Toward Efficient and Reliable Genome Analysis using Main-Memory Database Systems

Sebastian Dorok
Bayer Pharma AG
University of Magdeburg
Germany
sebastian.dorok@ovgu.de

Sebastian Breß
University of Magdeburg
Germany
sebastian.bress@ovgu.de

Horstfried Läpple
Bayer HealthCare AG
Germany
lapple@alumni.stanford.edu

Gunter Saake
University of Magdeburg
Germany
gunter.saake@ovgu.de

ABSTRACT

Improvements in DNA sequencing technologies allow to sequence complete human genomes in a short time and at acceptable cost. Hence, the vision of genome analysis as standard procedure to support and improve medical treatment becomes reachable. In this vision paper, we describe important data-management challenges that have to be met to make this vision come true. Besides genome-analysis performance, data-management capabilities such as data provenance and data integrity become increasingly important to enable comprehensible and reliable genome analysis. We argue to meet these challenges by using main-memory database technologies, which combine fast processing capabilities with extensive data-management capabilities. Finally, we discuss possibilities of integrating genome-analysis tasks into DBMSs and derive new research questions.

1. MOTIVATION

Mutations in organisms' genomes can trigger diseases such as cancer or cardiovascular disorders as well as influence the efficacy of drugs for disease treatment [9]. Hence, genomes have to be analyzed to detect mutations and to determine their implications on organisms' life in order to improve disease detection and treatment [3].

Foundation for comprehensive genome analysis is DNA sequencing that makes the genetic information encoded in genomes readable. In recent years, next generation sequencing techniques were developed, which sequence complete human genomes in several days [15]. The improvements in DNA sequencing facilitate new use cases for genome analysis such as personalized medicine that tailors disease treatment and drugs towards patients' genomes in order to enable tailor-made treatment for patients [20]. Thus, more and more genome sequencing data is generated in even shorter time that must be stored, integrated, processed, and analyzed.

Compared to the performance boost in DNA sequencing, processing and analysis of genome sequencing data is still very time consuming [14]. For that reason, data management, processing, and analysis techniques for genome data have to be improved to enable efficient and reliable genome analysis. Otherwise, the effective use of genome sequencing data will be limited [23].

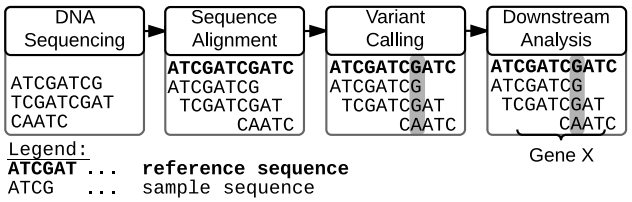

Figure 1: Exemplary Genome Analysis Process

In this paper, we contribute a summary of data-management challenges in genome analysis, which must be met to keep pace with DNA sequencing, but also to enable effective use of genome sequencing data. Moreover, we characterize different approaches for comprehensive genome analysis systems regarding their applicability to meet these challenges and discuss the use of main-memory database technologies as basis for such genome analysis systems.

The paper is structured as follows. We discuss the basic steps of genome analysis in Section 2 and describe open data-management challenges in genome analysis in Section 3. In Section 4, we motivate the use of main-memory database management systems as basis for tomorrow's genome-analysis systems and identify new research questions in Section 5.

2. BACKGROUND

Genome analysis comprises several steps to detect mutations in organisms' genomes and to assess their impacts on organisms' health and drug efficacy. In Figure 1, we depict an exemplary genome analysis process consisting of four steps. In the first step, the organism's genome is transformed into a digitally readable form via DNA sequencing. With current DNA sequencing techniques, it is impossible to read large genomes at once [23]. For that reason, the complete DNA is duplicated and arbitrarily fragmented into millions or billions of small overlapping base pair sequences. Then, the base pair sequence of each fragment is translated into a string composed of As, Cs, Gs, and Ts, called base calling.

SSDBM '14, June 30 - July 02 2014, Aalborg, Denmark

http://dx.doi.org/10.1145/2618243.2618276.

Thereby, each character encodes one of the bases adenine, cytosine, guanine, and thymine respectively. Next, the sequences are aligned to a known reference sequence in order to restore the original DNA sequence. Afterwards, in the variant calling step, differences between aligned genomes and reference genomes are determined by comparing the genome sequences site by site. Especially at sites where differences are detected, further downstream analyses are performed to assess the impact of mutations (e.g., a mutation has a cancer-causing characteristic or a negative impact on drug efficacy). Therefore, additional information from external data sources such as information on gene coding regions, known mutations, or protein interactions are needed.

Sequence alignment and variant calling are the most processing intensive steps within genome analysis. To efficiently compute them, algorithms are needed that use heuristics as well as index structures to speed up the computation. In practice, these algorithms operate on flat files and are implemented within command-line tools (e.g., samtools [11]). Moreover, approaches exist that use MapReduce frameworks and cloud computing to speed up sequence alignment and variant calling by parallelizing the processing [10]. To hide data transfer times to the cloud, stream-processing approaches are developed that start computation as soon as the first data items arrive at the processing node [8].

Downstream analyses rely on integrated data sources to establish relationships between genome mutations and their impacts on organism's life [3]. Beside command-line tools, approaches exist that use relational database management systems (DBMSs) to facilitate downstream analyses as DBMSs provide excellent data integration capabilities for heterogeneous data sources (e.g., Atlas [22]). Furthermore, approaches exist that already integrate some of the genome analysis steps into relational DBMSs [18, 19]. However, an approach that efficiently integrates all genome analysis steps into a DBMS does not yet exist.

3. DATA MANAGEMENT CHALLENGES

We now describe open data-management challenges in genome analysis. Thereby, we compare the applicability of different platforms to meet these challenges.

3.1 Analysis comprehensibility and reliability

Personalized medicine requires comprehensibility and reliability throughout the complete genome-analysis process. This is especially required because of the omnipresent data and result uncertainties within the genome analysis [16, 23]. If genome analysis is not comprehensible and reliable, medical assessments based on genome-analysis results are not transparent and trustworthy and the benefits of personalized medicine will be limited.

Using data-management capabilities, such as data integrity, data security, user management, and data provenance, is the foundation for a comprehensible and reliable genome-analysis process. Data-integrity and data-security capabilities ensure reliable analysis by keeping data consistent and valid at any time. Data-provenance capabilities allow to track what data contributed how to a certain analysis result. Moreover, user-management features allow to define roles and responsibilities of users within the complete genome analysis process making analyses more transparent and comprehensible.

Flat-file based command-line tools and distributed processing approaches for genome analysis are not designed for comprehensive data management, but for performance. In contrast, DBMSs are designed to provide comprehensive data-management capabilities. Thus, matured mechanisms for integrity control, data security, and user management exist. Moreover, DBMSs can be effectively used to provide data-provenance capabilities [6]. Nevertheless, currently, DBMSs are either used as central data storage for external tools or for downstream analysis to facilitate data integration. To provide a holistic approach that guarantees analysis comprehensibility and reliability throughout the complete genome analysis process, an integration of all steps of genome analysis into a DBMS is required.

3.2 Efficient large-scale data processing

Sequence alignment and variant calling are the most processing intensive tasks in genome analysis. The runtime of sequence alignment and variant calling depends on the number of reads to align and the size of the reference genome. Next generation sequencing techniques can sequence large genomes (e.g., the human genome that comprises nearly 3.2 billion base pairs) in several days and generate large amounts of reads that must be aligned [12].

Hatem et al. show in a recent performance benchmark that current state-of-the-art sequence alignment tools can align 1 billion base pairs in one hour [7]. In contrast, current DNA sequencers have already a throughput of 2.5 billion base pairs per hour [12]. In order to process the increasing amounts of sequencing data efficiently, approaches were developed that use distributed processing frameworks such as MapReduce to speed up the processing (e.g., Crossbow [10]). But with increasing amounts of data to process, data transfer times to cloud environments become significant. For example, Langmead et al., the authors of Crossbow, report that transferring their evaluation dataset to the Amazon cloud lasts over an hour [10]. Even between local nodes, the transfer of large amounts of data is significant. For example, CLC bio reports that pre- and postprocessing tasks, such as index creation and data transfer between storage and processing nodes, lasts over 11 hours that is 70% of the overall processing time of their alignment process [4].

Pavlo et al., show that large-scale data processing with parallel DBMSs is more efficient than using a MapReduce framework [17]. Thereby, they explain the speedup with the use of compression techniques that reduce the data volume and more sophisticated execution strategies that perform data transfers between nodes in a distributed processing environment only if necessary. Thus, we expect that the integration of processing-intensive genome-analysis tasks into DBMSs could improve the processing performance of a genome-analysis system. Moreover, we argue that the use of main-memory DBMSs as platform for genome analysis would further increase the performance potentials (cf. Table 1) as these systems have already shown their abilities to speed up database applications [13].

3.3 Extensible high-performance analysis

In order to reveal new insights on genome mutations and their impacts on organisms, it is necessary to enrich genome sequencing data with further information [3]. Thereby, the required information is stored in many heterogeneous data sources. Thus, matured data-integration capabilities are needed to integrate these heterogeneous data sources making them accessible for comprehensive downstream analysis. According to Mardis, downstream analysis is the most time consuming task in genome analysis [14]. Many users participate in analyses and access the database for various purposes. Thus, analyses must not only be fast, but also the interface

Data-management challenge	Flat-file based (e.g., command-line tools)	Disk-based DBMS	Main-memory DBMS
Analysis comprehensibility and reliability	−	+ + +	+ + +
Efficient large-scale data processing	+	+	++
Extensible high-performance analysis	◯	◯	◯

Table 1: Applicability of different platforms for genome analysis to meet data-management challenges Legend: − = not applicable, ◯ = possible, + = good, ++ = very good, + + + = outstanding

to the database must enable various kinds of analyses.

Currently, approaches exist that let users define work flows for genome analysis using a fixed set of specialized command-line tools [2]. The complexity and integration effort of such approaches increases with a higher number of supported tools. Thus, extending such approaches, if possible, is associated with high implementation efforts. Other approaches use traditional disk-based DBMSs to store data from different data sources in a homogeneous data schema, but are designed for specific use cases and provide specialized interfaces beside the standard SQL interface. To the best of our knowledge, there is no evaluation of performance and extensibility of these genome-analysis approaches.

We argue that, in contrast to command-line tools, DBMSs are the right platform for extensible genome analysis as they decouple the analysis application from the internal data representation. Thus, data is accessible via high-level query languages (e.g., SQL) and analyses can be integrated into the DBMS (e.g., using stored procedures), whereby the DBMS manages the efficient data access. Results of Schapranow and Plattner indicate that main-memory DBMSs enable high-performance analysis in the field of genome analysis [21]. Nevertheless, the applicability of DBMSs to provide the extensibility needed for genome analysis must be investigated in the future (cf. Table 1).

4. OUR VISION

In this section, we present our vision of a future genome-analysis system that meets the data-management challenges presented in Section 3. Therefore, we first describe our system design and then, we describe how to meet the challenges.

4.1 System design

Main-memory DBMSs are the most applicable approach to meet all open data-management challenges from Section 3 (cf. Table 1). For that reason, we suggest to use a main-memory DBMS as processing and analysis platform to enable efficient and reliable genome analysis. In Figure 2, we depict the system design of such a genome-analysis system.

All data is stored within the main-memory database. Every genome-analysis task should directly operate on the main-memory database. Therefore, we propose to integrate genome-analysis tasks (e.g., variant calling) into the DBMS. In case the integration of a genome-analysis task is not feasible, the DBMS should provide efficient interfaces that enable existing genome analysis tools to operate on the main-memory database. For example, sequence alignment could be implemented as special bulk-load functionality for genome data that reuses existing sequence-alignment tools.

4.2 Addressing data-management challenges

We now describe how our proposed genome-analysis system meets the data-management challenges from Section 3.

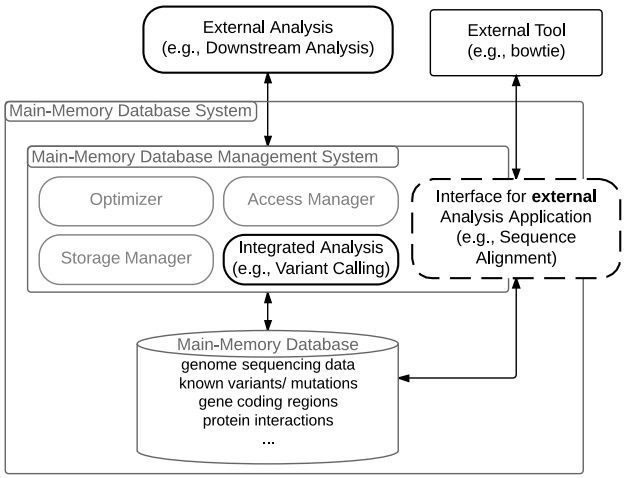

Figure 2: Genome Analysis using Main-Memory Database Systems

Extensible high-performance analysis.

Relational DBMSs provide declarative query languages such as SQL. Moreover, languages exists that allow the definition of stored procedures that are directly processed inside the database. Thus, just the analysis result must be transferred to a client instead of all data needed for the analysis. Another advantage of using declarative query languages for analyses is the use of optimized database operators that are designed to provide efficient data access.

Compared to command-line tools that have many parameters to control their execution, SQL is a more maintainable way to describe analysis tasks. Furthermore, the integration of analysis into the DBMS allows easier tracking of data access and manipulations that is crucial to guarantee data integrity as well as maintaining data provenance throughout the complete genome analysis process.

First results on variant calling indicate that an integration of genome analysis tasks into a main-memory DBMS is feasible and can be beneficial regarding performance [5].

Efficient large-scale data processing.

Main-memory DBMSs speed up database applications by orders of magnitudes. The performance boosts of main-memory DBMSs arise not only from the fact that data is held in main memory permanently but also from the use of optimized data structures and hardware-conscious algorithms [13] as well as architectural redesigns [24]. In order to benefit most from performance boosts due to main-memory use, genome analysis algorithms must be efficiently integrated into the DBMS. Therefore, we propose three strategies to integrate genome-analysis tasks into main-memory DBMSs:

Reuse of existing main-memory DBMS operators. This strategy reuses existing DBMS operators to efficiently implement genome analysis tasks (e.g., via SQL or stored procedures). Advantage of this strategy is that the used operator implementations are optimized for different hardware by DBMS vendors. Major disadvantage is the restricted ability to express complex algorithms.

Extending functionality of main-memory DBMSs. To avoid the major disadvantage of the previous strategy, this strategy implements genome analysis functionality directly into the DBMS. Certainly, the developer has to care about optimal and efficient implementation.

Interfaces to internal storage structures. In case neither the reuse of DBMS operators nor the extension of the DBMS is possible, the third strategy connects existing genome analysis tools via efficient interfaces to the main-memory DBMS. On the one hand, it is possible to reuse existing and mature genome-analysis tools. On the other hand, the interfaces must provide efficient access while guaranteeing data integrity.

Furthermore, DNA sequencing reads have promising characteristics for compression as the alphabet to describe them is limited to the four letters A, C, G, and T in best case. When storing every base of a read separately in one column instead of the complete string, dictionary encoding can be effectively applied as the dictionary comprises just four values. Thus, every single base value can be stored using just two bits instead of eight. Such dictionary encoding scheme can be efficiently applied to column-stores [1] that are often used in main-memory DBMSs. In recent work, we have shown that DNA sequencing reads can be efficiently compressed using compression schemes such as dictionary encoding [5].

Analysis comprehensibility and reliability.

DBMSs provide a comprehensive set of data management capabilities such as data integrity, data integration, data security, and also data provenance. Thus, a DBMS is the right platform to fulfill the requirements regarding comprehensive and reliable genome analysis.

5. CONCLUSION

In this work, we motivate to store and analyze genome data inside main-memory database systems, because they do not only provide excellent support for data management, but also allow fast analysis of large amounts of data. Our intention of using main-memory database systems for genome analysis are twofold. First, we want to improve data management quality of the complete genome analysis process. Second, we want to benefit from techniques that are already implemented in database systems, such as compression and query optimization to accelerate all genome-analysis steps.

In order to realize our vision of efficient and reliable genome analysis by using main-memory DBMSs, at least the following research questions need to be answered:

RQ 1 How to integrate processing intensive tasks (e.g., sequence alignment) efficiently into main-memory DBMSs?

RQ 2 Are high-level query languages such as SQL sufficient to express arbitrary genome-analysis tasks?

RQ 3 How to enable external analysis tools to access data structures of a main-memory DBMS efficiently?

RQ 4 How to guarantee consistency when allowing external tools to access internal data structures?

RQ 5 How to efficiently integrate, store, and process peta bytes of genome data in a main-memory DBMS?

6. ACKNOWLEDGEMENTS

The work of Dorok is supported by Bayer HealthCare AG.

7. REFERENCES

[1] Daniel J. Abadi et al. Integrating compression and execution in column-oriented database systems. In *SIGMOD*, pages 671–682, 2006.

[2] Daniel Blankenberg et al. Galaxy: a web-based genome analysis tool for experimentalists. *Curr Protoc Mol Biol*, 89:19.10.1–19.10.21, 2010.

[3] Y. Bromberg. Building a genome analysis pipeline to predict disease risk and prevent disease. *J. Mol. Biol.*, 425(21):3993–4005, 2013.

[4] CLC bio. Read mapping. Techn. rep., CLC bio, 2012.

[5] Sebastian Dorok et al. Toward efficient variant calling inside main-memory database systems. In *BIOKDD-DEXA*, 2014.

[6] Mohamed Y. Eltabakh et al. bdbms - a database management system for biological data. In *CIDR*, pages 196–206, 2007.

[7] Ayat Hatem et al. Benchmarking short sequence mapping tools. *BMC Bioinformatics*, 14(1):184, 2013.

[8] Romeo Kienzler et al. Incremental DNA sequence analysis in the cloud. In *SSDBM*, pages 640–645, 2012.

[9] Eric S. Lander. Initial impact of the sequencing of the human genome. *Nature*, 470(7333):187–197, 2011.

[10] Ben Langmead et al. Searching for SNPs with cloud computing. *Genome Biol.*, 10(11):R134, 2009.

[11] Heng Li et al. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[12] Lin Liu et al. Comparison of next-generation sequencing systems. *J. Biomed. Biotechnol.*, 2012:1–11, 2012.

[13] Stefan Manegold et al. Optimizing database architecture for the new bottleneck: memory access. *The VLDB Journal*, 9(3):231–246, 2000.

[14] Elaine Mardis. The $1,000 genome, the $100,000 analysis? *Genome Med*, 2(11):84, 2010.

[15] Michael L. Metzker. Sequencing technologies - the next generation. *Nat. Rev. Genet.*, 11(1):31–46, 2009.

[16] Jason O'Rawe et al. Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing. *Genome Med*, 5(3):28, 2013.

[17] Andrew Pavlo et al. A comparison of approaches to large-scale data analysis. In *SIGMOD*, pages 165–178, 2009.

[18] Astrid Rheinländer et al. Prefix tree indexing for similarity search and similarity joins on genomic data. In *SSDBM*, pages 519–536, 2010.

[19] Uwe Röhm and José A. Blakeley. Data management for high-throughput genomics. In *CIDR*, 2009.

[20] Wolfgang Sadée and Zunyan Dai. Pharmacogenetics/ -genomics and personalized medicine. *Hum. Mol. Genet.*, 14(suppl 2):R207–R214, 2005.

[21] Matthieu-P. Schapranow and Hasso Plattner. HIG - an in-memory database platform enabling real-time analyses of genome data. In *BigData*, pages 691–696, 2013.

[22] Sohrab P. Shah et al. Atlas - a data warehouse for integrative bioinformatics. *BMC Bioinformatics*, 6:34, 2005.

[23] Jay Shendure and Hanlee Ji. Next-generation DNA sequencing. *Nat Biotechnol*, 26:1135–1145, 2008.

[24] Michael Stonebraker et al. The end of an architectural era (It's time for a complete rewrite). In *PVLDB*, pages 1150–1160, 2007.