AligNET: A Web Application for Mapped NGS Reads using Open-Source Genome Browser Libraries

Jamie Mari O. Ciron and Mylah Rystie U. Anacleto

Abstract— **Next-generation sequencing (NGS) refers to technology used to determine the composition of genomic data, which is essential in medical and evolutionary biology. However, generated reads provide little information on their own, and require to be aligned onto a reference through mapping. Mapped reads may be viewed through genome browsers, which are software that provide an overview of the genome and allows for annotation and analysis. While conventional genome browsers allow for a more extensive examination, displaying genomic data in a comprehensive manner is a bottleneck given the complexity of these types of data as well as technological limitations such as storage, memory, and screen size. The study has developed an application that specializes in visualizing mapping results, which not only simplifies the user interface but also lessens the risk for errors in analysis. Existing genome browser libraries were also integrated to provide access to reliable and well-maintained visualization features. The application was evaluated by groups of biology students and bioinformaticians from the University of the Philippines Los Baños (UPLB) using the System Usability Scale (SUS), resulting in scores of 89.00 and 95.00 respectively which are both considered above average.**

I. INTRODUCTION

A. Background of the study

The Human Genome Project (HGP) is defined by the National Human Genome Research Institute [1] as the first sequence of the human genome. Specifically, this refers to identifying the genes that make up the genome and analyzing the chemical composition of the genetic material [2]. The project is considered a massive feat as it provided the foundation for the study of human genomics and medicine. Understanding the genetic makeup of an individual can provide a model for the study of existing diseases and disorders, as well as the development of personalized medicine. The HGP was accomplished with the help of technology, particularly sequencing technology, which has provided a way for scientists to sequence thousands of human genomes per year [3].

Next-generation sequencing (NGS), according to Alekseyev et al. [4], refers to the technology that aids in extracting sequences and identifying variances in nucleic acid samples. The term next-generation is attributed from the emergence of NGS from Sanger sequencing which is the first-generation sequencing technology and the gold standard in nucleic acid sequencing methods. Compared to Sanger sequencing, NGS

offers better scalability, speed, and throughput [5] [6]. The general methodology of NGS involves extracting DNA or RNA and then fragmenting it to form a library. Polymerase Chain Reaction (PCR) then amplifies the fragments gathered to increase the fragment quantity and thus improve sequencing reliability. The resulting fragments then go through sequencing, which determines the order of nucleotide bases of each fragment, resulting in shorter segments called reads.

Genomic data, such as NGS reads, are used as inputs in a software called the genome browser. Genome browsers contain a graphical interface that allows users to view and examine genomic data easily and quickly [7]. According to Tao et al. [8], technologies such as genome browsers help humans perceive and understand biological information better by creating an overview of the whole genome and simplifying the pattern recognition process. Specifically, presenting the composition of the genome and the position of genes provides a way for biologists to evaluate functional relationships with other organisms [9]. As stated by Nusrat et al. [10] in their study, genomic visualization also bridges the gap between highly complex sequencing outputs and the cognitive abilities of researchers. Aside from genome visualization, genome browsers contain many features, such as gene annotation, variance calling, and sequence alignment.

Sequence alignment involves arranging two or more sequences to identify their similarity, which may be affected by evolutionary change [11]. Since NGS reads are generally short, ranging from 25 to 100 base pairs, they are aligned onto a reference genome, which helps detect the presence of mutations such as Single Nucleotide Polymorphism (SNP) and indels (insertions and deletions) [12] [13]. In humans, mutations indicate a predisposition to disease or a drug reaction; in animals and plants, mutations provide information regarding genetic linkages and diversity [14].

One common problem with existing genome browsers is that they are data-specific and unadaptable to other genome assemblies [15]. According to Mistadi [9], creating a high-level and easily comprehensible visual representation of genomic data also poses a challenge due to the inherent complexity and large size of these data. Another challenge with existing genome browsers is the lack of customization options and restricted data access, a problem with proprietary genome browsers since access to the code and data is limited.

Based on a study by Wang et al. [7], organizations provide

high-quality and updated annotation and genomic data through their genome browsers. An example is the CLC Genomics Workbench. However, presenting a large amount of information may present a challenge as genomic data increases in complexity [16]. Open-source libraries can help create customized genome browsers with more specialized functions to provide users a simpler and user-friendly alternative. An example is igv.js, which is an embeddable genome visualization component developed by Robinson et al. [17]. The igv.js component allows for easily visualizing the genome without the need for a back-end server, external dependencies, and data pre-processing. Open-source applications also address problems with data privacy since the source code will be made available to its users.

Developing an open-source application that specializes in displaying mapped NGS reads, with the help of existing genome browser libraries, may help address the aforementioned research gaps by maximizing the use of existing features and frameworks. The open-source nature of the study may also provide a starting point for developing similarly specialized applications by encouraging collaboration and customization.

B. Statement of the problem

Existing genome browsers are usually capable of displaying mapped reads; however, they also contain other features that make the software overly complex. According to Medina et al. [18], the efficiency of existing genome browsers is affected by the increasing amount of information brought about by emerging sequencing technologies. Developing a web application using open-source genome browser libraries that can display mapped sequences may help address these challenges. It may also be a starting point for developing other related applications.

C. Objectives of the study

The general objective of the study is to develop a web application that enables its users to map their NGS reads to available references using open-source genome browser libraries. Specifically, this study aims:

- to integrate a built-in library containing genome references;
- to develop a visual interface for accessing and navigating mapping results;
- to integrate existing open-source genome browser libraries; and
- to implement user system features, such as log-in and sign-up, in order to protect sensitive user data

D. Significance of the study

Analyzing mapped reads helps researchers gain information regarding the organism and its genomic makeup. However, conventional genome browsers are usually overloaded with features, making them difficult to use and inaccessible for

researchers. According to Pavlopoulos et al. [16] as cited by Mistadi [9], visualizing genomic data must be done in a coherent and concise way to improve data analysis and avoid errors. Existing genome browsers also tend to focus on a specific dataset, making them inefficient for other types of NGS reads [15]. Developing a web application specializing in visualizing any type of mapped reads will help users who wish to focus on studying variances and the composition of a genome.

As stated by Wang et al. [7], developing an application for browsing genomes is time and labor consuming especially if done from scratch. The application will thus integrate open-source genome browser libraries to optimize development and provide a comprehensive user interface with reliable features. Given the open-source nature of the web application, the study may serve as a starting point for implementing improvements or creating similar applications.

The application will also benefit research groups handling genomic data by ensuring accessibility through deployment and promoting collaboration through additional features such as file sharing.

E. Scope and limitations of the study

The study will focus on developing a web application primarily for visualizing aligned reads and the corresponding reference genome. As such, the assessment of the application will focus on usability, specifically the implementation and functionality of features, over other factors such as performance.

The study will consist of two phases: Special Problem (SP) 1, which spans from September 2023 until January 2024, and SP 2, which spans from February 2024 until May 2024. The topic writing and proposal will be done during SP 1, while the development and user testing of the web application will be done during SP 2.

The application will utilize genome references and mapped reads that are publicly accessible online in the implementation of the reference library and in user testing. Access to any other files imported by users will only be accessible to the researcher within the time frame of the study.

F. Date and place of study

The entire study will be conducted from September 2023 to May 2024 at the Institute of Computer Science (ICS) in collaboration with the Department of Physics, Institute of Mathematical Sciences and Physics (IMSP) through the Computational Interdisciplinary Research Labs (CINTERLABS). Both institutes, together with CINTERLABS, are from the College of Arts and Sciences (CAS), UPLB.

II. REVIEW OF RELATED LITERATURE

Numerous studies have shown the importance of aligned next-generation sequencing (NGS) reads to reference, especially in expression analysis and variance calling. However,

given the complexity of genomic data, the need for developing a user-friendly visualization software remains relevant today. The following review of related literature aims to provide an overview of next-generation sequencing technology and sequence alignment, as well as the current state of existing visualization tools and software.

A. Nucleic acid sequences and next-generation sequencing

Nucleic acids are naturally occurring chemical compounds found in living organisms, which are essential in storing and expressing genomic information [19]. Nucleic acids are present in two main types: deoxyribonucleic acid (DNA), which is responsible for the transmission of genetic information, and ribonucleic acid (RNA), which helps in protein synthesis. The main component of nucleic acids is called a nucleotide, which comprises a sugar molecule connected to a phosphate group and nitrogenous bases. The sequence formed by the nucleotide bases determines how cells develop and function and the traits the organism expresses.

Next-generation sequencing (NGS) is a high-throughput technology that allows for determining the sequences of nucleotides within nucleic acids, as well as the variances within these sequences. The term next-generation comes from the emergence of NGS from Sanger sequencing technology, which is the first-generation. Sanger sequencing was able to sequence the first part of the human genome and the first complete bacterial genome [20], which is why it is considered the gold standard for sequencing technologies. First-generation sequencing greatly influenced the field of comparative genomics and metagenomics. However, it was labor-intensive and expensive due to being limited to past technologies. While Sanger sequencing can only sequence one DNA fragment at a time, NGS can sequence thousands to millions of DNA molecules with a single machine run [21]. Aside from speed, NGS also offers better accuracy in identifying variances, making it more effective in terms of cost and time.

Although there are various NGS platforms, they all share the same general methodology which involves template preparation, sequencing and imaging, and data analysis [5]. The first step, template preparation, involves building a library from extracted nucleic acids. The result of template preparation is then amplified using Polymerase Chain Reaction (PCR) to create more copies of the library and to allow for simultaneous sequencing within multiple regions of the sample [22]. The amplified library then goes through sequencing to obtain the sequence of nucleotides in each fragment, which produces raw sequence data, also referred to as NGS reads.

B. Alignment of NGS reads to a reference genome

Data analysis in NGS involves processing and interpreting the sequencing data generated. One method used for examining these data is read mapping. Canzar and Salzberg [23] explain that since NGS produces many short reads containing limited information on their own, they must be mapped onto a reference, which contains the ideal nucleotide sequence of the organism. Read mapping is a type of sequence alignment that involves determining the position of reads relative to the reference, and then mapping these reads to identify regions of similarity. While sequence alignment could be used on DNA, RNA, and protein sequences, read mapping on the other hand usually involves DNA and RNA reads processed by NGS.

According to Mount [24], sequence alignment is essential in understanding the functional, structural, and evolutionary relationship between sequences. Differences between two sequences indicate genomic variances, which manifest in the form of Single Nucleotide Polymorphisms (SNPs) [13]. According to Seal et al. [14], the presence of SNPs and other mutations in humans indicates a predisposition to disease or a drug reaction. Understanding a person's genetic makeup also provides information regarding their lifestyle and medical history. The field of precision medicine benefits from sequence alignment since it involves the development of preventive medicine and treatment strategies, which take into account individual variances [12].

In animals and plants, mutations provide information regarding genetic linkages and diversity, making sequence alignment crucial in the study of phylogenetics. Wiltgen [11] describes sequence alignment as identifying evolutionary relationships between sequences, providing a way to quantify the evolutionary distance between two organisms. Similarity between sequences suggests the possibility of a shared ancestor and contributes in the classification of species and studies on biodiversity.

Technology plays a role in improving the process of sequence alignment through the use of algorithms and the development of mapping software. These tools make use of computational power to produce more accurate alignments efficiently. However, there are still challenges when it comes to read mapping, given the restrictions of existing technology regarding resources and performance. In a study by Trapnell and Salzberg [13], there are two main challenges when mapping short reads: how quickly NGS reads can be aligned to the reference and how a program can handle repeats in the reference.

C. Short-read mapping algorithms

Computers are able to perform mapping by representing sequences using strings of symbols consisting of letters, digits, and other types of characters [25]. These strings are used as input parameters in a mapping algorithm. An algorithm is a term commonly used in computer science to describe a step-by-step process applied to an input to produce an output [26]. In the context of sequence alignment, a mapping algorithm uses NGS reads and a reference genome as inputs, and alignment information and quality scores as outputs. Given the complexity of genomic data, various mapping algorithms have been developed to adapt to different types of genomes and to address problems in sequence alignment. In a study by Kim et al. [27], short-read alignment algorithms are categorized into optimal alignment, which results in the best alignment, and heuristic, which outputs near-best alignments.

Optimal alignment algorithms use dynamic programming, which is a method of simplifying complex problems by solving smaller subproblems and combining the obtained solutions to solve the initial problem. Eddy [28] explains in his study that dynamic programming is essential in solving the alignment problem since there are around $\frac{2^2 N}{\sqrt{2\pi N}}$ possible alignments given two sequences of length N. Given that the problem statement for optimal alignment algorithms is to find the overall best alignment, this problem is broken down into smaller subproblems by finding the most optimal alignment within shorter versions of the sequences. This concept is called recursion, which is the core concept of dynamic programming. While dynamic programming guarantees the best alignment, it is highly complex and computationally demanding since it retains the alignment scores for each subproblem solved. Given the lengths M and N of sequences x and y, the complexity of the algorithm is $O(MN)$, which means that the computation time grows quadratically as the sequence lengths increase.

Heuristic algorithms, on the other hand, are designed to provide nearly correct answers, or a solution for some subproblems to improve computational efficiency [29]. A heuristic approach relies on approximating or allowing a certain degree of error. In read mapping, this is done through the seed-and-extend approach. This method involves identifying short regions of high similarity, called seeds, using only portions of the reads [30]. Extending refers to using optimal algorithms on the seeded locations to identify potential candidates for the optimal alignment. The seed-and-extend method is useful when dealing with variances and with dealing with repeats in the reference genome.

D. Existing genome browsers and open-source development

Numerous tools and software, both proprietary and open-source, have been developed for mapping NGS reads to reference. Organization-owned genome browsers are able to provide a large variety of high-quality genomic data and annotations [7]. An example of a proprietary genome browser is the CLC Genomics Workbench developed by QIAGEN, which contains various tools that aid in pre-processing NGS data, genome and transcriptome assembly, and interactive visualization [31]. While the application is feature-rich, its full use is locked behind a paywall. Another example of a proprietary genome browser is the Genome Data Viewer by the National Center for Biotechnology Information (NCBI). The NCBI also holds a large database containing reference sequences, annotated genomic transcripts, and protein sequence records [32]. Easy access to the NCBI database allows for a more extensive analysis. The NCBI Genome Data Viewer is free for use; however there is no option to upload your own genomic data for comparative analysis. Generally, closed-source software lacks flexibility. This confines users to only the provided features and limits customization. In a study by Payne [33], he describes that proprietary software restricts its users the freedom to additional rights and the ability to modify the code, which may lead to security consequences. This poses a threat to data privacy especially in the context of human genomic data. While existing genome browsers contain plenty of features for extensive data analysis, this creates a challenge for visualizing a high volume of data. Displaying genomic data through a comprehensive interface is a bottleneck especially in developing applications for analyzing and visualizing these data due to limitations such as storage, memory, and screen size [16].

On the other hand, the term open-source refers to freely available software wherein anyone is allowed to contribute to its development and improvement [34]. The development of an open-source application for genomic data addresses potential security issues since the source could be made publicly accessible. This also encourages collaboration in the maintenance and improvement of the program. Focusing on the visualization of mapped reads rather than including all features from conventional genome browsers may also improve usability by simplifying the user interface.

The integration of open-source libraries may also help improve the accessibility and reliability of the program. For example, the Genomic Interaction Visualization Engine (GIVE) developed by Cao et al. [35] aims to allow its users to build a custom genome browser even without programming experience. This library improves the process of data visualization by incorporating memory management techniques to optimize the transfer of data between the hosting website and the internet browser. Another example is the igv.js, which is an embeddable genome visualization component developed by Robinson et al. [17]. The igv.js component allows for easily visualizing the genome without the need for a back-end server, external dependencies, and data pre-processing.

III. MATERIALS AND METHODS

A. Development Tools

The web application was developed on a laptop with the Windows 11 operating system and the following specifications:

- Processor: AMD Ryzen 5 5625U (12 CPUs), 2.3GHz
- Graphics: AMD Radeon Graphics
- Memory: 16GB
- Storage: 512GB SSD

The application was developed using Visual Studio Code (VSCode) as its integrated development environment (IDE) and Windows Subsystem for Linux (WSL) to access Linux command-line tools and utilities. The technology stack used consists of Django (v.5.0.1), NextJS (v.14.1.0) with TailwindCSS, and SQLite. Django is an open-source Python web framework that will be used for developing and implementing the server side of the application, known as the back-end. Python is a high-level programming language known for its readability and standard library that provides access to existing modules and packages for better ease of use. NextJS is a React framework that will be used to design the user interface (UI) or the front-end. TailwindCSS is an open-source CSS framework that provides predefined and customizable components to simplify the process of creating and designing the user interface. SQLite, a portable database included in Python will handle, store, and retrieve data using structured query language (SQL).

The application incorporated open-source libraries and modules such as Biopython, a Python library for computational biology; pysam, a Python module for reading and manipulating genomic data files; IGV.js, a JavaScript genome visualization component based on the Integrative Genomics Viewer (IGV); shadcn, a React component library; and NextAuth.js together with django-allauth for implementing Google SSO. The IGV.js component is licensed under the MIT License, allowing for the free use, modification, and distribution of the component provided that the copyright and permission notice will be included in the source code of the study.

B. Application Design

1) User flow diagrams

The following user flow diagrams demonstrate how the user will interact with the web application:
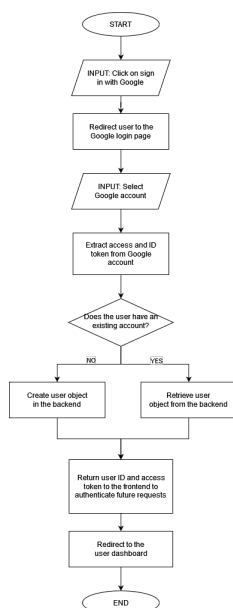
Fig. 1: User flow diagram for Google single sign-on (SSO)

2) User interface

The following screenshots illustrate the layout and features of the web application.

Fig. 2: User flow diagram for importing and viewing reference files
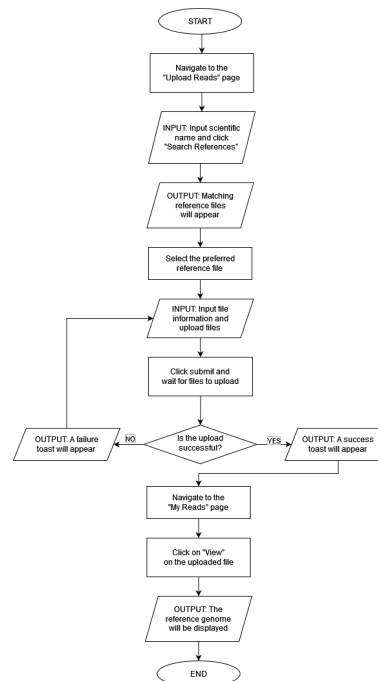
Fig. 3: User flow diagram for importing and viewing reads files

3) Application features
   a) Google Single Sign-on (SSO)
   Users may create an account by logging in through their Google accounts. The process of signing up and logging in is simplified with Google SSO, which reduces the need for inputting and remembering additional information such as username and password. Users with accounts will be granted
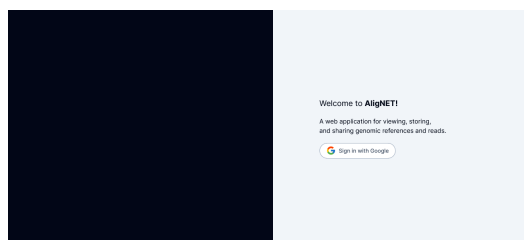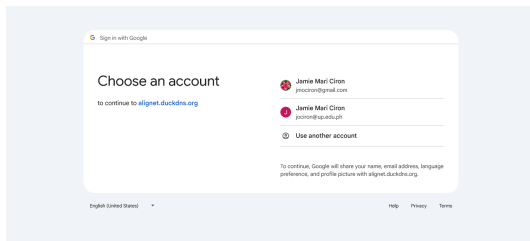

Fig. 4: Landing page

Fig. 5: Account select page
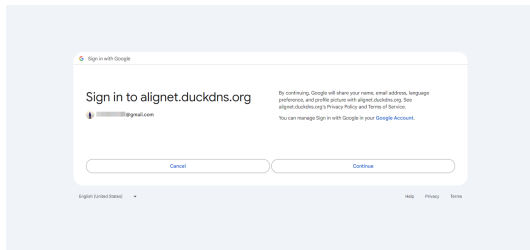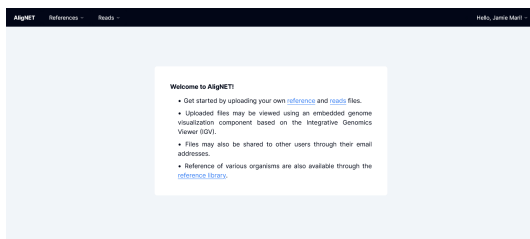

Fig. 6: Information consent page
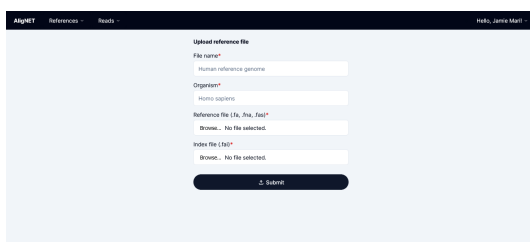

Fig. 7: User dashboard
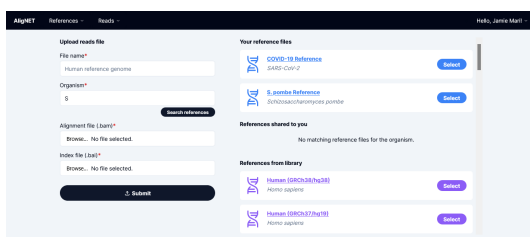

Fig. 8: Upload reference page
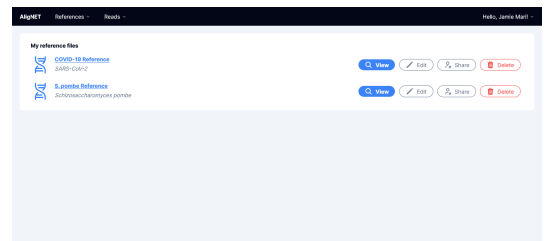

Fig. 9: Upload reads page

Fig. 10: My files page

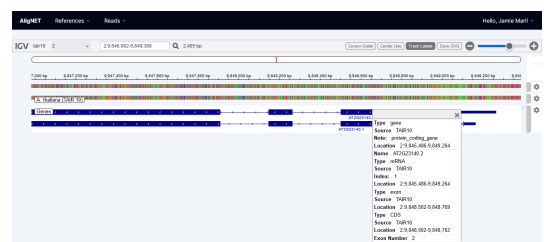Fig. 11: View reference page

Fig. 12: View reads page

Fig. 13: Viewing read annotations
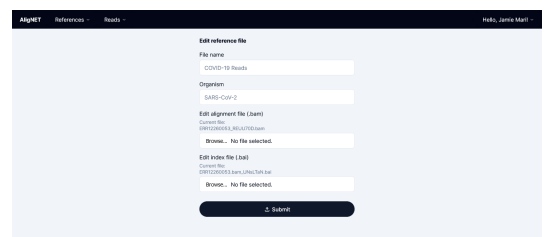

Fig. 14: Edit file page
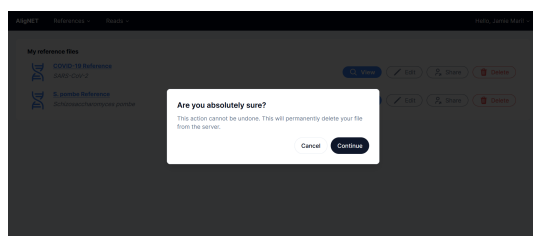
Fig. 15: Delete file modal
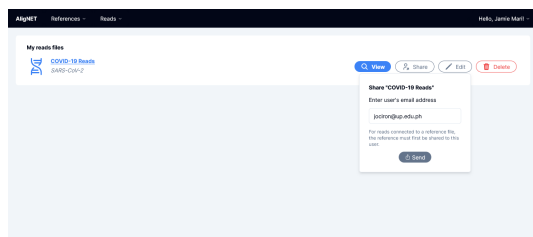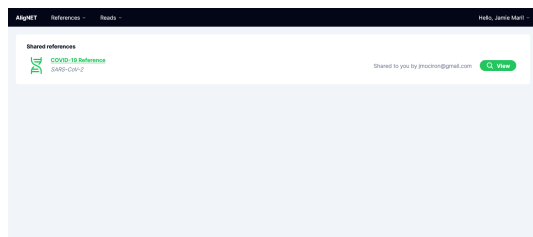

Fig. 16: File sharing modal
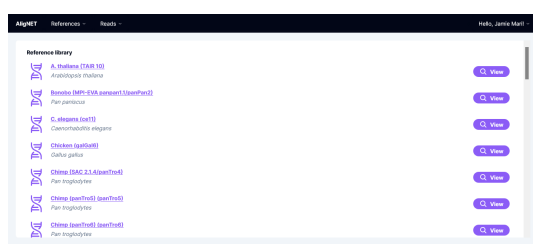

Fig. 17: Shared files page
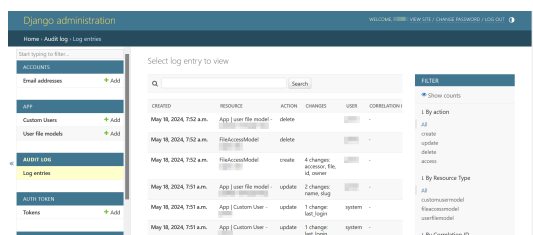

Fig. 18: Reference library page

Fig. 19: Audit logs page

access to all of the application's features, and may also upload and save their own genomic files.

b) Built-in reference library
The process of selecting a reference genome will be simplified by integrating a library of references from existing databases online. This minimizes the need to search online and to wait for long download times.

c) Import reference and reads files
Users may upload their own references and reads, and view them through the visualization component. The application will accept reference files in FASTA format, and reads files in the Binary Alignment Map (BAM) format which is catered for better efficiency and storage [36].

d) Genome visualization interface
Users may browse through the entire reference genome and the aligned reads. There will be tools for navigation and zooming, as well as dedicated windows with varying views, to enhance the browsing experience. If the file has annotations included, the user may click on the interface to view annotation details and other information.

e) File sharing
Imported files may be shared to other users through their email addresses. The shared files may be viewed on the recepients' accounts.

f) Audit logs
The admin may view changes in the database, specifically involving the user, file, and file access models. The logs display the type of action done (create, update, and delete), the timestamp, the user involved, and which fields are modified within the model.

C. Data Gathering and Testing

The reference library contains FASTA files hosted by the developers of IGV through Amazon S3. Sample files were retrieved from NCBI Reference Sequence (RefSeq), a collection of comprehensive and well-annotated sequences [37], and the NCBI Sequence Read Archive (NCBI-SRA), a public repository containing high-throughput sequencing data and alignment information [38].

Testing was done on two groups of users: professionals in the field of microbiology and students under the BS Biology program in UPLB, majoring in any of the three fields: Genetics, Cell and Molecular Biology, and Microbiology. Sample files together with a video demonstration showing the functionality of all implemented features were provided to all test users. Additionally, they were provided the option to meet with the researcher via Zoom for a more hands-on and guided testing experience.

After testing, the test users were asked to answer a Google Form containing the System Usability Scale (SUS) to measure the website's usability. The results were then compiled to compute the SUS score.

Tester	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
T1	4	1	5	1	4	1	5	1	5	1	95.00

TABLE I: SUS distribution and mean scores from the group of professionals

Tester	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
T1	5	1	5	2	5	1	5	1	5	1	97.50
T2	5	1	5	2	5	1	5	1	4	1	95.00
T3	4	1	5	1	4	2	5	1	4	1	90.00
T4	5	3	4	2	5	1	3	2	3	3	72.50
T5	5	2	5	1	5	1	5	2	4	2	90.00

TABLE II: SUS distribution and mean scores from the group of students

IV. RESULTS AND DISCUSSION

The web application has been successfully deployed through Digital Ocean and nginx, and is accessible through the following link: http://alignet.duckdns.org/. The API endpoints may be viewed separately using the Swagger interface at this link: http://alignet.duckdns.org/swagger, while the admin interface can be accessed through this link: http://alignet.duckdns.org/admin.

The System Usability Scale (SUS) was used to determine the usability of the application within its target users. The SUS contained 10 questions, answerable with a Likert scale from Strongly Disagree (1) to Strongly Agree (5). The ratings from the users were converted to a percentile rank. The application must score at least 68 to be considered above average in terms of usability [39]. The following statements were used in the SUS:

1) I think that I would like to use this system frequently.
2) I found the system unnecessarily complex.
3) I thought the system was easy to use.
4) I think that I would need the support of a technical person to be able to use this system.
5) I found the various functions in this system were well integrated.
6) I thought there was too much inconsistency in this system.
7) I would imagine that most people would learn to use this system very quickly.
8) I found the system very cumbersome to use.
9) I felt very confident using the system.
10) I needed to learn a lot of things before I could get going with this system.

To compute for the SUS score, the score to each odd-numbered question was subtracted by 1 while the score to each even-numbered question was subtracted to 5. The sum of the converted scores per user is multiplied by 2.5 to convert the range of possible scores from 0 to 100. After computing for the SUS score of each user, the mean of all scores was calculated to determine the overall score. The application scored 95.00 from the group of professionals and 89.00 from the group of students, both of which are considered above average in terms of usability.

V. CONCLUSION

The study was able to accomplish its objectives evident in the implemented features in the web application. Specifically, the application allows its users to log-in with their Google accounts, upload genomic files, browse through a built-in reference library, and navigate through their imported files. Additional features such as file sharing and audit logs to encourage collaboration while maintaining user and file privacy have also been implemented. The study was also able to achieve its goal of providing accessibility through deployment and delivering a user-friendly experience, as supported by the computed SUS score of 95.00 and 89.00.

VI. RECOMMENDATION

It is strongly recommended to host the web application through a paid server with better virtual resources (CPU, RAM, and storage) to improve the overall performance of the website. Additionally, implementing progress indicators for file upload and download will better inform users about the status of their files especially given network and server limitations. In terms of the visualization features, testers have recommended implementing other layouts aside from the current linear view especially with longer genomic files. Future researchers may also implement more features related to file processing using Biopython and pysam.

REFERENCES

[1] N. H. G. R. Institute, "Glossary of Genetic Terms," 11 2023.
[2] F. Collins and L. Fink, "The Human Genome Project," 1995.
[3] F. Carrasco-Ramiro, R. Peiró-Pastor, and B. Aguado, "Human genomics projects and precision medicine," *Gene Therapy*, vol. 24, pp. 551–561, 8 2017.
[4] Y. O. Alekseyev, R. Fazeli, Y. Shi, R. K. Basran, T. A. Maher, N. S. Miller, and D. G. Remick, "A Next-Generation Sequencing Primer—How does it work and what can it do?," *Academic Pathology*, vol. 5, p. 2374289518766521, 1 2018.
[5] A. Grada and K. Weinbrecht, "Next-Generation Sequencing: Methodology and application," *Journal of Investigative Dermatology*, vol. 133, pp. 1–4, 8 2013.
[6] I. Illumina, "Next-Generation Sequencing (NGS) — Explore the technology," 2023.
[7] J. Wang, L. Kong, G. Gao, and J. Luo, "A brief introduction to web-based genome browsers," *Briefings in Bioinformatics*, vol. 14, pp. 131–143, 7 2012.
[8] Y. Tao, Y. Liu, C. Friedman, and Y. A. Lussier, "Information visualization techniques in bioinformatics during the postgenomic era," *Drug Discovery Today: Biosilico*, vol. 2, pp. 237–245, 11 2004.
[9] A. Mistadi, *Requirements of Modern Genome Browsers*. PhD thesis, Concordia University, 2015.
[10] S. Nusrat, T. Harbig, and N. Gehlenborg, "Tasks, techniques, and tools for genomic data visualization," *Computer Graphics Forum*, vol. 38, pp. 781–805, 6 2019.
[11] M. Wiltgen, *Algorithms for structure comparison and analysis: Homology Modelling of Proteins*. 1 2019.
[12] Y. Hao, J. Meehan, W. Tong, and H. Hong, "Alignment of short reads: a crucial step for application of Next-Generation sequencing data in precision medicine," *Pharmaceutics*, vol. 7, pp. 523–541, 11 2015.
[13] C. Trapnell and S. L. Salzberg, "How to map billions of short reads onto genomes," *Nature Biotechnology*, vol. 27, pp. 455–457, 5 2009.
[14] A. Seal, A. Gupta, M. Mahalaxmi, A. Riju, T. R. Singh, and V. Arunachalam, "Tools, resources and databases for SNPs and indels in sequences: a review," *International Journal of Bioinformatics Research and Applications*, vol. 10, p. 264, 1 2014.
[15] T. Röder, S. Oberhänsli, N. Shani, and R. Bruggmann, "OpenGenome-Browser: a versatile, dataset-independent and scalable web platform for genome data management and comparative genomics," *BMC Genomics*, vol. 23, 12 2022.

[16] G. A. Pavlopoulos, A. Oulas, E. Iacucci, A. Sifrim, Y. Moreau, R. Schneider, J. Aerts, and I. Iliopoulos, "Unraveling genomic variation from next generation sequencing data," *BioData Mining*, vol. 6, 7 2013.

[17] J. Robinson, H. Thorvaldsdottir, D. Turner, and J. Mesirov, "igv.js: an embeddable JavaScript implementation of the Integrative Genomics Viewer (IGV)," *Bioinformatics*, vol. 39, 1 2023.

[18] I. Medina, F. Salavert, R. R. Sánchez, A. De María, R. Alonso, P. Escobar, M. Bleda, and J. Dopazo, "Genome Maps, a new generation genome browser," *Nucleic Acids Research*, vol. 41, pp. W41–W46, 6 2013.

[19] N. H. G. R. Institute, "The Human Genome Project," 9 2023.

[20] A. Kozińska, P. Seweryn, and I. Sitkiewicz, "A crash course in sequencing for a microbiologist," *Journal of Applied Genetics*, vol. 60, pp. 103–111, 1 2019.

[21] A. T. Vincent, N. Derôme, B. Boyle, A. I. Culley, and S. J. Charette, "Next-generation sequencing (NGS) in the microbiological world: How to make the most of your money," *Journal of Microbiological Methods*, vol. 138, pp. 60–71, 7 2017.

[22] R. S. Goswami, *PCR Techniques in Next-Generation Sequencing*. 1 2016.

[23] S. Canzar and S. L. Salzberg, "Short read Mapping: an Algorithmic Tour," *Proceedings of the IEEE*, vol. 105, pp. 436–458, 3 2017.

[24] D. W. Mount, *Bioinformatics*. Cold Spring Harbor Laboratory Press, 1 2001.

[25] P. Gagniuc, *Algorithms in Bioinformatics*. 7 2021.

[26] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. 3 ed., 7 2009.

[27] J. Kim, M. Ji, and G. Yi, "A review on sequence alignment algorithms for short reads based on Next-Generation sequencing," *IEEE Access*, vol. 8, pp. 189811–189822, 1 2020.

[28] S. R. Eddy, "What is dynamic programming?," *Nature Biotechnology*, vol. 22, pp. 909–910, 7 2004.

[29] E. S. Donkor, N. T. K. D. Dayie, and T. Adiku, "Bioinformatics with basic local alignment search tool (BLAST) and fast alignment (FASTA)," *Journal of Bioinformatics and sequence Analysis*, vol. 6, pp. 1–6, 4 2014.

[30] G. J. Filion, R. Cortini, and E. Zorita, "Calibrating Seed-Based heuristics to map short reads with sesame," *Frontiers in Genetics*, vol. 11, 6 2020.

[31] M. Matvienko, "CLC Genomics Workbench: Tools for re-sequencing, transcriptomics analyses, and for de novo assembly," 2015.

[32] N. O'Leary, M. W. Wright, J. R. Brister, S. Ciufo, D. Haddad, R. McVeigh, B. Rajput, B. Robbertse, B. Smith-White, D. Ako-Adjei, A. Astashyn, A. Badretdin, Y. Bào, O. Blinkova, V. Brover, V. Chetvernin, J. Choi, E. Cox, O. Ermolaeva, C. Farrell, T. Goldfarb, T. Gupta, D. H. Haft, E. Hatcher, W. Hlavina, V. Joardar, V. K. Kodali, W. Li, D. Maglott, P. Masterson, K. M. McGarvey, M. R. Murphy, K. O'Neill, S. Pujar, S. H. Rangwala, D. Rausch, L. D. Riddick, C. L. Schoch, A. Shkeda, S. S. Storz, H. Sun, F. Thibaud-Nissen, I. Tolstoy, R. E. Tully, A. R. Vatsan, C. Wallin, D. Webb, W. Wu, M. Landrum, A. Kimchi, T. Tatusova, M. DiCuccio, P. Kitts, and K. D. Pruitt, "Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation," *Nucleic Acids Research*, vol. 44, pp. D733–D745, 11 2015.

[33] C. Payne, "On the security of open source software," *Information Systems Journal*, vol. 12, pp. 61–78, 1 2002.

[34] J. E. Stajich and H. Lapp, "Open source tools and toolkits for bioinformatics: significance, and where are we?," *Briefings in Bioinformatics*, vol. 7, pp. 287–296, 5 2006.

[35] X. Cao, Z. Yan, Q. Wu, A. Zheng, and S. Zhong, "GIVE: portable genome browsers for personal websites," *Genome Biology*, vol. 19, 7 2018.

[36] Z. R. International, "What are SAM BAM Files?," 11 2021.

[37] N. C. for Biotechnology Information, "About RefSeQ."

[38] N. C. for Biotechnology Information, "The Sequence Read Archive (SRA) Documenation."

[39] J. Sauro, PhD, "Measuring Usability with the System Usability Scale (SUS) – MeasuringU," 2 2011.