Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Genome analysis

Genome analysis

Automatically parallelizing bioinformatics workflows with Cannoli

Michael Heuer[1,*] **and Frank Austin Nothaft**[2]

[1] Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA and
[2] Databricks, Inc., San Francisco, CA 94105, USA

* To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Due to their computational complexity, secondary bioinformatics pipelines can take more than a day to run. While accelerated implementations of popular tools like the GATK exist, these implementations are typically proprietary and cover a limited set of bioinformatics workflows. Bioinformaticians frequently resort to manual methods to run an analysis in parallel, such as writing scripts that split by genomic locus. These scripts add complexity to maintaining a pipeline and may not achieve optimal scaling.
Results: Cannoli provides a user-friendly API and CLI that automatically parallelizes 21 common bioinformatics. Cannoli builds on top of Apache Spark and ADAM's pipe API, which provides fault-tolerant execution portably across a local machine, an on-premises compute farm, and cloud computing. Benchmarking on common variant calling and single-cell RNA-seq quantification pipelines demonstrates that Cannoli can reduce workflow runtime by FIXME×
Availability: Cannoli is open-source software, distributed under an Apache 2 license. Cannoli is available from `https://github.com/bigdatagenomics/cannoli`.
Contact: heuermh@berkeley.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

- Bioinformatics workflows are slow and complex.
  - Often involve many tools chained together, running on a single node
  - Bioinformaticians often rely on one-off scripts to parallelize tools
  - These scripts add complexity to a workflow and make workflows liable to fail
  - Since we are often parallelizing by genomic locus, we can automatically parallelize most tools
- Cannoli provides automatic parallelization of bioinformatics workflows in an easy-to-use and composable framework
  - Cannoli is built on top of Apache Spark (Zaharia *et al.*, 2012) and ADAM (Massie *et al.*, 2013; Nothaft *et al.*, 2015)
  - Cannoli supports 21 common bioinformatics tools using ADAM's pipe API (Nothaft, 2017), which allows a user to run tools in parallel across a cluster, with built-in fault tolerance
  - Each tool invocation takes a single line of code, and Cannoli uses Docker to simplify tool installation (da Veiga Leprevost *et al.*, 2017)
- In this application note, we walk through Cannoli's architecture and evaluate it on two pipelines
  - We implement a germline variant calling use case using BWA and FreeBayes through Cannoli
  - We implement an end-to-end scRNA-seq pipeline using Cannoli and Apache Spark SQL

2 Approach

- Cannoli provides both an API and a CLI for running a set of 21 bioinformatics tools
  - Cannoli wraps each tool in a `CannoliFn`, a one-line command that can be called in Scala to run the command
  - A `CannoliFn` transforms an ADAM (Massie *et al.*, 2013; Nothaft *et al.*, 2015) dataset into a new ADAM dataset

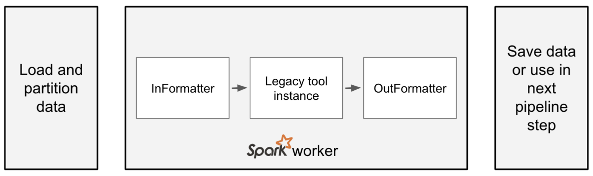

Fig. 1. Schematic of ADAM's pipe function.

- Inside a `CannoliFn`, we build a command for the tool we are running, and pass this command to ADAM's pipe API

Table 1. Tools supported in Cannoli

Aligners	
	bowtie bowtie2 bwa gem minimap2 star
Variant Callers	freebayes samtools mpileup
Variant Manipulators	snpeff vep bcftools vt
Other	bedtools blastn magic-blast

- This architecture makes it very easy to support a large number of tools
  - `CannoliFn`s have access to a `CommandBuilder`, which strings together arguments for a bioinformatics tool according to configured parameters
  - `CommandBuilder`s also abstract away whether the command is run using a Docker container—defaulting to a BioContainer (da Veiga Leprevost *et al.*, 2017)—or a pre-installed executable
  - Once a `CannoliFn` is built, it is accessible on top of an ADAM dataset and a thin wrapper exposes the function as a command-line tool
- The simplicity of this approach has allowed us to add support for the 21 tools described in Table 1.

3 Methods

Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text. Figure 2 shows that the above method Text Text Text Text Text Text Text Text Text Text Text Text. **?** might want to know about text text text text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text. Figure 2 shows that the above method Text Text Text Text Text Text Text Text Text Text Text Text. **?** might want to know about text text text text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text. Figure 2 shows that the above method Text Text Text Text Text Text Text Text Text Text Text Text. **?** might want to know about text text text text

4 Discussion

- We see Cannoli as an alternative to a workflow manager, but not inherently competitive

- Large number of existing workflow runners, many of which focus on metadata capture, some of which extend existing programming paradigms, some of which introduce new domain-specific languages, none of which focus on automatic parallelization
- Cannoli does not focus on metadata capture and extends an existing programming paradigm (ADAM/Scala), but instead provides automated parallelization
- Cannoli's API enables users to rapidly build and run pipelines that include ad hoc manipulations of data (e.g., align reads and then filter out low map-Q reads, call variants and apply region-specific predicates)
- This is useful for rapid experimentation, and eliminates common workflow smells (e.g., pipe VCF through `grep` to do variant filtration)
- Cannoli's CLI enables straightforward integration with existing workflow managers

5 Conclusion

- Cannoli improves the latency of bioinformatics tools, while improving ease of use and reproducibility

- Cannoli's API allows users to run 21 tools, with a single line of code per tool, which allows users to simply compose workflows
- These APIs are also exposed through a command-line, enabling easy interoperability with traditional bioinformatics workflows
- These APIs can easily be extended to support new bioinformatics tools
- We have demonstrated how the API and CLI can be used to accelerate variant calling and scRNA quantification workflows by FIXME×

Acknowledgements

Work on Cannoli started at the Bioinformatics Open Source Conference codefest, and was encouraged along by members of the Big Data Genomics project, as well as the broader open-source bioinformatics community.

Funding

This work was supported by NSF CISE Expeditions Award CCF-1139158, LBNL Award 7076018, and DARPA XData Award FA8750-12-2-0331, NIH BD2K Award 1-U54HG007990-01, NIH Cancer Cloud Pilot Award HHSN261201400006C and gifts from Amazon Web Services, Google, SAP, The Thomas and Stacey Siebel Foundation, Adatao, Adobe, Apple, Inc., Blue Goji, Bosch, C3Energy, Cisco, Cray, Cloudera, EMC, Ericsson, Facebook, Guavus, Huawei, Intel, Microsoft, NetApp, Pivotal, Samsung, Splunk, Virdata, VMware, and Yahoo!.

References

da Veiga Leprevost, F., Grüning, B. A., Alves Aflitos, S., Röst, H. L., Uszkoreit, J., Barsnes, H., Vaudel, M., Moreno, P., Gatto, L., Weber, J., *et al.* (2017). Biocontainers: an open-source and community-driven framework for software standardization. *Bioinformatics*, **33**(16), 2580–2582.

Massie, M., Nothaft, F., Hartl, C., Kozanitis, C., Schumacher, A., Joseph, A. D., and Patterson, D. A. (2013). ADAM: Genomics formats and processing patterns for cloud scale computing. Technical report, UCB/EECS-2013-207, EECS Department, University of California, Berkeley.

Nothaft, F. A. (2017). *Scalable systems and algorithms for genomic variant analysis*. Ph.D. thesis, UC Berkeley.

Nothaft, F. A., Massie, M., Danford, T., Zhang, Z., Laserson, U., Yeksigian, C., Kottalam, J., Ahuja, A., Hammerbacher, J., Linderman, M., Franklin, M., Joseph, A. D., and Patterson, D. A. (2015). Rethinking data-intensive science using scalable analytics systems. In *Proceedings of the International Conference on Management of Data (SIGMOD '15)*. ACM.

Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M., Shenker, S., and Stoica, I. (2012). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the Conference on Networked Systems Design and Implementation (NSDI '12)*, page 2. USENIX Association.