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

A Special Problem
Presented to the Faculty of
The Institute of Computer Science
University of the Philippines Los Baños

Jamie Mari O. Ciron
May 2024

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

by

Jamie Mari O. Ciron

In partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

Mylah Rystie U. Anacleto
Adviser

Date Signed

Maria Art Antonette D. Clariño
Director
Institute of Computer Science

Date Signed

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.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my adviser, Mylah Rystie Anacleto, for providing me with all the guidance and support I needed throughout the proposal and development of this special problem.

I would also like to recognize and give thanks to all of those who have contributed in this study: to my respondents for thoroughly testing the application and providing recommendations for improvement, to CINTERLABS for providing invaluable insights and continuous support over the course of the study, and to DOST-SEI for providing me with financial assistance which has greatly helped with the deployment of the application.

Lastly, I would like to express my deepest appreciation and gratitude to my friends and family, who have been my support system not only as I worked on my special problem, but also as I went through my college years. Thank you for giving me the strength and resolve to pull through all challenges and the motivation to keep moving forward.

TABLE OF CONTENTS

I. INTRODUCTION

A. Background of the Study

The Human Genome Project (HGP) is defined by the National Human Genome Research Institute (National Human Genome Research Institute, 2023b) 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 (Collins & Fink, 1995). 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 (Carrasco-Ramiro et al., 2017).

Next-generation sequencing (NGS), according to Alekseyev et al. (2018), 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 (Grada & Weinbrecht, 2013; Illumina, Inc., 2023). 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 (Wang et al., 2012). According to Tao et al. (2004), 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 (Mistadi, 2015). As stated by Nusrat et al. (2019) 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 (Wiltgen, 2019). 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) (Hao et al., 2015; Trapnell & Salzberg, 2009). In humans, mutations indicate a predisposition to disease or a drug reaction; in animals and plants, mutations provide information regarding genetic linkages and diversity (Seal et al., 2014).

One common problem with existing genome browsers is that they are data-specific and unadaptable to other genome assemblies (Röder et al., 2022). According to Mistadi (2015), 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. (2012), 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 (Pavlopoulos et al., 2013). 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. (2023). 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 is 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. (2013), 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:

1. to integrate a built-in library containing genome references;
2. to develop a visual interface for accessing and navigating mapping results;
3. to integrate existing open-source genome browser libraries; and
4. 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. (2013) as cited by Mistadi (2015), 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 (Röder et al., 2022). The development of a web application specializing in visualizing any type of mapped reads may help users who wish to focus on studying variances and the composition of a genome.

As stated by Wang et al. (2012), developing an application for browsing genomes is time and labor consuming especially if done from scratch. The study has thus integrated open-source genome browser libraries to the web application 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 has also benefited 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 focused on developing a web application primarily for visualizing aligned reads and the corresponding reference genome. As such, the assessment of the application focused on usability, specifically the implementation and functionality of features, over other factors such as performance.

The study was conducted in 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 were done during SP 1, while the development and user testing of the web application were done during SP 2.

The application utilized 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 were only accessible to the researcher within the time frame of the study.

F. Date and Place of Study

The entire study was 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 (National Human Genome Research Institute, 2023a). 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 (Kozińska et al., 2019), 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 (Vincent et al., 2017). 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 (Grada & Weinbrecht, 2013). 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 (Goswami, 2016). 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 (2017) 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 (2001), 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) (Trapnell & Salzberg, 2009). According to Seal et al. (2014), 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 (Hao et al., 2015).

In animals and plants, mutations provide information regarding genetic linkages and diversity, making sequence alignment crucial in the study of phylogenetics. Wiltgen (2019) 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 to 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 (2009), 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 (Gagniuc, 2021). 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 (Cormen et al., 2009). 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. (2020), 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 (2004) 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 (Donkor et al., 2014). 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 (Filion et al., 2020). 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 (Wang et al., 2012). 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 (Matvienko, 2015). 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 (O'Leary et al., 2015). 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 (2002), 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 (Pavlopoulos et al., 2013).

On the other hand, the term open-source refers to freely available software wherein anyone is allowed to contribute to its development and improvement (Stajich & Lapp, 2006). 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. (2018) 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. (2023). 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 was 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 was 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 handled, stored, and retrieved 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 Single Sign-On (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 is included in the source code of the study.

B. Application Design

1. User Flow Diagrams

The following user flow diagrams demonstrate how the user interacts with the web application:

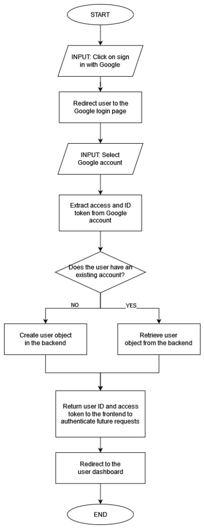

START
INPUT: Click on sign in with Google
Redirect user to the Google login page
INPUT: Select Google account
Extract access and ID token from Google account
Does the user have an existing account?
NO
YES
Create user object in the backend
Retrieve user object from the backend
Return user ID and access token to the frontend to authenticate future requests
Redirect to the user dashboard
END

Fig. 1: User flow diagram for Google SSO

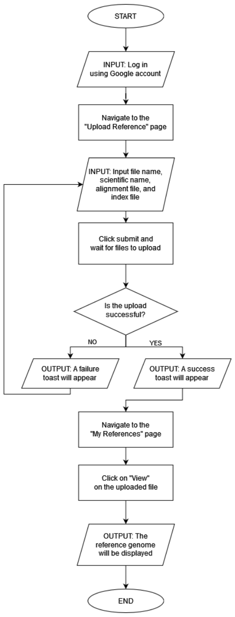

START
INPUT: Log in using Google account
Navigate to the "Upload Reference" page
INPUT: Input file name, scientific name, alignment file, and index file
Click submit and wait for files to upload
Is the upload successful?
NO
YES
OUTPUT: A failure toast will appear
OUTPUT: A success toast will appear
Navigate to the "My References" page
Click on "View" on the uploaded file
OUTPUT: The reference genome will be displayed
END

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

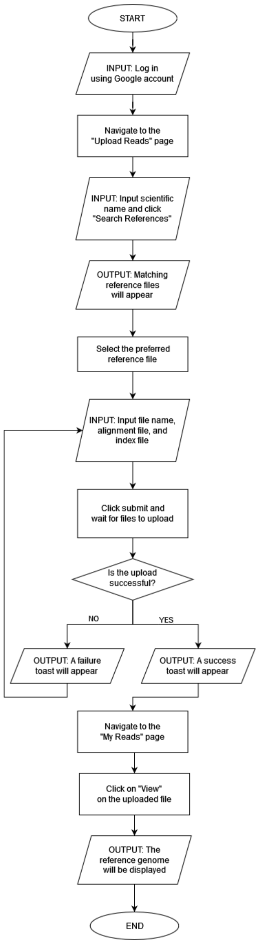

START
INPUT: Log in using Google account
Navigate to the "Upload Reads" page
INPUT: Input scientific name and click "Search References"
OUTPUT: Matching reference files will appear
Select the preferred reference file
INPUT: Input file name, alignment file, and index file
Click submit and wait for files to upload
Is the upload successful?
NO
YES
OUTPUT: A failure toast will appear
OUTPUT: A success toast will appear
Navigate to the "My Reads" page
Click on "View" on the uploaded file
OUTPUT: The reference genome will be displayed
END

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

2. 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 are granted 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 is 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 accepts reference files in FASTA format, and reads files in the Binary Alignment Map (BAM) format which is catered for better efficiency and storage (Zymo Research International, 2021).

d. Genome visualization interface

Users may browse through the entire reference genome and the aligned reads. There are tools made available 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 recipients' 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 (National Center for Biotechnology Information, n.d.-a), and the NCBI Sequence Read Archive (NCBI-SRA), a public repository containing high-throughput sequencing data and alignment information (National Center for Biotechnology Information, n.d.-b).

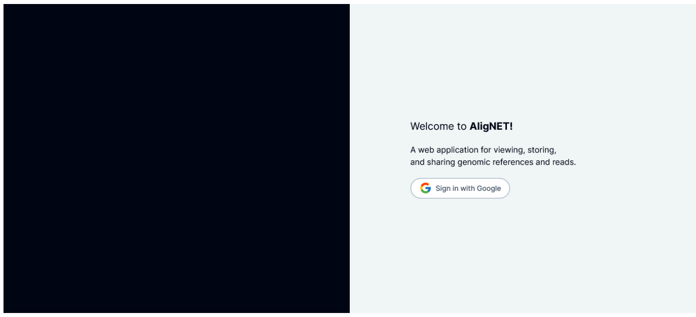
Testing was done on two groups of users: professionals in the field of microbiology and students under the BS Biology program in UPLB, majoring in any of the three fields: Genetics, Cell and Molecular Biology, and Microbiology. Sample files together with a video demonstration showing the functionality 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.

IV. RESULTS AND DISCUSSION

The web application was successfully deployed through Digital Ocean and nginx, and was made accessible through the following link: http://alignet.duckdns.org/. The API endpoints were made accessible with the Swagger interface at this link: http://alignet.duckdns.org/swagger, while the admin interface was made available through this link: http://alignet.duckdns.org/admin.

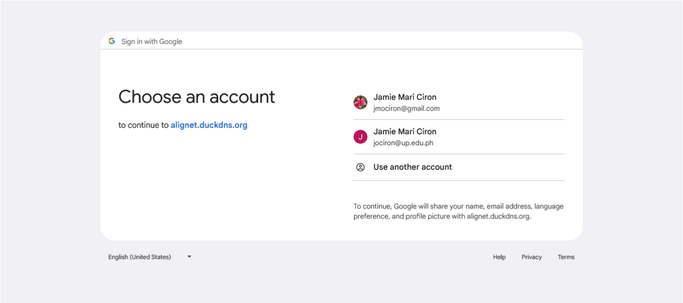
The deployment of AligNET allowed for better accessibility as it provided access to registered users through any web browser. Account creation was simplified through the implementation of Google SSO, which only requires users to log in using their Google accounts. If multiple Google accounts are logged in the same browser, users are asked to select which account to log onto AligNET. Upon successful login, the account is automatically created and the user is redirected to the user dashboard. Here, users are provided with a short overview on how to use AligNET, and are presented with the navigation bar. The navigation bar allows for quick access and redirection to all of the application's features.



Welcome to **AligNET!**

A web application for viewing, storing,
and sharing genomic references and reads.

Fig. 3: Landing page



Choose an account

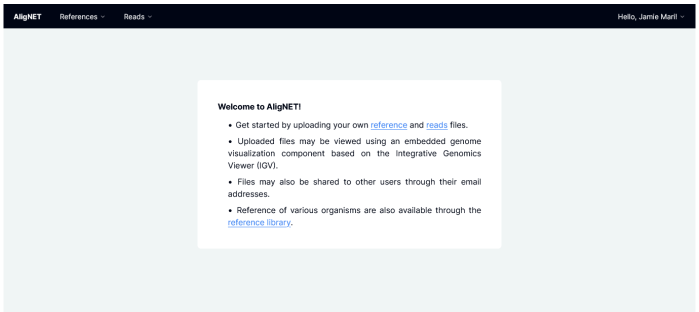
to continue to alignet.duckdns.org

Jamie Mari Ciron
jmociron@gmail.com

Jamie Mari Ciron
jociron@up.edu.ph

To continue, Google will share your name, email address, language preference, and profile picture with alignet.duckdns.org.

Fig. 4: Google account selection page

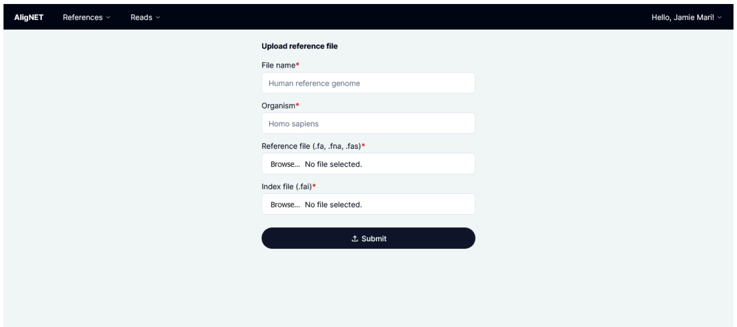


Welcome to AligNET!

- Get started by uploading your own reference and reads files.
- Uploaded files may be viewed using an embedded genome visualization component based on the Integrative Genomics Viewer (IGV).
- Files may also be shared to other users through their email addresses.
- Reference of various organisms are also available through the reference library.

Fig. 5: User dashboard and navigation bar

From the navigation bar, users may select whether to process or view reference or reads files first. Through the "References" menu, users can upload, modify, delete, and share their reference files with other users.



Upload reference file

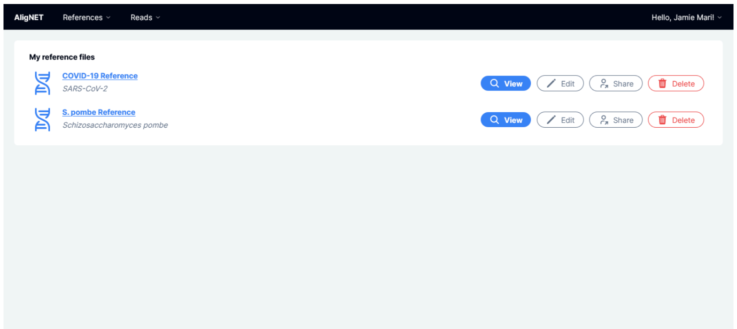
File name*
Human reference genome

Organism*
Homo sapiens

Reference file (.fa, .fna, .fas)*
Browse... No file selected.

Index file (.fai)*
Browse... No file selected.

Fig. 6: Upload reference page



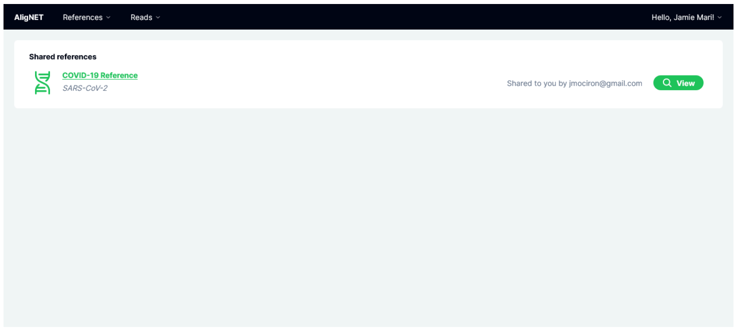
My reference files

COVID-19 Reference
SARS-CoV-2

S. pombe Reference
Schizosaccharomyces pombe

Fig. 7: My references page

File sharing is an added feature given the need for collaboration when conducting genomic analysis. Users may share their imported files to other users, and the recipients of these files will be granted the same viewing access as the owner of the file.

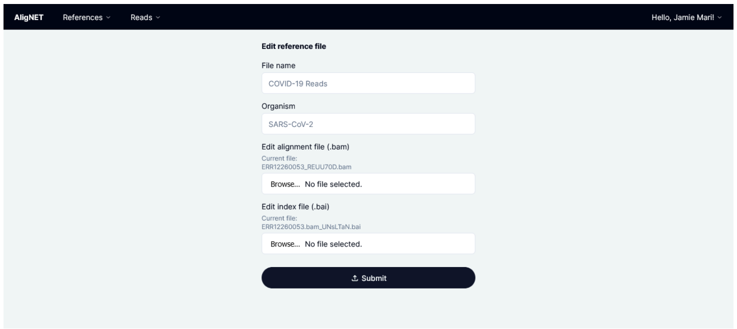


Shared references

COVID-19 Reference
SARS-CoV-2

Shared to you by jmociron@gmail.com

Fig. 8: Shared references page



Edit reference file

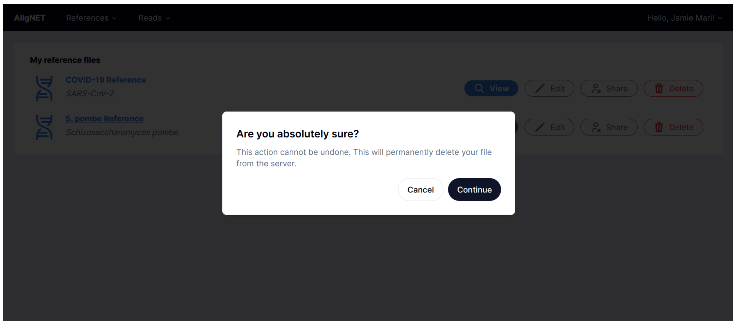
File name: COVID-19 Reads

Organism: SARS-CoV-2

Edit alignment file (.bam)
Current file:
ERR12260053_REUU70D.bam
No file selected.

Edit index file (.bai)
Current file:
ERR12260053.bam_UNsLTaN.bai
No file selected.

Fig. 9: Edit reference page



My reference files

COVID-19 Reference
SARS-CoV-2

S. pombe Reference
Schizosaccharomyces pombe

Are you absolutely sure?

This action cannot be undone. This will permanently delete your file from the server.

Fig. 10: Delete reference modal

Each reference file uploaded by or shared to the user may be viewed using the IGV.js visualization component. Through this, users may browse through the reference file and even customize their viewing experience.

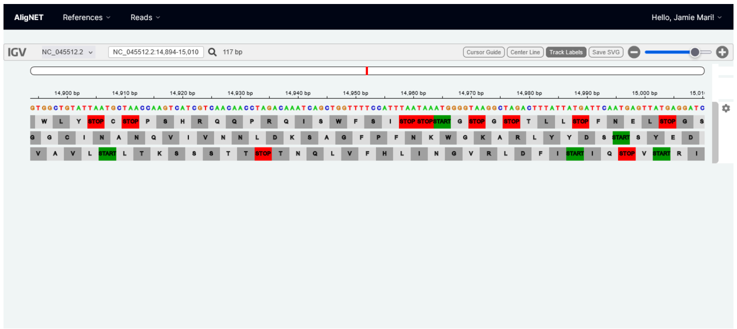
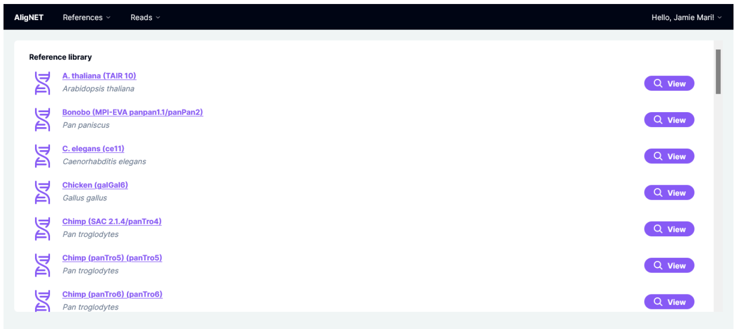

AligNET
References
Reads
Hello, Jamie Mari!
IGV
NC_045512.2
NC_045512.2:14,894-15,010
117 bp
Cursor Guide
Center Line
Track Labels
Save SVG
14,900 bp
14,910 bp
14,920 bp
14,930 bp
14,940 bp
14,950 bp
14,960 bp
14,970 bp
14,980 bp
14,990 bp
15,000 bp
15,01
GTGGCTGTATTAATGCTAACCAAGTCATCGTCAACAACCTAGACAAATCAGCTGGTTTTCCATTTAATAAATGGGGTAAGGCTAGACTTTATTATGATTCAATGAGTTATGAGGATC
W L Y STOP C STOP P S H R Q Q P R Q I S W F S I STOP STOP START G STOP G STOP T L L STOP F N E L STOP G S
G G C I N A N Q V I V N N L D K S A G F P F N K W G K A R L Y Y D S START S Y E D
V A V L START L T K S S S T T STOP T N Q L V F H L I N G V R L D F I START I Q STOP V START R I

Fig. 11: View reference page

The reference library is also available under the same menu, allowing users to work with various references sourced from IGV without having to separately browse and download them from other sources online.



Reference library

A. thaliana (TAIR 10)
Arabidopsis thaliana
View

Bonobo (MPI-EVA panpan1.1/panPan2)
Pan paniscus
View

C. elegans (ce11)
Caenorhabditis elegans
View

Chicken (galGal6)
Gallus gallus
View

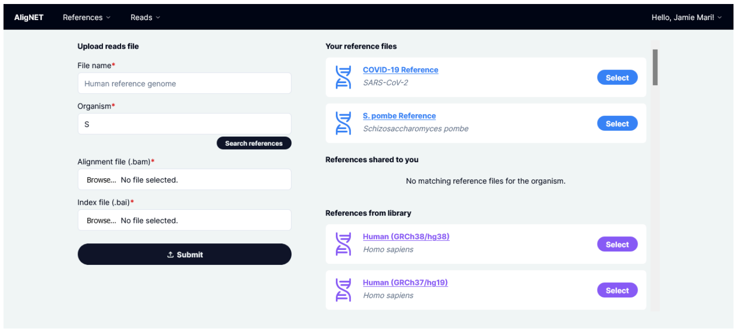
Chimp (SAC 2.1.4/panTro4)
Pan troglodytes
View

Chimp (panTro5) (panTro5)
Pan troglodytes
View

Chimp (panTro6) (panTro6)
Pan troglodytes
View

Fig. 12: Reference library page

The “Reads” menu enables users to edit, delete, and share their reads files, just like they can with reference files. However, the process of uploading reads is somewhat different from uploading references, as the website requires each set of reads to be linked to a corresponding reference file. This allows for the reads to be viewed and interpreted based on the connected reference. Users may choose from their uploaded or shared references, as well as from the references available in the library.



Upload reads file

File name*
Human reference genome

Organism*
S

Alignment file (.bam)*
Browse... No file selected.

Index file (.bai)*
Browse... No file selected.

Your reference files

COVID-19 Reference
SARS-CoV-2

S. pombe Reference
Schizosaccharomyces pombe

References shared to you

No matching reference files for the organism.

References from library

Human (GRCh38/hg38)
Homo sapiens

Human (GRCh37/hg19)
Homo sapiens

Fig. 13: Upload reads page

Similar to the reference files, reads files may also be viewed separately using the visualization component, which allows the user to navigate through the imported reads, view annotations, analyze alignments, and personalize their display.

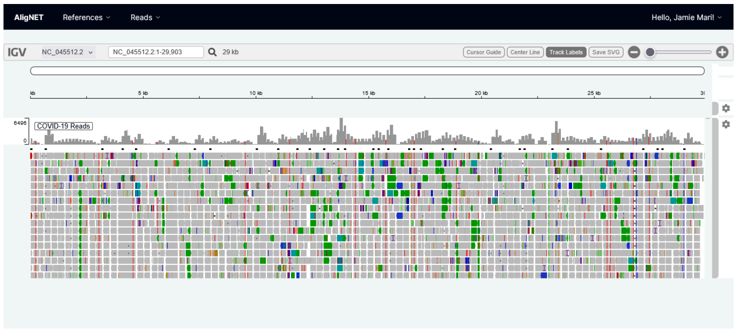


Fig. 14: View reads page

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 (Sauro, 2011). 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.

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

V. CONCLUSION

The study was able to address the need for an alternative to proprietary genome browsers through the integration of the IGV.js visualization component and the implementation of additional features to streamline the process of genomic analysis. Specifically, the web application allows its users to utilize a genome browser library through their browser and to import their files for future reference and for collaboration with other researchers. The application has also included a library for genomic references, simplifying the process of sourcing and references from various databases online. Additional features such as user log-in and sign-up together with audit logs have also been added to provide privacy and security. Overall, the study was able to achieve its goals 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. Creating a help center or Frequently Asked Questions (FAQ) page is also recommended to better orient new users on how to use and maximize each feature in the application. 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

National Human Genome Research Institute. (2023b, November 14). *Glossary of Genetic Terms*. Retrieved November 16, 2023, from https://www.genome.gov/genetics-glossary

Collins, F., & Fink, L. (1995). *The Human Genome Project.* PubMed. https://pubmed.ncbi.nlm.nih.gov/31798046/

Carrasco-Ramiro, F., Peiró‐Pastor, R., & Aguado, B. (2017). Human genomics projects and precision medicine. *Gene Therapy, 24*(9), 551–561. https://doi.org/10.1038/gt.2017.77

Alekseyev, Y. O., Fazeli, R., Shi, Y., Basran, R. K., Maher, T. A., Miller, N. S., & Remick, D. G. (2018). A Next-Generation Sequencing Primer—How does it work and what can it do? *Academic Pathology, 5*, 2374289518766521. https://doi.org/10.1177/2374289518766521

Grada, A., & Weinbrecht, K. (2013). Next-Generation Sequencing: Methodology and application. *Journal of Investigative Dermatology, 133*(8), 1–4. https://doi.org/10.1038/jid.2013.248

Illumina, Inc. (2023). *Next-Generation Sequencing (NGS) | Explore the technology.* https://www.illumina.com/science/technology/next-generation-sequencing.html

Wang, J., Kong, L., Gao, G., & Luo, J. (2012). A brief introduction to web-based genome browsers. *Briefings in Bioinformatics, 14*(2), 131–143. https://doi.org/10.1093/bib/bbs029

Tao, Y., Liu, Y., Friedman, C., & Lussier, Y. A. (2004). Information visualization techniques in bioinformatics during the postgenomic era. *Drug Discovery Today: BIOSILICO*, 2(6), 237-245. https://doi.org/10.1016/S1741-8364(04)02423-0

Mistadi, A. (2015). *Requirements of modern genome browsers.* Spectrum: Concordia University Research Repository. https://spectrum.library.concordia.ca/id/eprint/980113/

Nusrat, S., Harbig, T., & Gehlenborg, N. (2019). Tasks, techniques, and tools for genomic data visualization. *Computer Graphics Forum, 38*(3), 781–805. https://doi.org/10.1111/cgf.13727

Wiltgen, M. (2019). Algorithms for structure comparison and analysis: Homology Modelling of Proteins. In *Encyclopedia of Bioinformatics and Computational Biology* (pp. 38–61). https://doi.org/10.1016/b978-0-12-809633-8.20484-6

Hao, Y., Meehan, J., Tong, W., & Hong, H. (2015). Alignment of short reads: a crucial step for application of Next-Generation sequencing data in precision medicine. *Pharmaceutics, 7*(4), 523–541. https://doi.org/10.3390/pharmaceutics7040523

Trapnell, C., & Salzberg, S. L. (2009). How to map billions of short reads onto genomes. *Nature Biotechnology, 27*(5), 455–457. https://doi.org/10.1038/nbt0509-455

Seal, A., Gupta, A., Mahalaxmi, M., Riju, A., Singh, T. R., & Arunachalam, V. (2014). Tools, resources and databases for SNPs and indels in sequences: a review. *International Journal of Bioinformatics Research and Applications, 10*(3), 264. https://doi.org/10.1504/ijbra.2014.060762

Röder, T., Oberhänsli, S., Shani, N., & Bruggmann, R. (2022). OpenGenomeBrowser: a versatile, dataset-independent and scalable web platform for genome data management and comparative genomics. *BMC Genomics, 23*(1). https://doi.org/10.1186/s12864-022-09086-3

Pavlopoulos, G. A., Oulas, A., Iacucci, E., Sifrim, A., Moreau, Y., Schneider, R., Aerts, J., & Iliopoulos, I. (2013). Unraveling genomic variation from next generation sequencing data. *BioData Mining, 6*(1). https://doi.org/10.1186/1756-0381-6-13

Robinson, J., Thorvaldsdottir, H., Douglass, Turner, & Mesirov, J. (2023). IGV.js: an embeddable JavaScript implementation of the Integrative Genomics Viewer (IGV). *Bioinformatics, 39*(1), btac830. https://doi.org/10.1093/bioinformatics/btac830

Medina, I., Salavert, F., Sánchez, R. R., De María, A., Alonso, R., Escobar, P., Bleda, M., & Dopazo, J. (2013). Genome Maps, a new generation genome browser. *Nucleic Acids Research, 41*(W1), W41–W46. https://doi.org/10.1093/nar/gkt530

National Human Genome Research Institute. (2023a, September 7). *The Human Genome Project*. https://www.genome.gov/human-genome-project

Kozińska, A., Seweryn, P., & Sitkiewicz, I. (2019). A crash course in sequencing for a microbiologist. *Journal of Applied Genetics, 60*(1), 103–111. https://doi.org/10.1007/s13353-019-00482-2

Vincent, A. T., Derôme, N., Boyle, B., Culley, A. I., & Charette, S. J. (2017). Next-generation sequencing (NGS) in the microbiological world: How to make the most of your money. *Journal of Microbiological Methods, 138*, 60–71. https://doi.org/10.1016/j.mimet.2016.02.016

Goswami, R. S. (2016). PCR Techniques in Next-Generation Sequencing. In *Methods in molecular biology* (pp. 143–151). https://doi.org/10.1007/978-1-4939-3360-0_13

Canzar, S., & Salzberg, S. L. (2017). Short read Mapping: an Algorithmic Tour. *Proceedings of the IEEE, 105*(3), 436–458. https://doi.org/10.1109/jproc.2015.2455551

Mount, D. W. (2001). *Bioinformatics: Sequence and Genome Analysis*. Cold Spring Harbor Laboratory Press.

Gagniuc, P. (2021). *Algorithms in Bioinformatics*. https://doi.org/10.1002/9781119698005

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). The MIT Press. http://103.62.146.201:8081/xmlui/bitstream/handle/1/3490/Introduction.to.Algorithms.3rd.Edition.Sep.2010.pdf?sequence=1&isAllowed=y

Kim, J., Ji, M., & Yi, G. (2020). A review on sequence alignment algorithms for short reads based on Next-Generation sequencing. *IEEE Access, 8,* 189811–189822. https://doi.org/10.1109/access.2020.3031159

Eddy, S. R. (2004). What is dynamic programming? *Nature Biotechnology, 22*(7), 909–910. https://doi.org/10.1038/nbt0704-909

Donkor, E. S., Dayie, N. T. K. D., & Adiku, T. (2014). Bioinformatics with basic local alignment search tool (BLAST) and fast alignment (FASTA). *Journal of Bioinformatics and Sequence Analysis, 6*(1), 1–6. https://doi.org/10.5897/ijbc2013.0086

Filion, G. J., Cortini, R., & Zorita, E. (2020). Calibrating Seed-Based heuristics to map short reads with sesame. *Frontiers in Genetics, 11*. https://doi.org/10.3389/fgene.2020.00572

Matvienko, M. (2015). *CLC Genomics Workbench: Tools for re-sequencing, transcriptomics analyses, and for de novo assembly.* ResearchGate. https://www.researchgate.net/profile/Marta_Matvienko/publication/271444645_Genomics_Workbench_and_other_products_Qiagen_Bioinformatics_Workshop_at_PAG_2015/links/54c7dd610cf238bb7d0b9724/Genomics-Workbench-and-other-products-Qiagen-Bioinformatics-Workshop-at-PAG-2015.pdf

O'Leary, N., Wright, M. W., Brister, J. R., Ciufo, S., Haddad, D., McVeigh, R., Rajput, B., Robbertse, B., Smith-White, B., Ako-Adjei, D., Astashyn, A., Badretdin, A., Bào, Y., Blinkova, O., Brover, V., Chetvernin, V., Choi, J., Cox, E., Ermolaeva, O., . . . Pruitt, K. D. (2015). Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research, 44*(D1), D733–D745. https://doi.org/10.1093/nar/gkv1189

Payne, C. (2002). On the security of open source software. *Information Systems Journal, 12*(1), 61–78. https://doi.org/10.1046/j.1365-2575.2002.00118.x

Stajich, J. E., & Lapp, H. (2006). Open source tools and toolkits for bioinformatics: significance, and where are we? *Briefings in Bioinformatics, 7*(3), 287–296. https://doi.org/10.1093/bib/bbl026

Cao, X., Yan, Z., Wu, Q., Zheng, A., & Zhong, S. (2017). GIVE: toward portable genome browsers for personal websites. *bioRxiv* (Cold Spring Harbor Laboratory). https://doi.org/10.1101/177832

Zymo Research International. (2021, November 23). *What are SAM & BAM Files?* https://zymoresearch.eu/blogs/blog/what-are-sam-and-bam-files

National Center for Biotechnology Information. (n.d.-a). *About RefSeQ.* https://www.ncbi.nlm.nih.gov/refseq/about/

National Center for Biotechnology Information. (n.d.-b). *The Sequence Read Archive (SRA) Documenation*. https://www.ncbi.nlm.nih.gov/sra/docs/

Sauro, J., PhD. (2011, February 3). *Measuring Usability with the System Usability Scale (SUS) – MeasuringU*. https://measuringu.com/sus/