RESEARCH NOTE

Hadoop-CNV-RF: a clinically validated and scalable copy number variation detection tool for next-generation sequencing data

Getiria Onsongo[1*], Ham Ching Lam[2], Matthew Bower[3,4] and Bharat Thyagarajan[4,5]

[*] Correspondence:
gonsongo@macalester.edu
[1] Department of Mathematics, Statistics, and Computer Science, Macalester College, Saint Paul, MN, United States
Full list of author information is available at the end of the article

Abstract

Objective: Detection of small copy number variations (CNVs) in clinically relevant genes is routinely being used to aid diagnosis. We recently developed a tool, CNV-RF, capable of detecting small clinically relevant CNVs. CNV-RF was designed for small gene panels and did not scale well to large gene panels. On large gene panels, CNV-RF routinely failed due to memory limitations. When successful, it took about 2 days to complete a single analysis, making it impractical for routinely analyzing large gene panels. We need a reliable tool capable of detecting CNVs in the clinic that scales well to large gene panels.

Results: We have developed Hadoop-CNV-RF, a scalable implementation of CNV-RF. Hadoop-CNV-RF is a freely available tool capable of rapidly analyzing large gene panels. It takes advantage of Hadoop, a big data framework developed to analyze large amounts of data. Preliminary results show it reduces analysis time from about 2 days to less than 4 hours and can seamlessly scale to large gene panels. Hadoop-CNV-RF has been clinically validated for targeted capture data and is currently being used in a CLIA molecular diagnostics laboratory. Its availability and usage instructions are publicly available at: https://github.com/getiria-onsongo/hadoop-cnvrf-public.

Keywords: Copy Number Variation; Next-Generation Sequencing; Hadoop; High-Performance Computing; Amazon Web Services

Introduction

A recent comparative analysis of copy-number variant (CNV) detection tools by Roca et al laments the lack of suitable tools for CNV detection in the clinic using targeted next-generation sequencing (NGS) data[1]. They conclude there is no one tool that is best suited for CNV detection using custom targeted NGS. Most of the tools developed are for either whole-genome or whole-exome NGS data. In theory, tools for analyzing whole-exome data can be used to analyze targeted panels. Both datasets, whole-exome and custom targeted panels, share similar characteristics such as uneven coverage over targeted exons. A major limitation of using tools developed for whole-exome data in the clinic, especially on larger gene panels, is the large number of false positives generated. The few that have mechanisms for reducing false positives have restrictions that make their use impractical for small clinical labs without large computation infrastructure. XHMM[2], GermlineCNVCaller[3], and DECoN[4], for example, overcome the false positive problem by requiring that samples be analyzed in cohorts. XHMM requires that samples be analyzed in cohorts of at least 10, GermlineCNVCaller requires that samples be analyzed in cohorts of at least 30 samples, and DECoN recommends using 3 to 5 control samples. This requirement is impractical for small clinical labs as patient samples are analyzed individually and each patient may have a unique gene panel ordered. Additionally, all the aforementioned tools require aligned reads as input (analysis ready BAM files) leaving it to the user to develop a pipeline for integrating alignment and CNV detection. We need a freely available tool for identifying clinically relevant small CNVs that takes as input FASTQ files, scales to large gene panels, and is suitable for clinical use.

We recently developed a tool, CNV-RF[5], capable of detecting clinically relevant CNVs. CNV-RF was designed for small gene panels and did not scale well to large gene panels. Analyzing large gene panels routinely failed due to memory limitations. When successful, it took on average over 30 hours to complete the analysis. In practice, in a clinical setting where results are often relied upon to make clinical decisions, a failed analysis that needed re-analyzing could mean delaying a critical clinical decision by several days. To address this limitation, we have re-implemented CNV-RF using Java[6] to take advantage of Hadoop[7] in a new tool we call Hadoop-CNV-RF. Hadoop is a framework that enables programmers without experience in distributed systems to easily scale analysis to large distributed clusters of commodity hardware. Preliminary results show Hadoop-CNV-RF reduces runtime for analysis of targeted custom capture data from over 30 hours to about 4 hours and can easily scale to large gene panels.

Main text

Materials and Methods

Data

For the run time analysis comparison, we used targeted sequencing data using the TruSightOne Expanded Panel (Illumina, San Diego, CA), which enriches for coding exons in 6794 genes (target region size, 16.6 Mb). Sequencing was performed using a 2 x 150 bp paired-end read on a Novaseq instrument (Illumina, San Diego, CA). For comparative analysis, five samples (with matched controls) were analyzed on Amazon Web Services (AWS)[8] using both CNV-RF and Hadoop-CNV-RF and running times compared.

For the clinical validation, sequence data from a custom Agilent capture targeting 3,940 genes (target region size 9.6MB) was used. Prepared libraries were sequenced using a 2 x 150 bp paired-end read on a Novaseq instrument (Illumina, San Diego, CA).

Implementation

The algorithm implemented in Hadoop-CNV-RF and its dependencies was previously described in detail in CNV-RF[5]. We re-implemented steps that did not rely on third party software such as BWA[9] using the Map-Reducing framework thus making then inherently parallel. For steps that relied on third party software designed to run on a single-node, we were able to speed analysis by dividing input data into smaller subsets which were analyzed in parallel. Using BWA as an example below, we describe how we scaled such third party software used in the pipeline.

Scaling BWA Software on Hadoop

There are three main steps to scaling single-node software on a Hadoop cluster: 1) splitting the problem into smaller independent problems, 2) sending the smaller subproblems to different nodes, and 3) combining results solutions of subproblems into the final result.

Step 1: *Splitting the Problem*

Reads from input FASTQ files are split into smaller roughly equal partitions corresponding to the number of nodes in a cluster. Fig 1 is an example of how a FASTQ file is split assuming a user has a three node cluster.

Step 2: *Sending jobs to different nodes*
Hadoop has a distributed file system (HDFS) that is accessible by all nodes. To share data across nodes, input dataset partitions are uploaded to HDFS together with other datasets needed to execute a given program. For BWA, partitioned FASTQ files together with a reference genome are uploaded to HDFS.

To launch jobs on different nodes using partitioned datasets as input, Hadoop-CNV-RF takes advantage of Hadoop's MapReduce framework. This framework distributes jobs to different nodes and monitors progress, automatically restarting failed jobs until the task is complete.

Step 3: *Combining results*
Techniques for combining results are task-dependent. For BWA, this simply involves combining output BAM files for the individual FASTQ files into a single combined BAM file. The combined BAM file is then uploaded to HDFS for downstream processes.

Finally, to combine analysis steps performed using single-node software such as BWA and those implemented using MapReduce, we wrote a Java class that combines all five steps into a single pipeline. The full code was packaged into a JAR file invoked using a single command. From a usability perspective, users just need to execute a single command and the program takes care of chaining all the analysis steps.

Launching Hadoop-CNV-RF

AWS makes it possible for users to start a Hadoop cluster from a preconfigured Amazon Machine Image (AMI) containing all dependency software. We created an AMI containing all software dependencies needed by Hadoop-CNV-RF and made it publicly available. No installation of dependencies is needed to use Hadoop-CNV-RF. Users can simply point to this AMI when starting their Hadoop cluster.

To start the pipeline, users executes a secure shell (SSH) script which takes as input two configuration files. The first configuration file contains information about the Hadoop cluster, such as the number, type and size of compute nodes to use. The second configuration file contains credential information needed to access the cloud together with information needed by the pipeline, such as where to save the final results. The SSH script starts by copying FASTQ files to the cloud. It then launches a Hadoop cluster and monitors the cluster status every 60 seconds. Once the cluster is up and running, it starts the analysis, waits for the analysis to finish, downloads results from the cloud, and then shuts down the cluster. We provide this SSH script together with sample configuration files and detailed instructions on how to use the software.

Results

Running Time Analysis

We compared running times between the old (CNV-RF) and new (Hadoop-CNV-RF) implementations of our CNV detection algorithm. We analyzed five samples and the old pipeline took an average of 28.55 hours (standard deviation 4.22). The new pipeline took an average of 3.9 hours (standard deviation 0.98). The new implementation was not only an order of magnitude faster but also more robust. This analysis does not include extended analysis time due to failed runs which was common using the old pipeline. In practice, average runtimes for the old pipeline were much longer than 30 hours.

Clinical Validation

We validated Hadoop-CNV-RF to meet clinical standards. 23 samples were each analyzed twice using the pipeline and results examined to ensure CNVs on specific exons were detected both times. These samples included 20 samples with known CNV's (17 heterozygous deletions, 2 copy number gains and 1 complex CNV). These known CNV's ranged in size from single exon events to events involving multiple contiguous genes. In all 20 samples, these specific CNVs were detected on both analyses showing 100% concordance. Three additional samples with no CNV were analyzed to demonstrate specificity and all 3 samples yielded normal results. Duplicates were run to demonstrate concordance of calls across instruments (Nextseq and Novaseq) and flowcells (S1 and S2 flowcells on the Novaseq).

This pipeline is currently being used in the clinic and all potentially pathogenic CNVs identified by the pipeline were confirmed using quantitative polymerase chain reaction (qPCR). Since we started using the pipeline in the clinic, 13 calls have been identified, of which 8 validated by qPCR and were clinically reported, 2 were CNV's that were documented as common polymorphisms and were not validated by qPCR. The remaining 3 calls were considered false positives.

Discussion

We recently developed software to detect clinically relevant small CNVs using targeted re-sequencing data called CNV-RF. A major contribution of this software was its use of machine learning to reduce false positive rates. However, CNV-RF routinely failed due to memory limitation on a single computer. Additionally, it took on average over 30 hours to analyze large gene panels. In a clinical setting where it is not uncommon to analyze over 20 samples a week, this long running time was not practical and the existing pipeline could only analyze 10 to 15 samples in a routine work week. This resulted in delayed reporting of clinical results to clinicians and longer turnaround times that resulted in clinical samples being sent to other laboratories as we could not meet appropriate clinical expectations with the existing pipelines. To overcome this limitation, we re-implemented CNV-RF in a new software we call Hadoop-CNV-RF. Hadoop-CNV-RF takes as input FASTQ files and can rapidly scale to large gene panels. We are now able to analyze all 20 samples within 2 days. A major consideration when developing Hadoop-CNV-RF was to make it easy to use for clinicians. To that end, we created an Amazon Machine Image (AMI) with all the dependencies installed and step-by-step instructions on how to use it on AWS.

In addition to using our software on commercial cloud-computing infrastructure such as AWS, users have the option of using our software on traditional high-performance computing (HPC) resources if they have the resources to maintain one. It is also becoming common for research universities to maintain a HPC system for its researchers, and for researchers in the United States, a number of government agencies such as the National Science Foundation (NSF) maintain HPC systems such as XSEDE[10] that are available to researchers. For users with access to a supercomputer with a shared disk that uses the Portable Batch System (PBS) to distribute jobs, we provide instructions on setting up an ephemeral Hadoop cluster and launching Hadoop-CNV-RF. These instructions have been tested on an HP

Linux distributed cluster with 1800 total cores at the Minnesota Supercomputing Institute.

Finally, besides providing an accurate and reliable software for detecting clinically relevant CNVs, the new fast and user friendly software has significantly improved our clinical workflows. We no longer need to send samples to other labs due to the long running times of the old pipeline. The short turn-around times have resulted in greater satisfaction by testing clinicians.

Limitations

We have observed that on rare occasions (less than 5% of the time), Hadoop-CNV-RF fails to analyze genes on some of the chromosomes. We have yet to determine if this is an issue with our software or with the cloud provider infrastructure. To help users determine when this failure has occurred, Hadoop-CNV-RF provides a quality control file for identifying failures. We have not observed this failure when using Hadoop-CNV-RF on supercomputers at the Minnesota Supercomputing Institute.

Abbreviations

CNV: copy number variant
NGS: next-generation sequencing
HDFS: hadoop distributed file system
AMI: amazon machine image
SSH: secure shell
qPCR: quantitative polymerase chain reaction
HPC: high-performance computing
NSF: national science foundation
PBS: portable batch system

Acknowledgements

We would like to thank members of the Molecular Diagnostics Lab at M Health-Fairview clinic at the University of Minnesota for their help validating the pipeline. We thank the Scientific Computing Solutions group at the Minnesota Supercomputing Institute for support installing ephemeral Hadoop on their HPC cluster. Special thanks to Nick Dunn who put together instructions for installing ephemeral Hadoop on a HPC cluster. We thank Daniel Maseda for editorial help on the manuscript.

Author's contributions

GO conceived the research, implemented Hadoop-CNV-RF and wrote the paper. HCL integrated Hadoop-CNV-RF into the clinical pipeline, validated the software and analyzed samples using CNV-RF for comparative analysis. MB helped with clinical validation and is the main user of Hadoop-CNV-RF in the clinic. BT contributed to research ideas and helped write the paper. All authors read and approved the final manuscript.

Funding

GO was supported by Macalester College. HCL, MB and BT were supported by University of Minnesota.

Availability of data and materials

Source code is publicly available at: https://github.com/getiria-onsongo/hadoop-cnvrf-public.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Mathematics, Statistics, and Computer Science, Macalester College, Saint Paul, MN, United States. [2]Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, MN, United States. [3]Division of Genetics and Metabolism, University of Minnesota, Minneapolis, MN, United States. [4]Molecular Diagnostics Laboratory, M Health-Fairview, University of Minnesota, Minneapolis, MN, United States. [5]Department of Laboratory Medicine and Pathology, University of Minnesota, Minneapolis, MN, United States.

References

1. Roca, I., González-Castro, L., Fernández, H., Couce, M.L., Fernández-Marmiesse, A.: Free-access copy-number variant detection tools for targeted next-generation sequencing data. Mutation Research/Reviews in Mutation Research (2019)
2. Fromer, M., Purcell, S.M.: Using xhmm software to detect copy number variation in whole-exome sequencing data. Current protocols in human genetics **81**(1), 7–23 (2014)
3. McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., Garimella, K., Altshuler, D., Gabriel, S., Daly, M., *et al.*: The genome analysis toolkit: a mapreduce framework for analyzing next-generation dna sequencing data. Genome research **20**(9), 1297–1303 (2010)
4. Fowler, A., Mahamdallie, S., Ruark, E., Seal, S., Ramsay, E., Clarke, M., Uddin, I., Wylie, H., Strydom, A., Lunter, G., et al.: Accurate clinical detection of exon copy number variants in a targeted ngs panel using decon. Wellcome open research **1** (2016)
5. Onsongo, G., Baughn, L.B., Bower, M., Henzler, C., Schomaker, M., Silverstein, K.A., Thyagarajan, B.: Cnv-rf is a random forest–based copy number variation detection method using next-generation sequencing. The Journal of Molecular Diagnostics **18**(6), 872–881 (2016)
6. ORACLE: Java 8. https://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html. Accessed: 2020-01-23
7. Foundation, A.S.: Apache Hadoop. https://hadoop.apache.org/. Accessed: 2020-01-23
8. Amazon: Amazon Web Service. https://aws.amazon.com/. Accessed: 2020-01-23
9. Li, H.: Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. arXiv preprint arXiv:1303.3997 (2013)
10. Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G.D., *et al.*: Xsede: accelerating scientific discovery. Computing in science & engineering **16**(5), 62–74 (2014)

Figures

Figure 1 Example of splitting a FASTQ file into smaller files. This figure illustrates how to split a FASTQ file if using a three node cluster. The input file is split into three roughly equal smaller FASTQ files.