan.jimenez2@upr.edu

Ricardo Gonzalez-Mendez

University of Puerto Rico, Medical  
Sciences Campus  
P.O. Box 365067  
San Juan, PR  
ricardo.gonzalez7@upr.edu

Alexander Ropelewski

Pittsburgh Supercomputing Center  
300 South Craig Street  
Pittsburgh, PA (USA)  
ropelews@psc.edu

## ABSTRACT

Appropriate optimization of bioinformatics workflows is vital to improve the timely discovery of variants implicated in cancer genomics. Sequenced human brain tumor data was assembled to optimize tool implementations and run various components of RNA sequence (RNA-seq) workflows. The measurable information produced by these tools account for the success rate and overall efficiency of a standardized and simultaneous analysis. We used the National Center for Biotechnology Information Sequence Read Archive (NCBI-SRA) database to retrieve two transcriptomic datasets containing over 104 million reads as input data. We used these datasets to benchmark various file systems on the Bridges supercomputer to improve overall workflow throughput. Based on program and job timings, we report critical recommendations on selections of appropriate file systems and node types to efficiently execute these workflows.

## CCS CONCEPTS

• **General and reference** → **Cross-computing tools and techniques** → Performance • **Applied computing** → **Life and medical sciences** → **Computational biology** → Computational Genomics

## KEYWORDS

ACM proceedings, Workflows, Performance, Timings, Bioinformatics, File Systems, Memory, Genome, Transcriptome, Supercomputing

### ACM Reference format:

I. Jimenez-Ruiz, R. Gonzalez-Mendez, A. Ropelewski. 2017. In *Proceedings of ACM PEARC conference, New Orleans, USA, July 2017 (PEARC'17)*, 4 pages. <http://dx.doi.org/10.1145/3093338.3093372>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

PEARC'17, July 09-13, 2017, New Orleans, LA, USA

© 2017 Copyright is held by the owner/author(s).

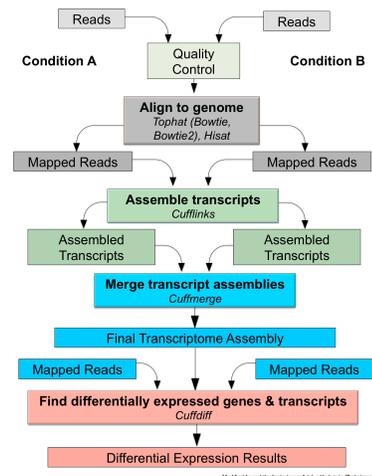
ACM ISBN 978-1-4503-5272-7/17/07.

<http://dx.doi.org/10.1145/3093338.3093372>

## 1 INTRODUCTION

During the last half-decade, advances in Next Generation Sequencing (NGS) technologies have made it possible to rapidly sequence transcriptomes. With NGS, researchers can now generate massive amounts of experimental data inexpensively, most of which must be assembled and analyzed on high performance computers (HPC). The design and construction of efficient workflows to aid in the annotation, assembly, and analysis of NGS data is vital to improve both the analysis throughput and the cost-effectiveness of using HPC for research or clinical purposes.

In this work, we analyze a common RNA-seq transcriptomic workflow (See Fig 1) on Phase-1 of Bridges [2], a newly installed \$9.65M National Science Foundation (NSF) awarded supercomputing system at the Pittsburgh Supercomputing Center.

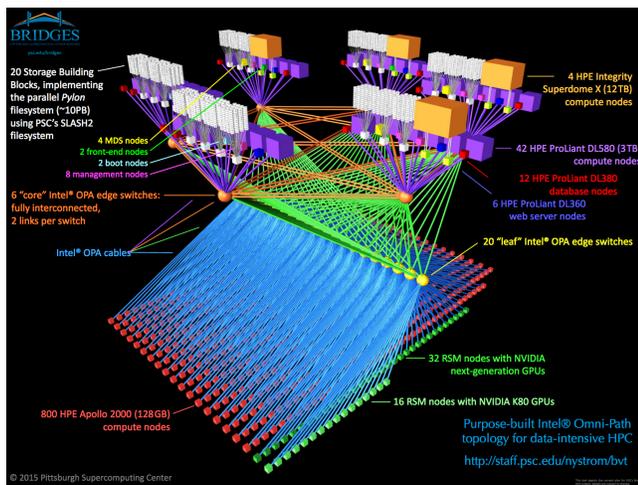


**Figure 1.** This figure shows the steps and programs needed to execute a genome-guided assembly using Cufflinks [1]. This workflow follows protocol from <http://cole-trapnell-lab.github.io/cufflinks/manual/>.

The Phase 1 system consists of tiered, large-shared-memory resources with nodes having 12TB, 3TB, and 128GB each, dedicated nodes for database, web, and data transfer, high-

performance shared and distributed data storage, Hadoop acceleration, powerful new CPUs and GPUs, and the first use of Intel's OmniPath interconnection network in a computing system.

Phase-1 Bridges (see Figure 2) supports four different file systems, two persistent storage systems (/pylon1 and /pylon2) and two volatile file systems (\$LOCAL and \$SRAMDISK) on most node types. The /pylon1 file system is a parallel Luster file system [3] meant to be used as working space for running jobs [4] that is periodically wiped. In contrast, the /pylon2 file system is a persistent file system running Slash2 [5] that was developed for long-term storage of files.



**Figure 2.** In this figure from <https://www.psc.edu/index.php/bridges-virtual-tour>, the implemented OmniPath interconnection and the different node types that comprise the Bridges supercomputer during Phase-1 are shown.

The volatile file systems are used to hold temporary data generated from running jobs. \$LOCAL consists of physical disk drives attached to each Bridges node. Once a job finishes, the \$LOCAL files are inaccessible. Because files must be written and read directly from the file system from within a running job, data must typically be copied to \$LOCAL at the beginning of a job and back out before the job ends [4]. The maximum amount of local space varies by each node and the RSM (128GB) nodes have a maximum of 3.7TB.

The second volatile file system, \$SRAMDISK, uses memory rather than physical disk which can potentially offer the fastest IO. It is comparable to \$LOCAL, except the size available is determined by the physical size of the memory on the nodes reduced by the amount of memory needed by the running program(s). Like \$LOCAL, the space is inaccessible once the job finishes and data that needs to be saved must be copied to the persistent file system before the job ends. If a job terminates abnormally all memory files are lost [4].

## 2 EXPERIMENTAL AND COMPUTATIONAL DETAILS

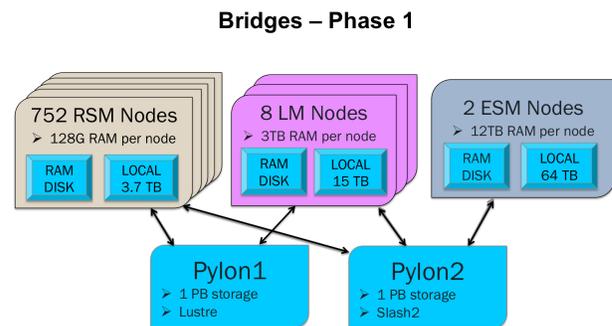
### 2.1 Test Data

To benchmark the workflow, two transcriptomic datasets of glioblastomas in human subjects were retrieved from the NCBI-SRA database: SRR3477485 (3,792 MBases, "primary tumor") and SRR3477486 (6,474 MBases, "recurrent tumor"). The FastQC [6] (v.0.11.2) program was used to examine the quality of the datasets, and showed that quality improvements such as trimming and/or filtering were not necessary.

### 2.2 Benchmarking

The programs implemented to benchmark the file systems included the aligners Bowtie1 (v.1.1.1) [7] and HISAT (v.2.0.4) [8], the mapper TopHat (v.2.2.1) [9], and the transcript assembler Cufflinks (v.2.2.1) [1].

The access patterns of the Bridges file systems are shown in Figure 3. Due to the volatility of the \$SRAMDISK and \$LOCAL file systems, data was copied into these file systems from /pylon1 and /pylon2 upon workflow startup. After the workflows were executed on these file systems, timings based on output to /pylon1 and /pylon2 (individually) were collected.

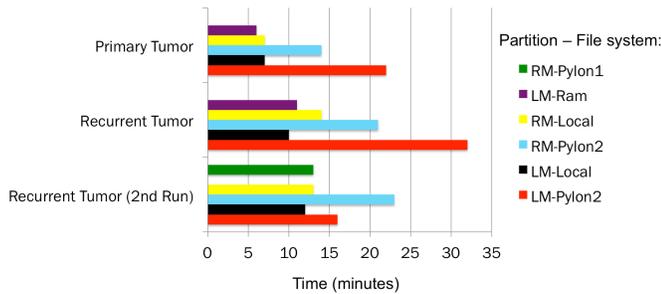


**Figure 3.** In this figure, we show how each file system was accessed to run all programs.

## 3 RESULTS AND DISCUSSION

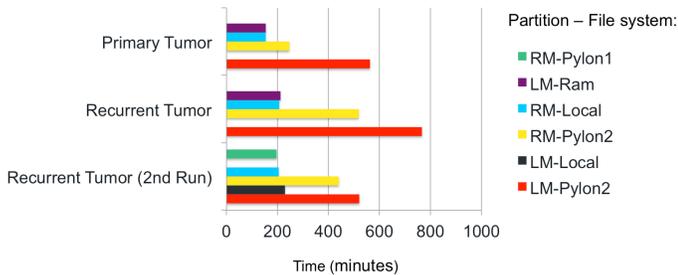
Figures 4, 5, and 6 show timings for the file systems, using the two datasets.

On the primary tumor run (Figure 4), \$SRAMDISK and \$LOCAL all performed comparably. On the timings of \$LOCAL, we saw small variations between runs. The /pylon2 file system was showing the highest variation (around 35%). Comparing the recurrent tumor runs, we saw small variability (around 20%) in the \$LOCAL file systems.



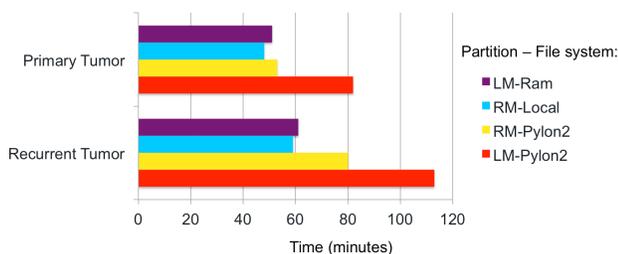
**Figure 4. Timings for mapping: Hisat.** In this figure, we show the timings for the mapping program HISAT [8] where the recurrent tumor test case was run twice.

The Bowtie [7] runs in Figure 5 took substantially longer than the comparable HISAT [8] run with the same dataset. Here, the performances of /pylon1, \$LOCAL and \$RAMDISK file systems were comparable. Again, large variations were observed on the /pylon2 file system.



**Figure 5. Timings for mapping: TopHat (Bowtie1).** In this figure, we show the timings on each file system for TopHat [9], which aligned the reads with Bowtie (v 1.1.1) [7].

In figure 6, we can see that /pylon2 timings on regular memory nodes are more comparable to those of \$LOCAL and \$RAMDISK than the runs on the LM nodes.



**Figure 6. Timings for mapping: Cufflinks (TopHat, Bowtie1).** In this figure, we show the timings for the transcriptome assembler program Cufflinks (v.2.2.1) [1]. The performance of \$LOCAL and \$RAMDISK are comparable.

In this last figure, there is also noticeable variation between the primary and recurrent tumor. From a benchmarking perspective, the /pylon2 file system was comparable to \$LOCAL and \$RAMDISK on regular memory nodes. This might be an indication that the timings seen on /pylon2 are load-dependent.

## 4 CONCLUSIONS

Bioinformatics workflows consist of a series of connected programs that read in and write out data files of varying characteristics. They must frequently be re-engineered to perform optimally on HPC architectures that contain multiple node-types and file systems. Understanding the input and output characteristics of the workflow’s individual programs is a vital step to optimize the performance. Within such workflows, the straightforward choices would be to avoid copying files and carefully weigh the cost of keeping intermediate data files for potential future reanalysis versus re-generating the data files when needed. On a unique four-file-system arrangement such as on Bridges, the decisions become more complex as each of the file systems vary in terms of persistence, size and service charges.

For the test cases outlined in this paper, the volatile file systems \$LOCAL and \$RAMDISK both performed comparably. However, at the current time on Bridges, \$RAMDISK has usage charges associated with it whereas \$LOCAL does not. Our recommendation for similar genomic workflows is to prefer \$LOCAL over \$RAMDISK exclusively for these service charge reasons.

We also found that within the test workflows, the persistent storage system /pylon1 performed similarly to both \$LOCAL and \$RAMDISK. We recommend that the best use of this file system on similar workflows would be for staging results and as intermediate storage for output files. We note here that /pylon1 has been recently replaced on Bridges with /pylon5, which is similar in architecture and performance to /pylon1. Our recommendations for the best use of /pylon5 follow those of /pylon1.

Overall within our tests the /pylon2 file system had the most variability and worst performance. We believe that this is due to a combination of /pylon2 system-wide I/O load and due to the non-native OmniPath mounting of Slash2 on the Phase 1 Bridges system. Our recommendation for using the file system on a similar workflow would be to use /pylon2 for long-term storage and archiving needs.

## 5 ACKNOWLEDGMENTS

We would like to thank Dr. David Boone and Dr. Uma Chandran from the Department of Biomedical Informatics at the University of Pittsburgh and Dr. Humberto Ortiz from the Department of Computer Science at the University of Puerto Rico, Rio Piedras Campus (UPRRP) for their support.

This work was supported by the NIH Big Data to Knowledge

(BD2K) Enhancing Diversity in Biomedical Data Science Grant 5R25MD010399-002 to the UPRRP and the National Institutes of Health Minority Access to Research Careers (MARC) grants T36-GM-095335 to the Pittsburgh Supercomputing Center, as well as the National Institutes of Health Biomedical Technology Resource grant P41-GM-103712. The computing resources used were provided through the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by the National Science Foundation grant OCI-1053575. Specifically, it used the Bridges supercomputer system at the Pittsburgh Supercomputing Center (PSC) acquired through NSF Award ACI-1445606.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

## 6 REFERENCES

- [1] C. Trapnell, B.A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M.J. van Baren, et al. 2010. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat Biotechnol* 28(5):511-5. DOI: 10.1038/nbt.1621.
- [2] N.A. Nystrom, M.J. Levine, R.Z. Roskies, and J.R. Scott. 2015. Bridges: a uniquely flexible HPC resource for new communities and data analytics. In *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure (XSEDE '15)*. ACM, New York, NY, USA, Article 30, 8 pages. DOI: <http://dx.doi.org/10.1145/2792745.2792775>
- [3] J. Corbet. 2003. "*Lustre 1.0 released*". *Linux Weekly News*. *LWN.net*. [Cited March 13, 2017] Available from: <https://lwn.net/Articles/63536/>
- [4] S. Williams. 2011. Website: Pittsburgh Supercomputing Center (PSC). [Cited March 13, 2017]. Available from: <https://www.psc.edu/homepage/about-psc>.
- [5] P. Nowoczynski, J. Sommerfield, J. Yanovich, J.R. Scott, Z. Zhang, and M. Levine. 2012. The data supercell. In *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the campus and beyond (XSEDE '12)*. ACM, New York, NY, USA, Article 13, 11 pages. DOI: <http://doi.acm.org/10.1145/2335755.2335805>
- [6] S Andrews. 2010. FastQC: a quality control tool for high throughput sequence data. (Version 0.11.2) [Cited March 13, 2017]. Available from: <http://www.bioinformatics.babraham.ac.uk/projects/fastqc>.
- [7] B. Langmead, C. Trapnell, M. Pop, S.L. Salzberg. 2009. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* 10(3):R25. DOI: 10.1186/gb-2009-10-3-r25.
- [8] D. Kim, B. Langmead, S.L. Salzberg. 2015. HISAT: a fast spliced aligner with low memory requirements. *Nat Methods*. 12(4):357-60. DOI: 10.1038/nmeth.3317.
- [9] C. Trapnell, L. Pachter, S.L. Salzberg. 2009. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*. 25(9):1105-11. DOI: 10.1093/bioinformatics/btp120.