Massive Genomic Data Processing and Deep Analysis

Abhishek Roy[†], Yanlei Diao[†],
[†]Department of Computer Science
University of Massachusetts, Amherst
{aroy,yanlei}@cs.umass.edu

Evan Mauceli[‡], Yiping Shen[‡], Bai-Lin Wu[‡]
[‡]Department of Laboratory Medicine
Harvard Medical School & Children's Hospital Boston
{evan.mauceli,yiping.shen,bai-lin.wu}@childrens.harvard.edu

ABSTRACT

Today large sequencing centers are producing genomic data at the rate of 10 terabytes a day and require complicated processing to transform massive amounts of noisy raw data into biological information. To address these needs, we develop a system for end-to-end processing of genomic data, including alignment of short read sequences, variation discovery, and deep analysis. We also employ a range of quality control mechanisms to improve data quality and parallel processing techniques for performance. In the demo, we will use real genomic data to show details of data transformation through the workflow, the usefulness of end results (ready for use as testable hypotheses), the effects of our quality control mechanisms and improved algorithms, and finally performance improvement.

1. INTRODUCTION

Genomics has revolutionized almost every aspect of life sciences in the past decade. At the same time, technological advancement such as next-generation sequencing is transforming the field of genomics into a new paradigm of data-intensive computing [1]. A large sequencing center such as the Broad Institute of Harvard and MIT can produce 10 terabytes of genomic data each day. The flood of data needs to undergo complex processing that mines biological information from vast sets of small sequence reads while handling numerous errors inherent in the data. At present, the processing of a single person's genomic data takes 10-12 days of machine time at state-of-the-art sequencing centers.

The genomic data characteristics and complex data processing needs have severe implications on real-world deployments. First and foremost, data quality is of paramount concern to diagnostic labs such as the Genetic Diagnostic Laboratory at Children's Hospital Boston (GDL-CHB). In the current practice, GDL-CHB sends de-identified DNA samples to a certified commercial sequencing company. The company then delivers the results (about 100-200GB) containing both short read sequences and detected genomic variants for each sample, with a turn-around-time of two months. Currently, a major hurdle is to detect true genomic variations due to a high rate of false positives (genomic variations reported by data processing software but invalidated by laboratory work) in the processed data. Alternatively, diagnostic labs, such as GDL-CHB, have to process raw genomic data themselves, involving developing software to align the genome, detect variants, and assess data quality.

Second, even large research institutes that have the ability to fully process genomic data feel a pressing need to address daunting performance and scalability challenges. In particular, the major challenges are to significantly increase the amount of data processed each day while reducing the latency in processing an urgent DNA sample (e.g., to reduce the delay of 12 days for finding a treatment strategy for a cancer patient or an acute infectious disease).

To address the above challenges, we design and develop a workflow system for massive genomic data processing and deep analysis. Our system has the following key features:

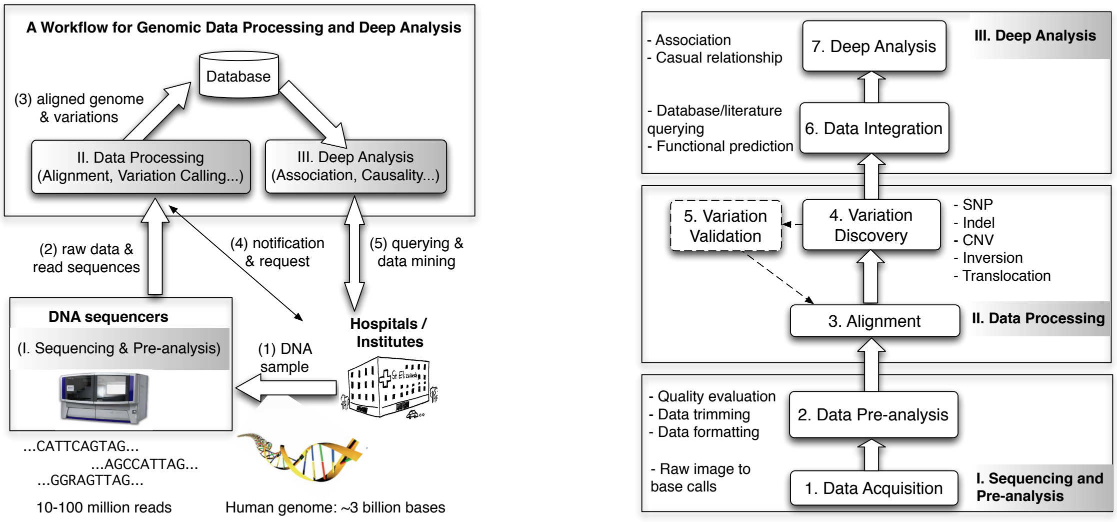
1. End-to-End Processing: Our system provides end-to-end processing of genomic data, including (*i*) alignment of read sequences of a sample against a reference genome, (*ii*) variation discovery based on an aligned genome and the reference genome, and (*iii*) deep analysis based on patient information and detected genomic variations, such as finding associations of genomic variants and patient phenotypes. The scope of processing in our system stands in contrast with existing systems that focus only on a particular task. For instance, well-known systems such as BWA [8], Bowtie [7], and Wham [10] are designed for alignment only. The Genome Analysis Toolkit (GATK) [4] focuses on local re-alignment (to improve alignment quality) and simple variation detection. Our system provides a much deeper processing pipeline and eventually outputs patterns of statistical significance, such as the association of genomic variants and patient phenotypes, which can be used as testable hypothesis for immediate validation via lab work.

2. Error Tracking and Diagnosis: Our system is designed to have data quality as a first-class concept, across input data, immediate results from alignment and variation discovery, and the final output of deep analysis. In particular, we extend the notion of "quality scores" beyond that only for read sequences in existing systems, and devise new ways to compute quality scores for all intermediate and final results. We further have a range of quality control mechanisms that prune low quality data (due to *data issues*) and reject poor alignment and variation detection results (due to *software issues*).

3. Scalable Data Processing: Our workflow further explores parallel processing to distribute massive input data sets and intermediate data sets to multiple nodes. Our profiling results show that the most expensive operations in the workflow include alignment, quality score recalibration, and variation discovery. Our techniques focus on these operations to improve overall system performance.

As part of a bigger initiative at Children's Hospital Boston, our workflow is an important initial step towards genetic diagnosis and treatment of patients and ultimately the vision of "personalized medicine." By returning patterns of statistical significance that can be used as testable hypotheses, our system can dramatically reduce the time and human costs of today's labor-intensive screening for biological hypotheses. Our system will also help track and mitigate errors in processed data, which have long plagued genome processing systems. Finally, the improved performance of our system will be important for analyzing urgent DNA samples to find treatment strategies for cancers and acute infectious diseases.

Proceedings of the VLDB Endowment, Vol. 5, No. 10



(a) Overall system architecture.

(b) Workflow for data processing and deep analysis.

Figure 1: A workflow for genomic data processing and deep analysis.

In our demonstration, we will present a working prototype system using real genomic data sets and processing workloads. We will run the workflow from alignment, to variation discovery, and finally to association mining to find associations between genomic variations and phenotypes. We will compare our association mining algorithm against the state-of-the-art [5] to demonstrate the changes needed for genomic data analysis. We will also compare our system with a baseline implementation using existing software tools to show the effectiveness of our quality control mechanisms. Finally, we will perform parallel processing of expensive operations such as alignment and variation discovery, and show the improved performance.

2. SYSTEM OVERVIEW

Our system for genome data processing and analysis works in an environment shown in Figure 1(a). Users of the system are hospitals and research institutes. A user starts by sending DNA samples to a third-party sequencing service and requests the produced genomic data to be transferred to our system, as shown by the arrow (1) in the figure. For each DNA sample, a sequencer produces raw images and then converts the image data to short read sequences (or reads, for brevity) of the genome. The read sequences are transferred to our system by shipping hard disks, as shown by the arrow (2). The data volume is usually hundreds of gigabytes per genome sample. Once the data arrives at our system, the user can issue a request to process the data, including alignment of read sequences and detection of variations against a reference genome, as shown by the shaded box labeled as "II. Data Processing". The output of this module, including a whole genome sequence and variations detected, are stored in a database for further analysis. Afterwards the user can upload additional patient information, and initiate extensive analyses that combine genomic data and patient data. Such analyses are handled by the module labeled as "III. Deep Analysis", which automatically discovers patterns of both statistical significance and biological meanings.

Details of the workflow for data processing and deep analysis are shown in Figure 1(b). For completeness, this workflow also includes data acquisition and pre-analysis at the sequencing service.

1) *Data acquisition*: A human genome has approximately 3 billion bases and each base has a letter of 'A', 'C', 'G' or 'T'. Most current sequencing technologies capture image data for each base being sequenced. Such raw data is then parsed into short read sequences of l bases (l depends on the sequencing machine), where each base has a specific base call, a letter of 'A', 'C', 'G' or 'T', and an assigned quality value (the likelihood that the base call is correct). For each genome sample, a sequencer usually produces 10's to 100's millions of read sequences of 30-1000 bases each.

2) *Pre-analysis*: The pre-analysis step evaluates the quality of each read sequence, removes poor quality reads, trims the poor quality bases at the two ends of each read, and formats the data for downstream processing (e.g., using the FASTQ format).

3) *Alignment:* Then the short read sequences are aligned against a reference genome. Figure 2 shows an example where the sequenced genome differs from the reference genome with two true mutations, $A \rightarrow C$ and $C \rightarrow A$. In this example, nine read sequences are aligned against the reference genome with up to five mismatches allowed per read—such mismatches must be allowed in order to detect mutations, which occur in every person's genome. The first four reads differ from the reference genome on the two bases where mutations occur among others, but the letters do not agree with the true genome. Most likely these reads have been mis-aligned to this fragment of the genome. The bottom five reads have the correct letters for the two bases where mutations occur, but have three additional mismatches, in relatively close positions, that differ from the true genome. Such mismatches can either come from errors in raw data or indicate that these reads should be aligned somewhere else. As can be seen, proper alignment for variation detection is a challenging problem, which we discuss more shortly.

4) *Variation discovery:* After alignment, the next step detects a range of genomic variants against the reference genome, including single nucleotide variants (SNPs), small insertions/deletions (INDELs), and large structure variants such as copy number variants (CNVs), inversions, and translocations. There is hardly any commercial software that can detect all of these variants. In our system, we support most forms of variants above using customized algorithms.

5) *Validation:* In the early phase of the workflow development, we plug in an additional step to validate detected genomic variations using other reliable, but labor-intensive methods. The validation results, e.g., false positives of the detected variations, provide feedback for improving the alignment and variation detection algorithms.

6) *Search and Integration:* The reported genomic variations are used to search existing knowledge bases to obtain associated information and integrated with patient information such as phenotypes.

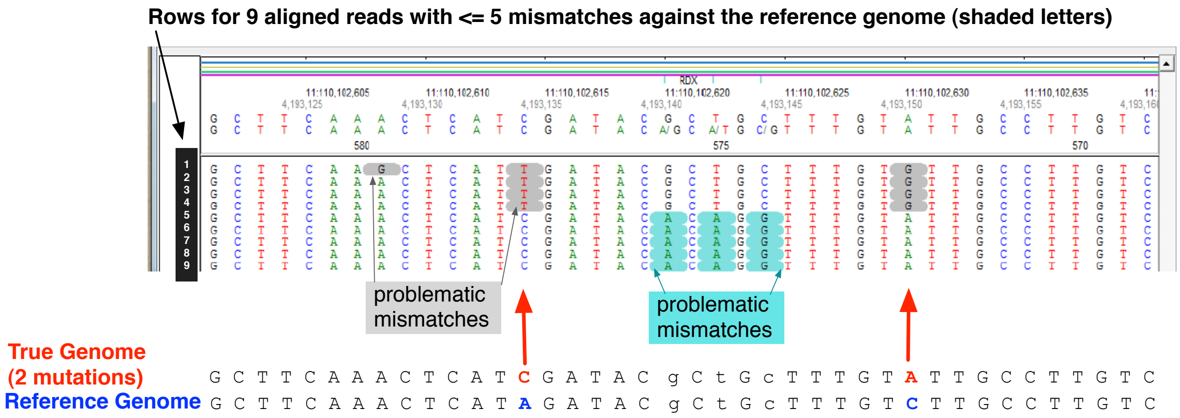

Rows for 9 aligned reads with <= 5 mismatches against the reference genome (shaded letters)
problematic mismatches
problematic mismatches
True Genome (2 mutations)
G C T T C A A A C T C A T C G A T A C g C t G c T T T G T A T T G C C T T G T C
Reference Genome
G C T T C A A A C T C A T A G A T A C g C t G c T T T G T C T T G C C T T G T C

Figure 2: Examples of poor alignments of read sequences.

7) *Deep analysis:* The last step produces high-level information with biomedical meanings from all available data, e.g., for understanding associations between genomic variations (e.g., SNPs and CNVs) and clinical phenotypes, causal relationships between those variations and phenotypes, or functional pathways in response to evolutionary, environmental, and physiological changes.

In the following, we discuss several key technical issues.

2.1 Error Tracking and Diagnosis

Today's genomic data processing systems are plagued with false positives of processed data: many detected genomic variations are reported to be false based on validation using safer but labor-intensive methods. Such errors severely affect subsequent biomedical analyses. The reasons for such errors are two-fold: (1) *Data errors:* Genomic data is inherently noisy. Due to the limitations of sequencing technology, errors occur in raw data at the rate of roughly 1 out of 100 bases. While sequencers usually create 30-50 reads of each base, the huge amount of noisy data still challenges software for alignment and variation detection. (2) *Software errors:* Genome processing software has to deal with both noisy data and intrinsic mutations in human genomes. When reads are aligned against a reference genome, it is hard to distinguish the mismatches due to data errors and those due to mutations. When such confusion propagates to variation detection, many false positives occur in output.

Quality scores of input and processed data: Our system is designed to have data quality as a first-class concept through all processing steps. We extend the notion of *quality score* beyond that for read sequences only in existing systems. More specifically, we devise techniques to assign quality scores to all forms of data in the workflow, including input data, intermediate data, and final output:

(1) Input data, the read sequences of a sample, is provided by the sequencer with a quality score for each base in a read. For example, the quality score of the 'A' letter in the read 'ACCGTT' is 10. This score, called "Phred quality score," is a property logarithmically related to the probability that the base-calling 'A' is an error.

(2) Propagating quality scores of the bases of a read to the alignments of the read is largely an unsolved problem. In our system, when a read is aligned to a position on the reference genome with m mismatches, we consider all possibilities of combining data errors and real mutations: Among the m mismatches, m_1 of them arise from data errors with the probabilities indicated by the quality scores of these m_1 mismatched bases, and the rest of $m - m_1$ mismatches are due to mutations, with the probability of having $m - m_1$ mutations in this part of the genome characterized by a Poisson distribution. We enumerate m_1 from 1 to m, compute probabilities of all these cases, and choose the case with the highest probability as the best explanation of this alignment. If a read has multiple alignments, we produce a probability for each of them.

(3) Propagating quality scores through variation detection algorithms is an even harder problem and requires fundamental research on this topic. We adopt a recent framework, called GASVPro [3], which combines the probabilities of alignment errors with the possibility that the number of reads per base deviates from an expected number. GASVPro, however, does not specify how to generate probabilities of alignment errors; our proposed technique above can be plugged into the this framework to produce such probabilities. Furthermore, GASVPro only considers two types of structural variation, deletion and inversion. Our system extends it to support other types such as translocation.

(4) Finally, our system further pushes quality scores through the Deep Analysis module to the final output. Take association rule mining for example. Each mined rule has structural variants in the head of the rule and phenotypes in the body of the rule. When structural variants are annotated with quality scores (probabilities for being wrong), techniques such as [2] can be used to deal with association rule mining in the probabilistic setting.

Quality control mechanisms: The quality scores that we develop allow us to employ a range of *quality control* mechanisms. Besides the obvious use of these scores to prune low quality data, they also allow us to mitigate various software errors. For instance, if a read has multiple possible alignments, existing software makes rather ad-hoc choices of one alignment or a few alignments. Our system can choose the top one or top few alignments based on quality scores. If a read r has two top alignments at locations u_1 and u_2 with similar scores, we can examine the set of reads mapped to each of u_1 and u_2 with their quality scores, derive the consensus for each location, and choose between u_1 and u_2 the more likely alignment of the read r based on its similarity with the consensus. Other quality control mechanisms are omitted due to space constraints.

2.2 Algorithm Development

We first developed a baseline workflow using existing software including BWA [8] for alignment, GATK [4] for SNP and INDEL calling, and association mining for genomic variations and patient phenotypes [5]. We then improved many algorithms used in the workflow for more functionality and improved results:

Alignment and variation discovery: Existing algorithms for alignment and variation discovery have some severe limitations. One is to use m mismatches for both SNP and INDEL calling. As the need for detecting large INDELs grows, using a small m value prevents the software from detecting them. In our system, we allow a larger value of *m* for mismatches in INDEL detection. However, the additional mismatches allowed will cause many false positives in alignment. Hence, the quality control mechanisms described above are used aggressively to prune erroneous alignments. Another limitation of existing software is that only one alignment of a read is considered for variation detection. As our quality control mechanisms have pruned many bad alignments, we pay the overhead of considering multiple alignments of a small set of reads in order to return genomic variations of higher quality.

Association mining: Unlike traditional association mining for transaction data, the genome-wide association study presents several main differences: First, as genomic variations are rare in nature, the extremely low support for such variations makes existing algorithms highly inefficient or unable to complete. Second, the interestingness metric for association rules is usually confidence, which produces too many trivial and repetitive rules in the genomic domain and hides truly interesting ones. Third, large structural variants such as CNVs are never fully aligned across different patients. Hence, they cannot be used as a fixed vocabulary of items as in existing algorithms. Instead, they should be divided into small fragments and mined for association by considering proximity of these fragments. Our algorithm extends a recent one [9] to support a new interestingness metric, extremely low support, and proximity-aware mining.

2.3 Parallel Processing for Performance

We further consider MapReduce style parallel processing for improved performance. We profiled our workflow to identify the cost associated with each step. We found that alignment, quality score recalibration, and some variation discovery algorithms are expensive operations. Therefore, we develop ways to parallelize them using the open-source Hadoop system. In the interest of space, we highlight a few below: (1) *Alignment:* Mapping reads to the reference genome is a computationally expensive step. The problem, however, is embarrassingly parallel as each sequence can be aligned independently of each other. We can run instances of the alignment program on different nodes using Hadoop. (2) *Quality score recalibration:* The quality scores returned by the sequencer often differ from the actual error rates present in the data because they can be affected by many covariates such as the machine cycle, the position of a base within a read, neighboring bases etc. To account for these factors, the quality scores in input data can be recalibrated (improved) based on the empirical error rates in groups of data, where the groups are defined by all possible values of user-defined covariates. We consider two methods to parallelize this step: we can either sort all the reads based on their mapped locations and then in parallel on multiple nodes, iterate over the set of reads overlapping with each location in the reference genome; or we can iterate over the unsorted reads and for each read probe the reference genome to update the empirical error rate.

Our design of parallel processing techniques addresses key issues regarding how to design multiple rounds of MapReduce jobs in a deep workflow of genomic processing, e.g., how to choose keys of MapReduce jobs, how to minimize the number of rounds of jobs, and how to choose between hash based and sort-merge based implementations of MapReduce. In addition, we consider optimization of the storage system to minimize intermediate data sizes.

3. RELATED WORK

We survey additional related work in this section. Crossbow [6] is a parallel pipeline for alignment and SNP detection. However, it currently does not support gapped alignment, hence of limited use. Seal [12] has integrated the BWA aligner with the Hadoop framework using Pydoop, an approach we adopt in our system. GATK [4] supports the MapReduce interface but not distributed parallelism. It can parallelize within a single multi-threading process or by *manually* dividing a region into independent pieces based on then chromosome and then running independent GATK instances. Hadoop-BAM [11] provides access to reads in binary, compressed BAM format stored in HDFS, which can be leveraged in our system.

4. DEMONSTRATION

In this demo, we will present a working prototype using real genomic data and real processing and analytical workloads. We have collected several terabytes of data, including (1) 10 trios (father, mother, and child) with 30 whole genome sequences and 3TB data, which is particularly useful for error tracking and diagnosis because genotypes in the child need to be consistent with those observed in the parents (otherwise, there is most likely an error); (2) 36 whole exome samples, each of which is 1% of a whole genome representing functionally relevant data; and (3) structural variants (CNVs) of four thousand patients with their phenotypes.

We will demonstrate the following features of our system: (1) *A workflow returning high-level biological information:* We will run the workflow from alignment, to variation discovery (including SNPs, small INDELs, and large structural variants), and finally to deep analysis. As an example of deep analysis, we find associations between genomic variations (common or rare) and phenotypes such as short, normal, or tall stature. We will show the data at each step of processing, including the relevant attributes and how they are transformed across steps, as well as how we compute novel quality scores for intermediate data and final output. (2) *Comparison of association mining algorithms*: We will compare our association mining algorithm against the state of the art [5] to demonstrate the changes needed for genomic data analysis. (3) *Data quality:* We will also compare our system with quality control mechanisms with the baseline workflow using existing software tools. We will show the difference in quality of processed results. (4) *Parallel processing:* We will run parallel processing of the most expensive components of the workflow on a cluster of nodes using Hadoop. We will show the resulting performance improvements.

5. REFERENCES

[1] M. Baker. Next-generation sequencing: adjusting to data overload. *Nature Method*, 7(7):495–499, 2010.

[2] C. Chui, et al. Mining Frequent Itemsets from Uncertain Data. In *PAKDD*, 2007.

[3] S. S. Sindi, et al. An integrative probabilistic model for identification of structural variation in sequence data. *Genome Biology*, 13(3), 2012.

[4] M. A. DePristo, et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics*, 43(5):491–498, 2011.

[5] J. Han, et al. Mining frequent patterns by pattern-growth: Methodology and implications. *SIGKDD Explorations*, 2(2), 2000.

[6] B. Langmead, et al. Searching for SNPs with cloud computing. *Genome Biology*, 10(11):R134+, Nov. 2009.

[7] B. Langmead, et al. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome biology*, 10(3), 2009.

[8] H. Li, et al. Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[9] J. Li, et al. Efficient discovery of risk patterns in medical data. *Artificial Intelligence in Medicine*, 45(1):77–89, 2009.

[10] Y. Li, et al. Wham: a high-throughput sequence alignment method. In *SIGMOD*, 445–456, 2011.

[11] M. Niemenmaa, et al. Hadoop-BAM: directly manipulating next generation sequencing data in the cloud. *Bioinformatics (Oxford, England)*, 28(6):876–877, Mar. 2012.

[12] L. Pireddu, et al. MapReducing a genomic sequencing workflow. In *Proc. of 2nd int'l workshop on MapReduce and its applications*, 2011.