INTRODUCTION

Topic: Evaluating bioinformatics tools that can be run on Big Data-related platforms (Hadoop/MapReduce and/or Spark), specifically for the analysis of microbial genomic data.

Project Goals: We aimed to evaluate bioinformatics genomic-analysis tools that make use of Big Data Platforms; construct a pipeline that used such tools; and develop team members' knowledge and skills in using Big Data methods on bioinformatics data.

Summary: We tested and evaluated available bioinformatics tools that implement the Big Data platforms that were taught in the Spring 2016 Big Data Analytics course at NYU-Tandon. We first proposed a small genomic analysis pipeline on short-read genetic data from the Human Microbiome Project (HMP), then proceeded to install and test the tools. Based on the unsuitability of the HMP data for available tools, we then proposed a second, simpler pipeline using single-species data to use in testing the tools. We found that though there are several bioinformatics tools that have been created for use with Big Data technologies, many if not most of the tools are outdated and/or have not been kept up to date through developer maintenance or user engagement. In many cases, the tools seem to have been created as "proofs-of-concept," but are not used actively in the bioinformatics community, thus failing to receive updates or support. However, at least one promising tool, called ADAM, appears to receive frequent updates and have an active programmer community; additionally, it relies on the more user-friendly platform of Spark. Underscoring its promise, we were able to successfully produce output using ADAM, such as transforming frequently used bioinformatics files (FastQ, FASTA and BAM) to ADAM files.

Background

What is bioinformatics? From Wikipedia "Bioinformatics is an interdisciplinary field that develops methods and software tools for understanding biological data." It is described as the intersection of biology and computer science (Wikipedia Bioinformatics).

NGS Sequencing: The data used in this project comes from high-throughput gene-sequencing technologies. In brief, these high-throughput technologies consist of so-called "Next Generation Sequencing" (NGS) approaches that break up genetic material into hundreds of thousands or millions of shorter, "shotgun" stretches of DNA that are sequenced in high-throughput workflows and then assembled into complete genomes, either with the aid of a reference genome or via *de novo* genome assembly (Yegnasubramanian 2013).

Bioinformatics terms and concepts:

- Short-read genome sequencing: NGS techniques produce short reads from query samples of genetic material, of around 100 nucleotide base pairs (Yegnasubramanian 2013).

- *De novo* genome assembly: *De novo* genome assembly refers to the assembly of these short sequenced reads to form complete genome sequences. It is done without the use of a complete, reference genome (Illumina 2016).
- Reference genome alignment: Short reads are mapped to a known genome from the species for assembly (Hughes 2013).
- K-mer counting: K-mers are substrings of a genetic sequence of length “k.” This is an important step in genome assembly and quality control (Marcais 2011).
- Short-read quality control: Refers to checking for sequencing errors in short reads and correcting those errors or removing those fragments that have errors (Hughes 2013).
- Metagenome clustering: Metagenome refers to genetic data collected from the many species in a particular environment, for instance the human microbiome. Clustering of this data involves grouping the sequences by same or similar species (Rasheed 2013).
- File types
    - FASTA: Format used to store sequence record (no quality scores). Here it is the format used to store the reference genome (Beckman Coulter c2016).
    - FastQ: Format that includes both sequence and quality scores, used to store sequencing read data (Beckman Coulter c2016).
    - BAM: Binary format for sequence-alignment data (Integrative Genomics Viewer BAM).
    - VCF: Format for storing single-nucleotide polymorphisms (SNPs) and other structural genetic variants (Integrative Genomics Viewer VCF Files).

Big Data Bioinformatics: Several tools and algorithms have been developed to take advantage of the Hadoop/MapReduce Big-Data framework to perform common bioinformatics tasks. These include performing QC on short-read data (of the kind employed in shotgun sequencing), alignment of these reads to reference genomes, *de novo* assembly of these reads into genomes in the absence of a reference genome, annotating genes via BLAST, and deriving statistical characterizations of genomic data, such as GC content. These computation methods benefit greatly from parallel processing as the data sets are large and algorithms require intensive computational power.

METHODS AND MATERIALS

Project Team: Michael Dhar, Jade Wang, Ashley Yang

Computing environment:

- **Virtual Machine in Google Cloud Compute (Refer to Appendix I, included as a zip file, for login details):**
    - Configurations
        - 4 vCPUs & 15 GB of mem
        - Machine Type: Ubuntu 14.04 LTS
        - Zone: us-east1-d

        - IP forwarding off
        - External IP: ephemeral
    - Network
        - Opened ports for traffic flow by adding port number into firewall rules
        - Ports opened: 80, 8080, 8888, 7180, 50010, 50020, 50075, 8032, 8042, 18080, 8020, 8040
- **Docker:** Used Docker image for Cloudera installation of Hadoop version 2.6.0
    - Docker image pulled: cloudera/quickstart:latest
- **Google Cloud SDK:** Tool used for transferring files from local PC to VM (Google Cloud SDK).
- **Google Docs** was used to share and collaboratively edit project ideas, notes on tools and troubleshooting, code and commands, and final reports.

Data:

- ***Staphyloccocus aureus*:** Our updated pipeline aimed to work with read data (FastQ files) and reference genome data (Fasta file) for *Staphylococcus aureus*. The reference genome file was obtained from NCBI (Wellcome Trust Sanger Institute, ASM28453v1), and the FastQ files for methicillin-resistant *Staphylococcus aureus* (MRSA) reads were obtained from Sanger Institute (Wellcome Trust Sanger Institute, Staphylococcus aureus). The pipeline tools were tested on small subsets of the data (paired-end FastQ reads), but analyzing a full set of reads from several strains would represent a significant, Big Data burden. For example, there are 62 MRSA strains available on the Sanger Institute; if each paired-end strain is roughly 1GB in size, the entire data set is about 62GB. In addition, there are many more strains available on NCBI that we could add to our analysis.

.

- **Initial data attempt: Human microbiome:** Our initially proposed pipeline aimed to use publicly available data on the Human Microbiome website (HMPC 2012). The project has made both its assembled and raw, short-read data available. Our pipeline originally aimed to analyze the published Metagenomic Shotgun Sequencing (mwgs) raw data from 16 different body sites in healthy individuals, available at: http://hmpdacc.org/HMASM/. This pipeline approach was eventually abandoned in favor of the above.

Big Data Software: The Big Data platforms relevant to this project are:

- Hadoop/MapReduce: Hadoop is a framework for distributed processing of large datasets. MapReduce is the programming framework for Hadoop (ASF 2016 Feb 13).
- Pig Latin: This is a high-level programming language for the Hadoop framework (ASF 2015 June 6).
- Apache Spark: This is a cluster computing framework for use on large datasets. It can be used as an independent framework or on top of Hadoop (ASF, Spark Overview).

Bioinformatics Software: The project investigated several types of software that pertain to our analysis pipeline and that can be used in the Hadoop/MapReduce framework. To find available programs, we performed simple Google searches, read review papers and blog posts about bioinformatics software (for example, Tiwari 2012), and investigated relevant online discussion forums, such as Biostar and Stackoverflow. See "Testing and Troubleshooting" below for the results of our evaluations of these tools. Refer to Appendix II, at the end of this document, for links to software source code and project sites.

- **Quality Control:**
    - **Quake:** a quality-check program for raw FastQ reads described as being suitable for use with Hadoop (Kelley 2010).
- **Reference Genome Alignment:**
    - **CloudBurst:** a read-mapping tool that relies on MapReduce (Schatz 2009).
    - **Brisera:** an aligner that uses a Python implementation built on Spark (Bengfort 2015).
    - **BlastReduce:** a short-read aligner that uses MapReduce (Schatz 2008).
    - **Seal:** a toolkit that can run on the Hadoop framework. It contains tools for read alignment, demultiplexing, duplicate-removal, sorting and statistics for base quality recalibration (CRS4 c2011).
- ***De Novo* Genome Assembly**
    - **Contrail:** a genome assembler based on Hadoop (Schatz Contrail).
    - **CloudBrush:** a de novo assembler based on MapReduce (Chen 2013).
- **Additional Analysis:**
    - **SeqPig:** library for Apache Pig consisting of user-defined functions for processing unaligned and aligned reads (SeqPig 2013).
    - **MrMinCH**: a metagenome clustering algorithm that uses MapReduce via Pig scripts (Rasheed 2013).
    - **ADAM:** a genomics analysis platform designed for use on the Spark and other platforms. It can perform many types of statistical analysis (Massie 2013).
    - **Avocado:** a distributed pipeline for calling genetic variants, built on Spark and ADAM (Big Data Genomics 2014).

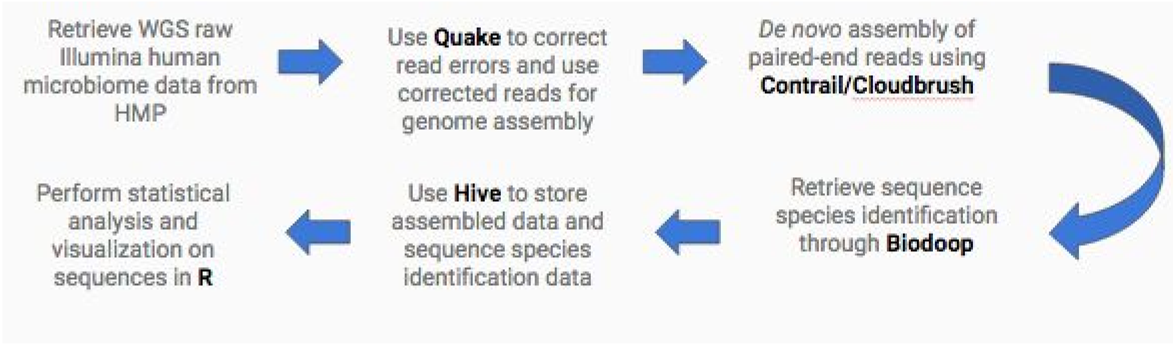
Pipelines:

The project proposed and attempted two pipelines. The initial pipeline was eventually abandoned when it was discovered that the core tools for *de novo* genome assembly were not fully developed or supported. The second pipeline was then attempted, with all tools tested and troubleshot thoroughly. (See "Tests and Troubleshooting" below.)

Initial Pipeline Attempt: This pipeline initially aimed to complete the following genomic analysis steps with the corresponding tools, using data from the Human Microbiome Project (see Fig. 1 for summary):

- Quality control (Quake)
- *De Novo* Genome Assembly (Contrail, CloudBrush)
- BLAST (not attempted; changed direction to 'second pipeline')
- Database (not attempted; changed direction to 'second pipeline')

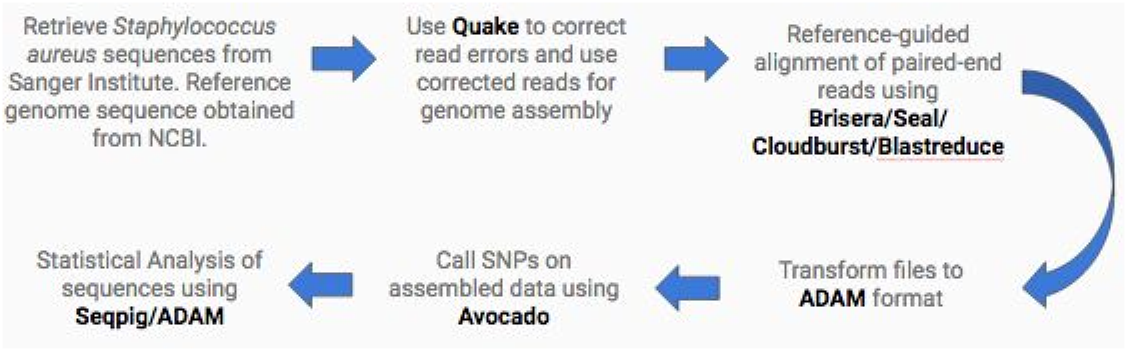
Fig. 1: Original Proposed Pipeline Summary


Retrieve WGS raw Illumina human microbiome data from HMP
Use **Quake** to correct read errors and use corrected reads for genome assembly
De novo assembly of paired-end reads using **Contrail/Cloudbrush**
Retrieve sequence species identification through **Biodoop**
Use **Hive** to store assembled data and sequence species identification data
Perform statistical analysis and visualization on sequences in **R**

Second Pipeline Attempt: This pipeline was proposed as a framework for testing tools; it attempted the following genomic analysis steps with the corresponding tools on MRSA *S.Aureus* data (see Fig 2. for summary):

- Quality control (Quake)
- Reference Genome Alignment (Brisera, CloudBurst, Seal)
- GC content statistics, variant calls, etc. (Seqpig, ADAM-Avocado)

Fig. 2: Revised Analysis Pipeline Summary


Retrieve *Staphylococcus aureus* sequences from Sanger Institute. Reference genome sequence obtained from NCBI.
Use **Quake** to correct read errors and use corrected reads for genome assembly
Reference-guided alignment of paired-end reads using **Brisera/Seal/ Cloudburst/Blastreduce**
Transform files to **ADAM** format
Call SNPs on assembled data using **Avocado**
Statistical Analysis of sequences using **Seqpig/ADAM**

RESULTS

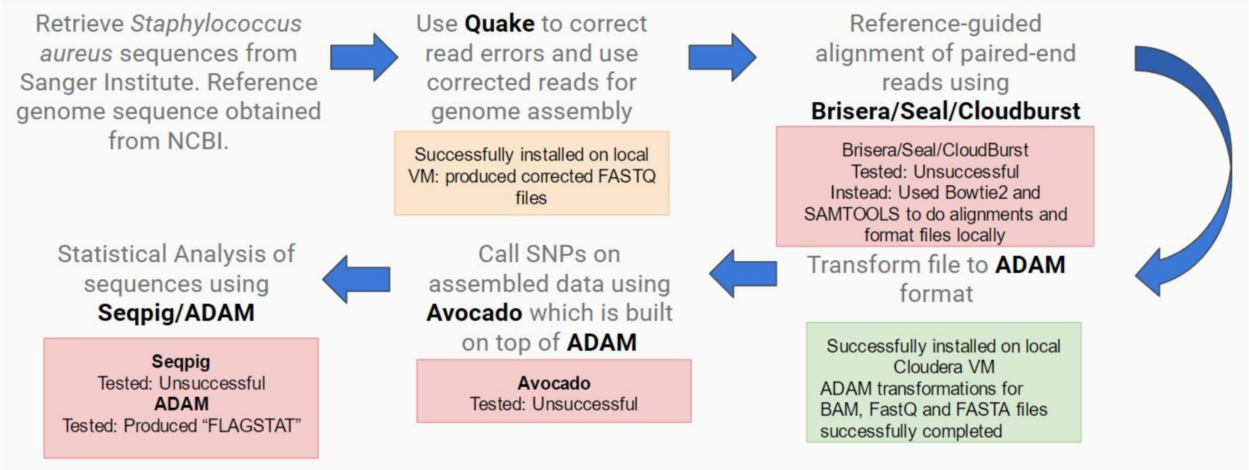
Initial Pipeline Result Summary: Testing and troubleshooting revealed unsuitability of available Hadoop-centered bioinformatics tools for the analysis of Human Microbiome data. Available raw reads were already pre-processed and no longer contained quality scores, making them unsuitable as Quake file input. No current *de novo* genome assemblers were available for sequence assembly. This pipeline was thus abandoned.

Second Pipeline Result Summary: We were able to successfully run Quake quality control locally, and streaming integration with Hadoop was attempted. See a sample view of the resulting quality-checked FastQs in Appendix III:SampleOutputs, included as a zip file. Attempts and troubleshooting were completed for reference genome aligners and additional analysis tools (see below). ADAM was successfully installed and run on both a local VM and the shared Google Cloud VM within a Cloudera environment, and an ADAM file was produced for use in further analysis. The reference genome and the aligned sequence were converted to ADAM files for further processing and analysis. See a sample view of the ADAM outputs in Appendix III:SampleOutputs; descriptions of the individual outputs for ADAM are below in the troubleshooting/testing section for ADAM. Commands and code used for all tools are available in Appendix I:Code-Commands, included as a zip file. Comments in the appendix code files indicate which commands were functioning.

TESTING AND TROUBLESHOOTING

We performed thorough troubleshooting and testing of the bioinformatics tools identified by the team as being potentially useful with the Big Data platforms. Below is a summary of our testing experience and results (Fig. 3), followed by detailed descriptions of the testing for each tool, along with our conclusion for that tool. (Refer to Appendix I:Code-Commands, included as a zip file, for the commands and scripts used for each program):

Fig. 3: Tool Analysis Summary:


Retrieve *Staphylococcus aureus* sequences from Sanger Institute. Reference genome sequence obtained from NCBI.
Use **Quake** to correct read errors and use corrected reads for genome assembly
Successfully installed on local VM: produced corrected FASTQ files
Reference-guided alignment of paired-end reads using **Brisera/Seal/Cloudburst**
Brisera/Seal/CloudBurst Tested: Unsuccessful Instead: Used Bowtie2 and SAMTOOLS to do alignments and format files locally
Transform file to **ADAM** format
Successfully installed on local Cloudera VM ADAM transformations for BAM, FastQ and FASTA files successfully completed
Call SNPs on assembled data using **Avocado** which is built on top of **ADAM**
Avocado Tested: Unsuccessful
Statistical Analysis of sequences using **Seqpig/ADAM**
Seqpig Tested: Unsuccessful **ADAM** Tested: Produced “FLAGSTAT”

Troubleshooting and Testing for Individual Tools:

- **Quality Control:**
  - **Quake:** The program and its dependencies (Jellyfish, Boost and R) were installed in the Cloudera environment. An issue with installing R was overcome by using yum install rather than package installation. The program was first tested running locally (without Hadoop integration) on the shared Google VM.

    Quake ran successfully on the *S. Aureus* data. However, the “--no_jelly” parameter had to be specified in order for the run to complete. If the parameter was not specified, Quake encountered a jellyfish error. We believe that this error is caused by the jellyfish program. Although the program was installed, when performing a test on jellyfish, 4 checks were skipped when only 1 should have been skipped (as indicated in the manual). The "--no_jelly" parameter forces Quake to use its own kmer counting tool, rather than that of jellyfish.

Since Quake ran successfully on the *S. Aureus* data through the command-line interface, we were able to conclude that the tool was installed correctly. Our next step was to run Quake in hadoop. There are two scripts in Quake that are appropriate for use through Hadoop streaming: count-qmers and reduce-qmers. The count-qmers script is the mapper and reduce-qmers is the reducer. The program's example bash script to run these two scripts was missing, so the developers were contacted. The developer Michael Schatz did now know what happened to the example Hadoop streaming bash script. Schatz felt that since the Hadoop streaming interface had changed a lot within the past 5 years since the script was written, the bash script would probably not work today. Another developer, David Kelley informed us that the Hadoop portion of the tool was not fully developed and was difficult to support.

Conclusion: Quake worked using the command-line interface, but some modifications from the user will be necessary in order for it to run in the Hadoop/MapReduce framework. Integrating with Hadoop would require knowledge of the C++ language and modification of the count-qmers.cpp file to read from standard input stream rather than from the input file. While the structure of the program suggests potential implementation with Hadoop via streaming, the code for doing so is no longer available. From the papers that we read, we were not aware of this and did not know that using Quake in Hadoop was more of a concept rather than a working model. Both Schatz and Kelley recommended a newer method and tool for error correction called Lighter. However, this tool was not developed for the MapReduce framework and could not be used in our pipeline.

The Quake tool was last updated in 2011.

- **Reference-Guided Genome Alignment:**
  - **CloudBurst:** This tool was installed and run on the shared Cloudera Hadoop environment. It was run using the developer's own sample command, with data coming from Human Microbiome fastQ reads, but no job initiated. This tool has not been updated since 2010, and was designed for Hadoop 0.20.X (most recent Hadoop is 2.7.2, and our Cloudera installation uses Hadoop 2.6.0).

    Conclusion: Tool is likely outdated and not suitable for current Hadoop use.

○ **Brisera:** Brisera was installed on the shared Cloudera Hadoop environment, to be run via Spark. The developer's suggested commands were attempted using *S. Aureus* data. The command for preparing the FASTA file for use produced an error that the temp/out file could not be located. The team contacted the program's developer, who suggested a fix to the code (force the program to create a tmp/out file within the Python script instead of relying on Spark to do so).

  While this fixed that specific error, another arose from the Python code. Error messages stated that the partition file could not be found in the tmp/out file. Permissions were then checked in the tmp output file to confirm that this was accessible to the program, and thus the error was not a permission issue. Then the convert_fasta.py script that was run to convert the Fasta file to a binary form for Spark was analyzed. It looked like the partition files were not being created. The lines in the script that created the partition files called on a class from the Python module 'convert.py' to convert the fasta file. This conversion is then copied to form a distributed dataset that can be parallelized in spark. After this, the parallelized data is saved into the temp/out folder as partition files. Since the parallelized partition files did not exist and were not being created, it is likely that the 'convert.py' script was the issue and not the 'convert_fasta.py' script. It was decided that too many bugs likely existed within the Python code.

  This program's GitHub page does not list any comments from users (save the ones this team posted during this project), and so there is likely little user community engagement and little maintenance from the developer.

  Conclusion: This tool seems to lack proper support and developer debugging of the Python code for reliable use.

○ **SEAL:** Seal was registered as a project on sourceforge.net in 2011 and was last updated in 2014. Initial installation using pip install failed due to a compilation error. As an alternative, Seal and its dependencies were installed individually. Dependencies installed were: Pydoop, Protobuf, HadoopBAM, Ant and JUnit. HadoopBAM path was set in the environment variable (HADOOP_BAM) to point to the root HadoopBam directory. This directory tells Seal where the HadoopBam jars are.

  Building seal was still unsuccessful. The error produced seems to indicate that pydoop (a python mapreduce and hdfs api for hadoop) could not be found. A Google search did not return any postings in forums regarding this error that led to an unsuccessful Seal build.

  Conclusion: The Seal build was unsuccessful, and thus installation failed.

    - **BlastReduce:** This program was described no later than 2008, and the site that is listed as hosting the source code no longer contains a BlastReduce section. This tool was superseded by CloudBurst.

      Conclusion: This tool is no longer available.

- ***De Novo* Genome Assembly:**
    - **Contrail:** The program was installed in the shared Cloudera Hadoop environment. Once the program is installed, the user must compile the source Java code into a jar for the program to be run in Hadoop. However, the jar compile failed on both new and older versions of Hadoop. We decided to test it on the older Hadoop version 0.20.0, because the tool was developed based on that version. It was tested both on the team's current Hadoop installation and on a second Google Cloud VM on which Hadoop 0.20.0 was installed. (This was done on the base VM, not via Docker, for which there is no container for the Hadoop 0.20.0 version.) Another, simple MapReduce program for Kmer counter written by the same authors who described Contrail was also attempted. It similarly failed to compile. Contrail has not been updated in four years, and was written for Hadoop 0.20.0.

      Conclusion: The compile errors referred to failures to recognize symbols for classes and the use of deprecated classes, even on Hadoop 0.20.0, for which the tool was developed. This suggests possible incompatibility with current Java installations. The program has little support and very little information on installation or instructions on usage. We could only find an old blog (Homolog_US 2011 Sept 8), which was not written by the developers, that listed usage instructions and the source-code link. Contrail does not appear suitable for current use.

    - **CloudBrush:** The program was installed and run on the Cloudera Hadoop environment, using both the developer's sample data and data from this project's initial pipeline (Human Microbiome reads). The program failed on both sets of data. It did initialize, and completed four of its internal MapReduce jobs, but failed on the fifth step, VerifyOverlap, giving the following error: "Comparison method violates its general contract!"

      Online commentary on StackOverflow suggests this problem is due to errors in the original Java code. Specifically, the "OverlapSizeComparator" class uses a defective comparison calculation (Krumwiede 2015). The team attempted to rewrite and recompile the code in Eclipse with this error corrected, but the program would still not run with the newly compiled code. The developer (https://github.com/ice91) was contacted about the issue, but no response was received.

Conclusion: This program has some promise, as it did initialize and complete several of its MapReduce jobs. However, it has internal coding errors and so does not appear to be well-maintained. The source code has not been updated in three years.

  - ***De Novo* Assembly Tools Summary:** The above two tools were the only *de novo* assemblers available for use with Hadoop discovered by the team. This recent (November 2015) biology Stackexchange discussion thread suggests that those remain the only two available (Selvam 2015).

- **Additional Analysis Tools:**
  - **ADAM:** ADAM interfaces with several platforms; here, it was attempted on Spark. ADAM looks promising for use in bioinformatics, as it has a healthy update history, with new releases coming every few months since the project's debut in early 2014. The latest release (0.19.0) arrived in Feburary 2016 (http://bdgenomics.org/). Furthermore, the program is designed for use with Spark, a much more user-friendly platform compared to Hadoop/MapReduce Java scripting or Pig Latin. However, this team encountered immediate problems with installing ADAM in the Cloudera Hadoop environment. ADAM installation requires Maven 3.1, which in turn requires Java JDK 1.7. The Cloudera environment operates on Java JDK 1.6. Java 1.7 was installed on the Cloudera Docker container with Hadoop, but the Cloudera environment failed to recognize the new Java, and Hadoop failed to work with it.

    As an alternative, ADAM installation was completed on one of the team member's local VMs in a Cloudera Hadoop environment using Java JDK 1.7. Installation was successful, and ADAM's reference genome alignment command was attempted on MRSA data. ADAM was successfully run and and an ADAM file produced for potential use in further analysis.

    ADAM was later installed and run on the Docker Cloudera-Hadoop environment in the Google Cloud VM after a method for updating the JAVA version on the system was found.

    After obtaining output files (FASTQ files) from Quake, generally the data needs to be aligned against a reference genome before being assembled. Unfortunately, the sequence-alignment tools available were not functional in our VMs, so non-Hadoop-based bioinformatics tools were used locally in order to continue the pipeline. Bowtie2 (Langmead 2012) was used to align the *S. aureus* sequence against the reference genome, and SAMtools (Heng 2009) was used to convert

Bowtie2 output to BAM format, to reduce memory usage. Picard tools (Broad Institute Picard) were also used on the reference genome to create a dictionary.

After copying the BAM file and dictionary into HDFS, we were able to transform the *S. aureus* BAM file into ADAM format. It should then be possible to calculate depth of coverage, by converting the reference genome into ADAM, creating a VCF file using a reference dictionary created by Picard-tools, and finally using that VCF file with the *S. aureus* ADAM files. However, the adam2vcf function currently throws an error: “BCF not yet supported.” The source code row 77 indicates that ADAM currently is not able to save a file as “VCF” (Heur 2016). As a result, depth of coverage could not be calculated using ADAM.

On the other hand, the count_kmers command ran successfully. It does ask for the K-mer length, which can be estimated using kmergenie (Chikhi 2013), a non-Hadoop-based tool, on a local machine.

ADAM can produce a variety of important results, such as the ADAM output, read statistics, and tag counts. The ADAM file can also be flattened, creating a file that is suitable for querying with tools such as Impala and to be used in further analysis by downstream Hadoop-based tools such as Avocado. However, Avocado was found not to work with the files produced by ADAM; error messages referenced a missing required Avro field.

See Appendix_III:_SampleOutput (included as a zip file) in the ADAM directory for samples of all the outputs. Some output files are truncated after 10 rows, because output files can be very large.

- Transforming FASTQ to ADAM file output: **Transform:FASTQ-ADAM** (top 10 row of data from output)
- Transforming BAM to ADAM file output: **Transform:BAM-ADAM** (top 10 rows of data from output)
- Converting FASTA to ADAM file output: **FASTA2ADAM:FASTA-ADAM** (top 10 rows of data from output)
- Printing Flagstats (read statistics): **ERR064898.flagstat** (entire output)
- Print Tags and Counts: **ERR064898.tags** (entire output)
- Print Sequence Dictionary: **ERR064898.dic** (entire output)
- Flattened ADAM file output: **reference.table** (entire output)

Conclusion: As noted above, ADAM holds much more promise than many of the other programs we tested, due to its new and frequent releases and because it is built on Spark. However, it presented installation problems on Google Cloud’s Cloudera VM, indicative that recent bioinformatics Hadoop/Spark programs require environments with up-to-date dependencies. However, the program was successfully run in a Cloudera environment, on a local VM and Google VM, underscoring its promise for bioinformatics professionals.

○ **SeqPig:** This program held promise as a useful tool in bioinformatics analysis. It offers a number of simple sample scripts to run on common bioinformatics data file types (BAM, FastQ, etc.). For instance, it offers a simple, five-line Pig Latin script that, with SeqPig running on top of the Pig shell, should produce statistical results concerning GC content in FastQ files for genomic reads (http://seqpig.sourceforge.net/#x1-320004). In addition, the team members for this course project have experience using Pig due to the class's previous assignments.

  However, following Seqpig installation, there were immediate issues, with the program failing to find files it needed to run. For Seqpig to work, environment variables need to be set for the home directories of Java, Hadoop, Pig and Seqpig. The team sought out online instructions for finding these paths. Some online sources suggest the Cloudera Hadoop installation has unique settings (Knicely 2013). A number of different JAVA_HOME settings were attempted, including /user/jdk1.7.0_67-cloudera and every directory within /usr/lib/jvm. The directory /usr/lib/jvm/jre-1.7.0-openjdk.x86_64 was seemingly recognized by Hadoop. Seqpig was attempted with this set as JAVA_HOME, in conjunction with HADOOP_HOME as /usr/lib/hadoop-mapreduce. Errors of the type "/usr/lib/pig/contrib: No such file or directory" remained. There are no alternative Pig folders within /usr/lib, and the directory /usr/lib/contrib does not exist, suggesting incompatibility between Seqpig and current editions of Hadoop and Pig.

  Perhaps in explanation of these errors, SeqPig is a somewhat old program. Though it is newer than many of the programs we tested, it was last described in 2013 and was tested only on Hadoop versions 0.20.2 and 1.0.4.

  Conclusion: Seqpig holds promise for utility and ease-of-use for bioinformatics professionals, but sensitivity to different environment setups, and a lack of releases for newer Hadoop versions, seemed to limit its usefulness to us. It is possible current Pig editions do not offer some dependencies Seqpig needs.

○ **MrMinCH:** The team noticed that the developers had listed an erroneous command for running a Pig script within the Pig "Grunt" interactive shell ("pig" had been used instead of "run," and "-param" was erroneously placed in front of the Pig script name). However, after correcting that error, the scripts still failed to run, due to reported errors in class naming. Changing the name of the troublesome class did not correct the error.

Conclusion: The errors discovered in the basic command to run a Pig script, combined with the persistent errors reported in trial runs, suggest that this program's code may not be well-maintained.

DISCUSSION

This course project aimed to implement the Hadoop/MapReduce framework, and the Spark framework, in bioinformatics for a few reasons. First, as all team members are bioinformatics MS students at NYU-Tandon, and this course focused much of its attention on Hadoop/MapReduce, this seemed the natural setting for our course project. Moreover, these Big Data frameworks hold promise for tackling current bioinformatics issues. Biological studies are generating sequencing data at a rate faster than the ability to analyze it (Pollack 2011). Meanwhile, certain Bioinformatic tools, such as the ABySS assembler, can run for days before the assembly of a genome is finished (based on team member's experience). We wanted to explore the MapReduce capability in Hadoop to expedite sequence analysis rate (if possible).

We found, however, that many tools built to facilitate such a marriage of Hadoop and bioinformatics were outdated and lacked current updates. Many of these tools seem to have been created more as "proof-of-concept" projects, but were then not adopted by the bioinformatics community. This is evidenced by the lack of user comments or requests for updates on the source-code pages for many of these programs. Several online commentaries on Hadoop/MapReduce in the bioinformatics setting echo those observations. In a Biostars post questioning why Hadoop was not used more frequently in bioinformatics, bioinformaticist Jeremy Leipzig wrote that "most of Hadoop for bioinformatics papers are proofs of concept, and real-world use of Hadoop in bioinformatics is quite low." Leipzig suggested that this is the case because "Hadoop combines two awesome bottlenecks to bring bioinformatics software to its knees -- using the network to disperse data and then relying on disk IO to access it" (Leipzig 2014).

Similarly, a blog post on DNA Nexus noted that this DNA data-analysis company abandoned Hadoop/MapReduce for a genomics platform it was building (Csordas 2014). The reasons given included that most bioinformatics tools are not written for Hadoop/MapReduce and that the platform would require moving data from place to place. A data-science blog post argued that the bioinformatics community has failed to adopt MapReduce because the platform lends itself more to transactional, retail data and because bioinformatics users usually have access to university supercomputer clusters, which are used instead (Huss 2013).

However, the ADAM project for bioinformatics analysis in the Spark setting provides reason for optimism. As we found, it has undergone frequent updates. In that same Biostar post, a moderator noted that "Most of the applications you mentioned can be and have already been implemented on top of Hadoop. A good example is the ADAM format, a Hadoop-friendly replacement of BAM, and its associated tools. They are under active development by professional programmers" (LH3 2014).

Since ADAM is built on top of Spark, it also presents a much more user-friendly interface, particularly for bioinformatics professionals, who may not have the significant Java or Pig Latin programming experience necessary to use older bioinformatics implementations of Hadoop.

Conclusion

Though the challenges in implementing Hadoop-related bioinformatics tools are noted above, such integration still holds promise. For example, though many projects are outdated and were written as proofs of concept, those proofs of concept do show that Hadoop could be used in bioinformatics -- with the appropriate updates and user engagement. In other words, it worked at one time; it was simply then abandoned.

Fortunately, ADAM seems to be succeeding on those counts, with frequent updates and an active stable of programmers. However, since the software is constantly being updated and further developed, there is a lack of current documentation. Our success in producing results from ADAM further demonstrates that it is an improvement upon many of the older, more outdated Hadoop-based bioinformatics tools we tested in this project. It may therefore be useful in helping to improve the rate of sequence analysis in bioinformatics, as mentioned as a goal in the Discussion. Data science blogger Csordas echoed that goal of ours in writing, "As far as I know, Hadoop is not used in production in bioinformatics anywhere. But the benefits of re-implementing the crucial bioinformatics algorithms in the Hadoop ecosystem would make it a very worthy challenge and would connect the biodata world back to the world of general/web data" (Csordas 2014).

References

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol, 215 (1990). 403-410. http://www.ncbi.nlm.nih.gov/pubmed/2231712.

Apache Software Foundation (ASF). 2016 Feb 13. Welcome to Apache Hadoop! [documentation]. Hadoop. http://hadoop.apache.org/.

Apache Software Foundation (ASF). 2015 June 6. Welcome to Apache Pig! [documentation]. Hadoop. https://pig.apache.org/.

Apache Software Foundation (ASF). Spark Overview [documentation]. Apache.org. http://spark.apache.org/docs/latest/.

Beckman Coulter. c2016. Understanding Sequencing Data File Formats [definitions page]. Beckman Coulter. http://www.beckmangenomics.com/genomic_services/bioinformatics/understanding_sequencing_data_file_formats.html.

Bengfort B. 2015 June 4. Brisera: A Spark implementation of a distributed seed and reduce algorithm [source code]. GitHub. https://github.com/bbengfort/brisera.

Big Data Genomics. 2014 May 4. Avocado [project site]. BDGenomics.org. http://bdgenomics.org/projects/avocado/.

Chen C. 2013 July 29. CloudBrush: A De Novo Next Generation Sequence Assembler Based on String Graph and MapReduce Cloud Computing Framework [source code]. GitHub. https://github.com/ice91/CloudBrush.

Chikhi R, Medvedev P. 2013 April 20. Informed and Automated k-Mer Size Selection for Genome Assembly. Oxford University Press. http://arxiv.org/pdf/1304.5665.pdf

CRS4. c2011. Biodoop-Seal. Sourceforge. http://biodoop-seal.sourceforge.net/.

Csordas A. 2014 Jan 3. Google invests into DNANexus: aging-driven big data without the Hadoop ecosystem? Pimm [blog]. https://pimm.wordpress.com/2014/01/03/google-invests-into-dnanexus-aging-driven-big-data-bioinformatics-without-mapreduce-hadoop/.

Google Cloud SDK [documentation]. Google Cloud Platform. https://cloud.google.com/sdk/docs/.

Heng L, Handsaker B, Wysoker A, et al. 2009 May 30. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 25(16):2078-2079. http://bioinformatics.oxfordjournals.org/content/25/16/2078.long.

Heur M. 2016. VariationRDDFunctions.scala [source code]. GitHub. https://github.com/bigdatagenomics/adam/blob/master/adam-core/src/main/scala/org/bdgenomics/adam/rdd/variation/VariationRDDFunctions.scala.

Homolog_US. 2011 Sept 8. Contrail--A De Bruijn Genome Assembler That Uses Hadoop [blog]. Homolog.us-Bioinformatics. http://www.homolog.us/blogs/blog/2011/09/08/contrail-a-de-bruijn-genome-assembler-that-uses-hadoop/.

Homolog_US. 2011 Aug 31. Using Hadoop for Transcriptomics -- An Example to Get Started [blog]. Homolog.us-Bioinformatics. http://www.homolog.us/blogs/blog/2011/08/31/using-hadoop-for-transcriptomics-an-example-to-get-started/.

Hughes J. 2013. Tutorial: Reference Assembly - Mapping Reads to a Reference Genome [forum tutorial]. Biostars. https://www.biostars.org/p/75489/.

Human Microbiome Project Consortium (HMPC). 2012. A framework for human microbiome research. Nature. 486: 215-221. http://www.nature.com/nature/journal/v486/n7402/full/nature11209.html.

Human Microbiome Project Consortium (HMPC). HMIWGS/HMASM - Illumina WGS Reads and Assemblies [data site]. Human Microbiome Project: http://hmpdacc.org/HMASM/.

Human Microbiome Project Consortium (HMPC). 2012. Structure, function and diversity of the healthy human microbiome. Nature. 486:207–214. http://www.nature.com/nature/journal/v486/n7402/full/nature11234.html.

Huss M, Westerberg J. 2013 Dec 26. Hadoop (and other parallel computing framework) solutions for genomics and proteomics [blog]. Follow the Data. Sordahttps://followthedata.wordpress.com/2013/12/26/hadoop-and-other-parallel-computing-framework-solutions-for-genomics-proteomics/.

Illumina. 2016. Introduction to De Novo Sequencing [method description]. Illumina. http://www.illumina.com/techniques/sequencing/dna-sequencing/whole-genome-sequencing/de-novo-sequencing.html.

Integrative Genomics Viewer. 2013. BAM [file format definition]. Broad Institute. https://www.broadinstitute.org/igv/BAM.

Integrative Genomics Viewer. 2013. VCF Files [file format definition]. Broad Institute. https://www.broadinstitute.org/igv/viewing_vcf_files.

Kelley DR, Salzberg SL. 2010 Nov 2. Clustering metagenomic sequences with interpolated Markov models. BMC Bioinformatics. 11:544. http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-544.

Kelley DR, Schatz MC, Salzberg SL. 2010. Quake: quality-aware detection and correction of sequencing errors. Genome Biology. 11:R116. https://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-11-r116.

Knicely J. 2013 July 2. Answer: Where is Hadoop home directory [forum post]. Vertica Forums. http://vertica-forums.com/viewtopic.php?t=1153.

Krumwiede K. 2015 July 31. Answer to Error: java.lang.IllegalArgumentException: Comparison method violates its general contract even using workaround [discussion thread]. StackOverflow. http://stackoverflow.com/questions/31375801/error-java-lang-illegalargumentexception-comparison-method-violates-its-genera.

Langmead B, Salzberg SL. 2012 March 4. Fast gapped-read alignment with Bowtie 2. Nature Methods. 9:357-359. http://www.nature.com/nmeth/journal/v9/n4/full/nmeth.1923.html.

Leipzig J. 2014. Answer: Why is Hadoop not used a lot in bioinformatics? [forum answer]. Biostars. https://www.biostars.org/p/115260/.

LH3. 2014. Answer: Why is Hadoop not used a lot in bioinformatics? [forum answer]. Biostars. https://www.biostars.org/p/115260/.

Marcais G, Kingsford C. 2011 July 26. JELLYFISH - Fast, Parallel K-mer Counting for DNA [software description]. University of Maryland Center for Bioinformatics and Computational Biology. http://www.cbcb.umd.edu/software/jellyfish/.

Massie M, Nothaft F, Hartl C, et al. 2013 Dec. 15. ADAM: Genomics Formats and Processing Patterns for Cloud Scale Computing. University of California, Berkeley, Technical Report N. UCB/EECS-2013-207. http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-207.html.

Pollack, A. 2011 Nov 30. DNA Sequencing Caught in Deluge of Data. The New York Times (New York Ed.). Sect. B:1.

Broad Institute. Picard [documentation]. GitHub. https://broadinstitute.github.io/picard/command-line-overview.html.

Rasheed Z, Rangwala H. 2013. MrMC-MinH: MapReduce-Based Metagenome Clustering Using Minwise Hashing [program site]. George Mason University Computer Science. http://cs.gmu.edu/~mlbio/MrMC-MinH/.

Schatz MC. 2008. BlastReduce: High-Performance Short-Read Mapping with MapReduce [PDF]. University of Maryland Computer Science. http://www.cs.umd.edu/grad/scholarlypapers/papers/MichaelSchatz.pdf.

Schatz MC. 2009 April 3. CloudBurst: highly sensitive read mapping with MapReduce. Bioinformatics. 25(11):1363-1369. http://bioinformatics.oxfordjournals.org/content/25/11/1363.

Schatz M, Sommer D, Kelley D, Pop M. Contrail: Assembly of Large Genomes using Cloud Computing. http://contrail-bio.sf.net/.

Selvam S. 2015 Nov 2. Answer: Are there any de novo assembly programs for Hadoop? [forum post]. Biology: StackExchange. http://biology.stackexchange.com/questions/30449/are-there-any-de-novo-genome-assembly-programs-for-hadoop.

SeqPig. 2013 Sept 28. SeqPig Manual [project description]. Sourceforge. http://seqpig.sourceforge.net.

Tiwari A. 10 Aug 2012. MapReduce and Hadoop Algorithms in Bioinformatics Papers [blog]. Abhishek Tiwari: Marketing, Technology and Architecture. http://abhishek-tiwari.com/post/mapreduce-and-hadoop-algorithms-in-bioinformatics-papers.

Wellcome Trust Sanger Institute. ASM28453v1 [genome assembly data]. NCBI. (http://www.ncbi.nlm.nih.gov/assembly/GCA_000284535.1.

Wellcome Trust Sanger Institute. Staphylococcus aureus [sequence data site]. Sanger.ac.uk. http://www.sanger.ac.uk/resources/downloads/bacteria/staphylococcus-aureus.html.

Wikipedia. Bioinformatics. https://en.wikipedia.org/wiki/Bioinformatics.

Yegnasubramanian S. 2013. Explanatory Chapter: Next-Generation Sequencing. Methods Enzymol. 529:201-208. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3978176/.

Appendix II: Software Links

Google Cloud SDK:

- Site: https://cloud.google.com/sdk/
- Documentation: https://cloud.google.com/docs/

QC:

Quake:

- Paper: http://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-11-r116
- Code Source: http://www.cbcb.umd.edu/software/quake/

Reference Genome Alignment:

CloudBurst:

- Paper: http://bioinformatics.oxfordjournals.org/content/25/11/1363
- Code Source: https://sourceforge.net/projects/cloudburst-bio/

Brisera:

- Code Source: https://github.com/bbengfort/brisera

BlastReduce:

- Paper: http://www.cs.umd.edu/grad/scholarlypapers/papers/MichaelSchatz.pdf
- Code Source: http://www.cbcb.umd.edu/software/blastreduce/ [no longer posted]

Seal:

- Code Source: https://sourceforge.net/projects/biodoop-seal/?source=recommended
- Program Site: http://biodoop-seal.sourceforge.net/

De Novo Genome Assembly:

Contrail

- Description: http://www.homolog.us/blogs/blog/2011/09/08/contrail-a-de-bruijn-genome-assembler-that-uses-hadoop/
- Kmer script description: http://www.homolog.us/blogs/blog/2011/08/31/using-hadoop-for-transcriptomics-an-example-to-get-started/
- Code Source: http://contrail-bio.svn.sourceforge.net/

CloudBrush:

- Code Source: https://github.com/ice91/CloudBrush
- Commentary: http://stackoverflow.com/questions/31375801/error-java-lang-illegalargumentexception-comparison-method-violates-its-genera)

Additional Analysis:

MrMC-MinH:

- Code Source: http://cs.gmu.edu/~mlbio/MrMC-MinH/

SeqPig:

- Program Description and Code: http://seqpig.sourceforge.net

ADAM:

- Project Site: http://bdgenomics.org/
- Tech Report: http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-207.html
- Code Source and Manual: https://github.com/bigdatagenomics/adam

Avocado:

- Project Site: http://bdgenomics.org/projects/avocado/
- Code Source: https://github.com/bigdatagenomics/avocado

Data links:

- Staph reference file obtained from NCBI (http://www.ncbi.nlm.nih.gov/assembly/GCA_000284535.1)
- FastQ files for methicillin-resistant *Staphylococcus aureus* (MRSA) reads obtained from Sanger Institute (http://www.sanger.ac.uk/resources/downloads/bacteria/staphylococcus-aureus.html).
- Human Microbiome Data: http://hmpdacc.org/HMASM/.

Big Data platforms:

Hadoop/MapReduce:
http://hadoop.apache.org/
https://www-01.ibm.com/software/data/infosphere/hadoop/mapreduce/

Pig Latin:
https://pig.apache.org/docs/r0.9.1/basic.html

Apache Spark: http://spark.apache.org/