A NoSQL Solution For Bioinformatics Data Provenance Storage

Ingrid Santana[1], Waldeyr Mendes C. da Silva[2], and Maristela Holanda[1]

[1] UnB, University of Brasília - Brazil
ingrid95sl@gmail.com, mholanda@unb.br
[2] IFG, Federal Institute of Goiás - Brazil
waldeyr.mendes@ifg.edu.br

Abstract. Provenance data can support the reproducibility of experiments providing the history of the data in a scientific workflow. Bioinformatics generates an increasing amount of data, which are often analyzed employing workflows. This paper proposes a way to manage automatic executions of Bioinformatics workflows, storing their provenance and raw data in the MongoDB NoSQL database system. It uses a program that manages three different data models, a referenced, an embedded, and a hybrid data model for purposes of comparison. The results showed general advantages and disadvantages for each data model and some particularities of Bioinformatics.

Keywords: NoSQL, MongoDB, Big Data, Bioinformatics, Data provenance.

1 Introduction

With the accelerated rate of advances in technology, the need has arisen for databases that are capable of both efficiently storing and processing massive amounts of data, as well as writing and reading them with high performance [1]. This scenario has given rise to the concept of Big Data, which is a term used to describe massive volumes of structured or non-structured data that can be managed mostly with NoSQL Databases [2].

In addition to managing the data storage of scientific *in silico* experiments, it is important to handle their production to support reproducibility. Data provenance is the history or the profile of an execution, capable of answering questions related to the origin of data [3].

This article presents a computational solution for managing the data provenance of Bioinformatics workflows using a NoSQL document-based database. The solution can be implemented using different data models, which are shown and discussed.

This article is organized as follows: Section 2 reviews some important concepts in Big Data, NoSQL, MongoDB, data provenance and Bioinformatics workflows. Section 3 discusses some related research works. The method used is presented in Section 4, and the results are shown in Section 5. The main conclusions and future work are listed under Section 6.

2 Background

2.1 Big Data and NoSQL

Nowadays, data generation is growing in velocity, volume, and variety. It has created a demand for data processing and storage with high performance written and read, creating the phenomenon known as Big Data [1]. NoSQL was born in this scenario and is an eclectic and unstructured database model capable of providing storage with flexibility [4], [5]. It has been shown to be able to collect a vast and variable volume of data in distributed environments and provide answers to scalability problems [6].

MongoDB is a document-based NoSQL database capable of providing relatively powerful query capabilities, document-level atomicity, and support for complex data types [1], [7]. It has been developed for the storage of information with high performance and scalability [5]. In addition, it is currently the most popular application in the category of document-oriented databases [5]. MongoDB organizes the documents in collections of documents, which are stored in the database under two primary storage formats [8], [5]:

- Embedded: a denormalized model that stores the structure by grouping data according to its familiarity in a large document.
- Referenced: a standardized model that works with standardized formats that use connections to create relationships with other documents.

There is also the possibility of creating a hybrid model that has both characteristics [5]. One factor to note is that the size of MongoDB documents is limited to 16 MB [8]. As an alternative to creating larger documents, the convention uses the GridFS concept as an alternative [8]. With GridFS, it is possible to insert larger documents by dividing the file into parts called chunks [8].

2.2 Data Provenance and Bioinformatics Workflows

Data provenance is characterized as the history or profile of workflow executions, capable of answering questions related to the origin of data, providing the lineage of data generation, use, and processing [3], [9]. Essential for tracking data among the different stages of a workflow [10], its usefulness lies in the fact that, once the data provenance is recorded, it is easier to create, recreate, evaluate or modify computational models or scientific experiments based on the accumulation of knowledge of what was done [11]. Databases to store data provenance has scientific importance because they enable the use of the data's origin as a raw material for reproducing the experiments [12].

In the field of Bioinformatics, most of the data generated come from the high-throughput of Next Generation Sequencing (NGS) [6] [13]. Genomics is a

Big Data science and will expand and achieve 1 Zbp in the next years according to the historical growth rate [14].

A workflow is a set of tasks with input and output information to be followed in order to achieve a final goal [15]. In Bioinformatics, the input files are the sequences generated by the NGS sequencers. Bioinformatics workflows are often organized in well-defined phases that make use of both these files and several programs or tools and libraries [12], [16]. The purpose of creating a workflow for biological data analysis and interpretation is to acquire specific, unambiguous and consistent knowledge that is repeatable under the same conditions [16].

3 Related Works

The usefulness of data provenance and its application in NoSQL databases based on documents is a point of discussion in Mattoso *et al* [11]. However, there are also several other works with different concepts and ideas worth highlighting. Tao Li *et al* [3] proposes a structure called ProvenanceLens, which provides provenance management in cloud environments and compares its performance by using MySQL, MongoDB, and Neo4J. Tao Li *et al* also enumerates characteristics that are desirable for its systems and categorize the provenance into types.

Costa *et al* [17] were able to capture the data provenance of a Workflow using the PROV-Wf model. However, the provenance was captured and stored in an RDBM, and the workflow was run in both local and cloud environments. Hondo *et al* [18] present a comparative study between the NoSQL Cassandra, MongoDB and OrientDB databases through the execution of a bioinformatics workflow and the storage of its data provenance. Continuing this work, Hondo *et al* [19] also conduct a study of bioinformatics workflows by storing data provenance in the previously mentioned databases and then conducting queries. Reis *et al* [5] carried out a comparative analysis between embedded, referenced and hybrid data models in MongoDB. The models, although not having bioinformatics data, were compared to each other with queries and were in the Big Data domain.

This work differs from previous ones by obtaining the data provenance of Bioinformatics workflows when executing it automatically with a program created to, not only perform the steps of the workflow, but also to acquire its data provenance. The data provenance information collected is then inserted into MongoDB by the same program in three different models (embedded, referenced and hybrid). These three data approaches are then compared to each other, taking into account the time the program took to create each database and also the capability and facility to make queries.

4 Method

Reis *et al.* [5] carried out a comparison and performance analysis of three different variations of a data model. The first variation used fully embedded documents. The second used fully referenced documents, and the third variation considered a hybrid form between the two previous ones. Hondo *et al.* [19] proposed a data provenance storage model for Bioinformatics workflows, which can be seen in Figure 1.

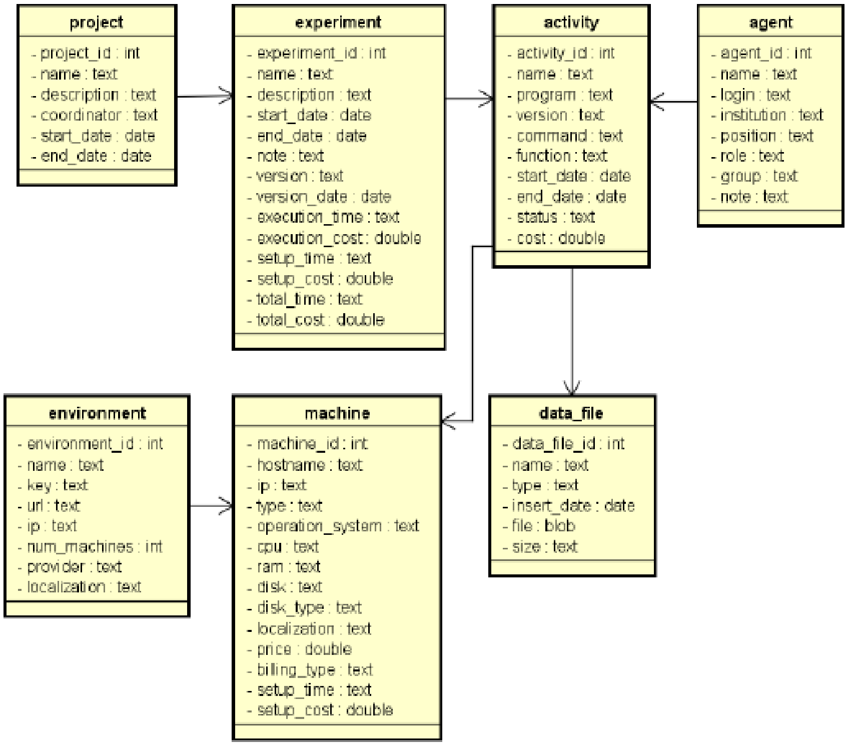

Fig. 1. Data model previously proposed by Hondo *et al.* [19].

In this work, inspired by the approach of Reis *et al.* [5], we adapted the model proposed by Hondo *et al.* [19] to create different document-based data models to store data provenance in MongoDB. We created three data models: a referenced model, an embedded model, and a hybrid model. The proposed models kept some features from [19] while others were adjusted. With the initial focus on the local storage of data provenance for general Bioinformatics workflows, the data models were updated to hold environment data, which are not present in [19]. Figures 2, 3 and 4 show the three proposed data models.

We have created a program able to manage the workflow execution, retrieving and storing the data provenance. This program gets the required data for the workflow execution, with the ability to separate the input and output files. Then, it executes the commands for the workflow activity through system calls. For

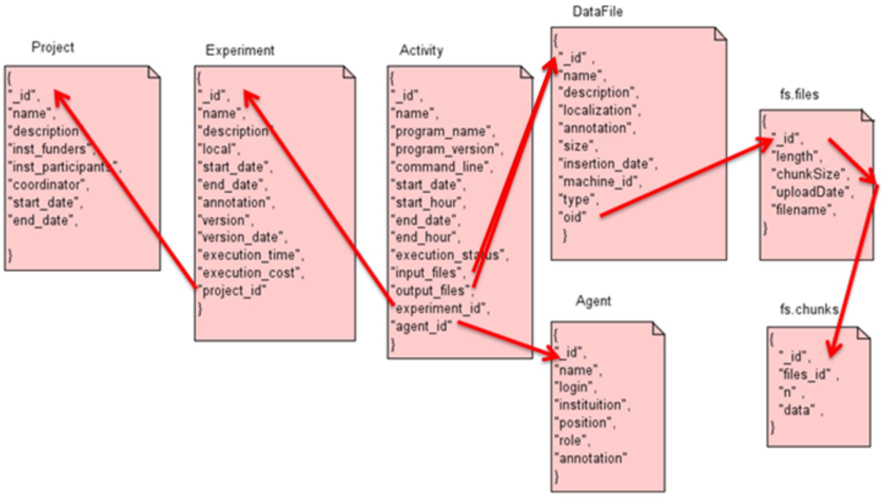
Project
{ "_id", "name", "description", "inst_funders", "inst_participants", "coordinator", "start_date", "end_date", }
Experiment
{ "_id", "name", "description", "local", "start_date", "end_date", "annotation", "version", "version_date", "execution_time", "execution_cost", "project_id" }
Activity
{ "_id", "name", "program_name", "program_version", "command_line", "start_date", "start_hour", "end_date", "end_hour", "execution_status", "input_files", "output_files", "experiment_id", "agent_id" }
DataFile
{ "_id" , "name", "description", "localization", "annotation", "size", "insertion_date", "machine_id", "type", "oid" }
fs.files
{ "_id", "length", "chunkSize", "uploadDate", "filename", }
fs.chunks
{ "_id", "files_id" , "n" , "data" , }
Agent
{ "_id", "name", "login", "instituition", "position", "role", "annotation" }

Fig. 2. Referenced data model for data provenance storing in MongoDB. Red arrows represent the references.

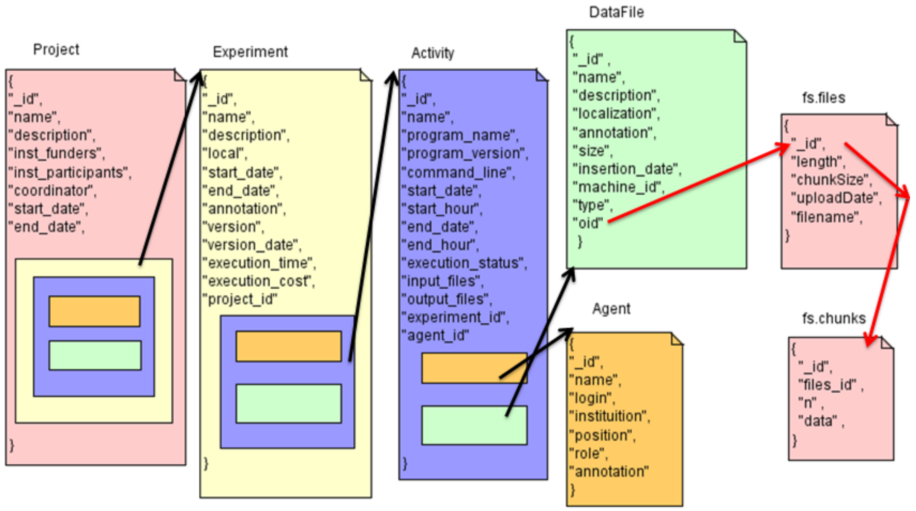
Project
{ "_id", "name", "description", "inst_funders", "inst_participants", "coordinator", "start_date", "end_date", }
Experiment
{ "_id", "name", "description", "local", "start_date", "end_date", "annotation", "version", "version_date", "execution_time", "execution_cost", "project_id" }
Activity
{ "_id", "name", "program_name", "program_version", "command_line", "start_date", "start_hour", "end_date", "end_hour", "execution_status", "input_files", "output_files", "experiment_id", "agent_id" }
DataFile
{ "_id" , "name", "description", "localization", "annotation", "size", "insertion_date", "machine_id", "type", "oid" }
fs.files
{ "_id", "length", "chunkSize", "uploadDate", "filename", }
fs.chunks
{ "_id", "files_id" , "n" , "data" , }
Agent
{ "_id", "name", "login", "instituition", "position", "role", "annotation" }

Fig. 3. Embedded data model for data provenance storing in MongoDB. Red arrows represent the references, and the black arrows represent embedded documents.

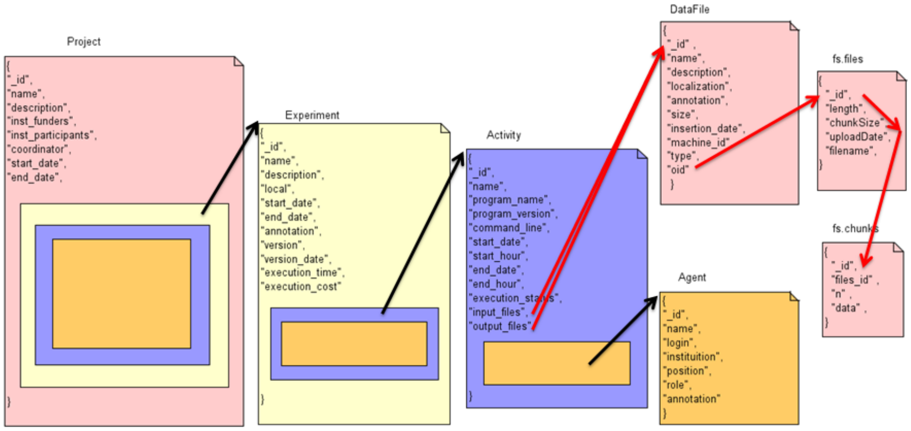

Fig. 4. Hybrid data model for data provenance storing in MongoDB. Red arrows represent the references, and the black arrows represent embedded documents.

each workflow step, the program retrieves the data provenance and store it in several structures and variables. After executing all the workflow commands and their completion, the provenance data and the files raw data are inserted into the MongoDB according to the three different proposed data models. The program inserts the data provenance in using each proposed data model sequentially so that each data model can have its exclusive database.

As the biological raw data are often bigger than the MongoDB size limit (16 MB) for documents, the program uses a conversion strategy (chunk) forcing the creation of GridFS collections. In the three models it is also possible to see the *fs.chunks* and *fs.files* collections referring to the collections automatically created by GridFS to support these biological raw data files.

After the creation of all three databases, the program finishes its execution. In addition, the program closes the connection with MongoDB after each database accomplishment.

Figure 5 summarizes a Bioinformatics workflow, which was used as a case study. In this workflow, using the Hisat tool [20], reads (sequences of DNA obtained from NGS sequencers) were mapped in a reference genome, which in this case was the 22nd chromosome of the human genome. After conversion of the mapping file format by the Samtools toolkit [21], a count of the number of mapped reads in gene regions was made using the HTSeq tool [22]. The experiment was repeated four times in order to minimize variances in the analyses. Before each run, the database was completely cleaned to avoid data overlapping and the data provenance was inserted into the MongoDB.

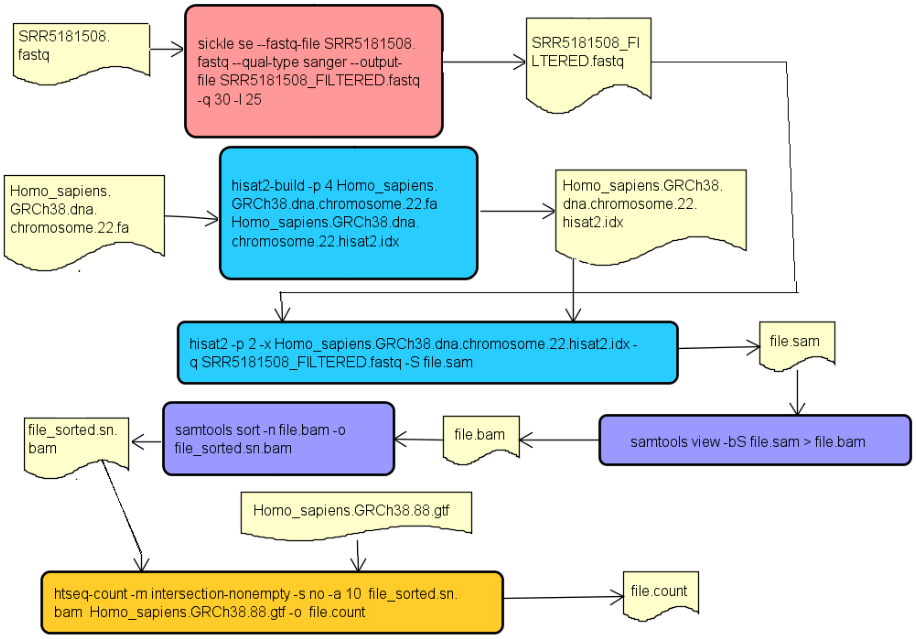
SRR5181508.
fastq
sickle se --fastq-file SRR5181508.
fastq --qual-type sanger --output-
file SRR5181508_FILTERED.fastq
-q 30 -l 25
SRR5181508_FI
LTERED.fastq
Homo_sapiens.
GRCh38.dna.
chromosome.22.fa
hisat2-build -p 4 Homo_sapiens.
GRCh38.dna.chromosome.22.fa
Homo_sapiens.GRCh38.dna.
chromosome.22.hisat2.idx
Homo_sapiens.GRCh38.
dna.chromosome.22.
hisat2.idx
hisat2 -p 2 -x Homo_sapiens.GRCh38.dna.chromosome.22.hisat2.idx -
q SRR5181508_FILTERED.fastq -S file.sam
file.sam
file_sorted.sn.
bam
samtools sort -n file.bam -o
file_sorted.sn.bam
file.bam
samtools view -bS file.sam > file.bam
Homo_sapiens.GRCh38.88.gtf
htseq-count -m intersection-nonempty -s no -a 10 file_sorted.sn.
bam Homo_sapiens.GRCh38.88.gtf -o file.count
file.count

Fig. 5. Executed Bioinformatics workflow. Colors represent a specific phase: pink for filtering, blue for mapping, purple for file conversion and orange for counting.

5 Results

Results showed that it was possible to adequately capture and store the workflow data provenance in all three proposed data models through the built program.

The data provenance, both directly related to the experiment and for the computational environment, was suitably stored into MongoDB applying the three formats: referenced, embedded, and hybrid. The insertion times for each model - shown in Figure 6 - were noted with the help of the log file generated by the built program, and they do not include the time it took to execute the workflow. When comparing the referenced, embedded and hybrid models, the difference in the amount of time it took to create each model was minimal. Nevertheless, the hybrid model took the shortest time to be created.

We designed and performed a set of queries to collect examples of relevant data provenance information, which allowed us to ensure the consistency of the data among the proposed data models. Also, it was possible to compare the complexity of query construction among the proposed models. For the sake of comparison, the queries answered the same questions asked in [19], and they quickly retrieved consistent data provenance information as can be seen in Table 1. Upon comparing the queries, the fully referenced model queries can be at least a little harder to create, and the information more, if the required information includes two or more documents and their relationships.

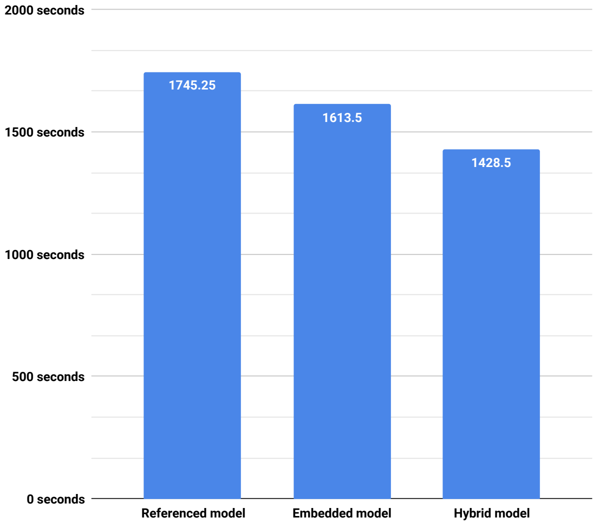

Fig. 6. Average data provenance insertion times (in seconds) into MongoDB for the three proposed data models.

Table 1. Queries for each model.

What are the names and versions of the used programs?	
Referenced	db.activity3.aggregate([{$match:{”experiment_id”:”1”}}, {$unwind:”$_id”},{$group:{_id:{program: ”$program_name”,version:’$program_version’}}}])
Embedded	db.default.aggregate([{$match:{$and:[{id:”1”}, {”experiment.id”:”1”},]}},{$unwind: ”$experiment.activity”},{$group:{_id: program: ”$experiment.activity.program_name”, version:’$experiment.activity.program_version’}}}])
Hybrid	db.project1.aggregate([{$match:{$and:[{id:”1”}, {”experiment.id”:”1”},]}},{$unwind: ”$experiment.activity”},{$group:{_id: program: ”$experiment.activity.program_name”, version:’$experiment.activity.program_version’}}}])
How many activities were performed in the first experiment?	
Referenced	db.activity3.count({experiment_id: ”1”})
Embedded	db.default.aggregate([{$match:{”experiment_id”: ”1”}}, {$project:{numberOfActivities: {$size:”$experiment.activity”}}}])
Hybrid	db.project1.aggregate([{$match:{”experiment_id”: ”1”}}, {$project:{numberOfActivities: {$size:”$experiment.activity”}}}])

6 Conclusion

The proposed data models were able to store data provenance from a Bioinformatics workflow in MongoDB, through a program that managed the whole process. The storage embraces both raw data and data provenance, including metadata of the environment.

Regarding the database size, the provenance data played a less important role representing a lower percentage of the data compared to the workflow raw data. Therefore, the difference in the amount of time it took to create each model was minimal. Nevertheless, the hybrid model showed better data insertion performance due to the combination of referenced and embedded documents that created a set of documents and collections which could be inserted into MongoDB more quickly.

Upon comparing the queries, we considered it to be at least a little harder to create queries for the referenced data model. Also, the information could be limited if the required data involved two or more documents and their relationships.

Future work includes the execution of distinct workflows in order to test and improve the power of the proposed program. It is also proposed to add the data and documents to cloud information, execute the project in a cloud environment, retrieve the cloud provenance data and, once again, compare the performance for each model. In addition, these studies may include an analysis of other data provenance information that is relevant to researchers and their work and comparison between different NoSQL databases.

References

1. J. Han, E. Haihong, G. Le, and J. Du, "Survey on nosql database," in *Pervasive computing and applications (ICPCA), 2011 6th international conference on.* IEEE, 2011, pp. 363–366.
2. E. Erturk and K. Jyoti, "Perspectives on a big data application: What database engineers and it students need to know," *Engineering, Technology & Applied Science Research*, vol. 5, no. 5, pp. 850–853, 2015.
3. T. Li, L. Liu, X. Zhang, K. Xu, and C. Yang, "Provenancelens: Service provenance management in cloud," *10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2014.
4. A. Moniruzzaman and S. A. Hossain, "Nosql database: New era of databases for big data analytics-classification, characteristics and comparison," *arXiv preprint arXiv:1307.0191*, 2013.
5. D. G. Reis, F. S. Gasparoni, M. Holanda, M. Victorino, M. Ladeira, and E. O. Ribeiro, "An evaluation of data model for nosql document-based databases," in *World Conference on Information Systems and Technologies.* Springer, 2018, pp. 616–625.
6. R. Bellazzi, "Big data and biomedical informatics: a challenging opportunity," *Yearbook of medical informatics*, vol. 9, no. 1, p. 8, 2014.
7. F. Gessert and N. Ritter, "Scalable data management: Nosql datastores in research and practice," 2016.

8. The MongoDB 4.0 Manual. Accessed: 2018-06-23. [Online]. Available: https://docs.mongodb.com/manual
9. P. Buneman, S. Khanna, and T. Wang-Chiew, "Why and where: A characterization of data provenance," in *International conference on database theory.* Springer, 2001, pp. 316–330.
10. V. Guimaraes, F. Hondo, R. Almeida, H. Vera, M. Holanda, A. Araujo, M. E. Walter, and S. Lifschitz, "A study of genomic data provenance in nosql document-oriented database systems," in *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on.* IEEE, 2015, pp. 1525–1531.
11. M. Mattoso, C. Werner, G. H. Travassos, V. Braganholo, and L. Murta, "Gerenciando experimentos cientficos em larga escala," *SEMISH Seminrio Integrado de Software e Hardware*, 2008.
12. R. De Paula, M. Holanda, L. S. Gomes, S. Lifschitz, and M. E. M. Walter, "Provenance in bioinformatics workflows," *BMC bioinformatics*, vol. 14, no. 11, p. S6, 2013.
13. M. Abdrabo, M. Elmogy, G. Eltaweel, and S. Barakat, "Enhancing big data value using knowledge discovery techniques," *IJ Information Technology and Computer Science*, vol. 8, pp. 1–12, 2016.
14. Z. D. Stephens, S. Y. Lee, F. Faghri, R. H. Campbell, C. Zhai, M. J. Efron, R. Iyer, M. C. Schatz, S. Sinha, and G. E. Robinson, "Big data: astronomical or genomical?" *PLoS biology*, vol. 13, no. 7, p. e1002195, 2015.
15. M. Mattoso, J. Dias, F. Costa, D. de Oliveira, and E. Ogasawara, "Experiences in using provenance to optimize the parallel execution of scientific workflows steered by users," in *Workshop of Provenance Analytics*, vol. 1, 2014.
16. S. Kanwal, F. Z. Khan, A. Lonie, and R. O. Sinnott, "Investigating reproducibility and tracking provenance–a genomic workflow case study," *BMC bioinformatics*, vol. 18, no. 1, p. 337, 2017.
17. F. Costa, V. Silva, D. De Oliveira, K. Ocaña, E. Ogasawara, J. Dias, and M. Mattoso, "Capturing and querying workflow runtime provenance with PROV: a practical approach," in *Proceedings of the Joint EDBT/ICDT 2013 Workshops.* ACM, 2013, pp. 282–289.
18. F. Hondo, P. Wercelens, W. da Silva, I. Lima, I. Santana, G. de Araujo, A. Araujo, M. E. Walter, M. Holanda, and S. Lifschitz, "Uso de bancos de dados nosql para gerenciamento de dados em workflow de bioinformática," in *Proceedings of 32nd Brazilian Symposium on Databases*, 2017, pp. 310–317.
19. F. Hondo, P. Wercelens, W. da Silva, K. Castro, I. Santana, M. E. Walter, A. Araujo, M. Holanda, and S. Lifschitz, "Data provenance management for bioinformatics workflows using nosql database systems in a cloud computing environment," in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM).* IEEE, 2017, pp. 1929–1934.
20. D. Kim, B. Langmead, and S. L. Salzberg, "Hisat: a fast spliced aligner with low memory requirements," *Nature methods*, vol. 12, no. 4, p. 357, 2015.
21. H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, and R. Durbin, "The sequence alignment/map format and samtools," *Bioinformatics*, vol. 25, no. 16, pp. 2078–2079, 2009.
22. S. Anders, P. T. Pyl, and W. Huber, "Htseqa python framework to work with high-throughput sequencing data," *Bioinformatics*, vol. 31, no. 2, pp. 166–169, 2015.