In life sciences, scientists collect increasing amounts of data that must be stored, integrated, processed, and analyzed efficiently to make effective use of them. Thereby, not only the huge volume of available data raises challenges regarding storage space and analysis throughput, but also data quality issues, different data formats, and compliance issues such as data provenance make it hard to handle life science data. To address these challenges, advanced data management techniques are required. Otherwise, the effective use of life science data will be limited. In our research, we want to develop data management techniques for life science data, especially for genome data. Thereby, one question is whether general purpose techniques and methods for data management are enough or whether we have to develop specialized solutions for the life science use case(s).

**Genome data compression (Bachelor/ Master)**

Technological advances in genome sequencing lead to ever increasing amounts of genome data. In order to reduce the amounts of genome data, compression techniques are applied such as binary encoding of source files (e.g., BAM files). Moreover, more sophisticated compression schemes such as referential compression exist that leverage characteristics of genome data.

The goal of this thesis is to create a survey about compression techniques that can be applied to genome data and to describe their advantages and disadvantages. Moreover, the applicability of such compression schemes in database systems should be examined.

Literature:


Storing genome data in main-memory DBMSs (Bachelor/ Master)

In our research, we developed a base-oriented database schema to store aligned genome data [1]. This database schema enables direct access to every single base at a certain position in a genome using standard SQL. So far we evaluated our database schema in prototypical DBMS implementations.

The goal of this thesis is to implement the base-oriented database schema in a commercial main-memory DBMS. Therefore, our partner IPK Gatersleben provides access to an Oracle 12c instance. Additionally, adaptations to IPK specific use cases should be determined and implemented. Afterwards, storage efficiency and query performance of the implemented solution should be evaluated.

Literature:


Storage layouts for genome data in main-memory DBMSs (Master)

In order to overcome the new access bottleneck in main-memory database management systems, cache-efficient algorithms and data structures are required [2]. In our research, we developed a base-oriented database schema to store aligned genome data [1]. This database schema enables direct access to every single base at a certain position in a genome using standard SQL. Certainly, for different use cases a sequence-oriented database schema could be more beneficial [3].

The goal of this thesis is to evaluate different database schemata to store aligned genome data regarding their compression efficiency, query flexibility, and query performance. Thereby, different storage layouts such as column-oriented and row-oriented storage should be considered.

Literature:


Probabilistic variant calling in RDBMSs (Bachelor/ Master)

In our research, we developed a base-oriented database schema to store aligned genome data [1]. This database schema enables direct access to every single base at a certain position in a genome using standard SQL. We use this database schema to call variants in
genres on demand using SQL and a user-defined aggregation function. Currently, we use a frequency-based approach to call variants that is well suited for high coverage genome data. For low coverage genome data, probabilistic variant calling approaches have been developed [2]. Such approaches require access to base values and base call quality values during computation. Our current framework for user-defined aggregation functions does only support accessing one column at a time.

The goal of this thesis is to develop a concept for probabilistic variant calling in an RDBMS using our base-oriented database schema for aligned genome data. In a next step, the concept should be implemented in our research prototype CoGaDB and compared against the existing frequency based approach.

Literature:


Fast loading of disk-resident files into main-memory DBMSs (Bachelor/Master)

To process huge amounts of data in a relational main-memory DBMS, cache-efficient algorithms and data structures are required [2]. If data often changes or often new data has to be imported, efficient data loading becomes as important as efficient data processing. Mühlbauer et al. present a technique to speed up the bulk loading functionality in main-memory database systems [3]. This technique is especially designed for CSV files. In this thesis, the ideas of speeding up bulk loading of CSV data should be applied to bulk load genome data files such as FASTA or SAM/BAM files. The evaluation system is CoGaDB [1]. Moreover, the efficient loading of disk-resident data already stored in CoGaDB’s native storage format should be investigated.

Literature:


Extending CoGaDB with a data visualization tool for genome data (Bachelor)

A key factor of success for a genome data management solution is the possibility to enable visualization and exploration of genome data. Currently, our research prototype for storing and querying aligned genome data lacks a data visualization mechanism.

The goal of this thesis is to extend our research prototype that is based on CoGaDB [1] with a visualization mechanism. To solve this task, we suggest to write a file export for CoGaDB that converts the results of a SQL query into a file format such as SAM or BAM [3]. Using such a solution, state-of-the-art visualization tools such as IGV [2] or Tablet [4] can access the file and visualize the content. Additionally, more complex integration scenarios can be explored such as modifying existing tools to directly retrieve data from CoGaDB via the SQL interface.

Literature:


Mapping graph databases to relational databases (Bachelor/ Master)

Many databases exist in bioinformatics and biology that are of different type, content, and structure. Besides relational database systems, other kinds such as graph database systems are in use as they provide a more natural way to store certain data and information. To make such information available via a unified access layer such as SQL, a mapping between graph and relational database schemata is necessary.

The goal of this thesis is to provide an overview of existing mapping approaches between graph and relational databases and to determine their characteristics. In a next step, the mappings should be implemented and evaluated. Therefore, it is possible but not mandatory to use the foreign data wrapper mechanism of PostgreSQL and to extend it to support graph databases such as Neo4J.

Literature:


Bring your own topic (Bachelor/ Master)

Besides the presented topics above, I encourage students to propose their own topics, just contact me.