INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 8 ISSUE 3 (2022) PAGES 26 - 36
RECEIVED :09.05.2021 PUBLISHED : 06.06.2022

June 06.2022

MINI REVIEW ON JAVA PROGRAMMING AND BIOJAVA.

Venkataramanan.S, Dinesh Davagandhi, Philemon Telajan

Faculty of Health and Life Sciences, Management and Science University, Seksyen 13,

41000, Shah Alam, Selangor, Malaysia

*Corresponding author's email: s_venkataramanan@msu.edu.my

ABSTRACT

Java is an object-oriented programming language with class-based program that requires little implementation requirements thus allowing compiled java code to execute on any Java-enabled systems without the need to recompile. Java programmes will be compiled into a bytecode, which can be executed on any program that runs on a Java virtual machine (JVM). In the field of bioinformatics, the application of java programming has made a significant impact which is apparent with its implementation in Biojava. Biojava is a free software initiative that aims to supply java programming tools for biological data processing. Biojava is a set of java library methods capable of handling a wide range of data, from DNA and protein sequences to 3D protein structures. The Biojava libraries may be used to automate a variety of common bioinformatics operations, such as processing a Protein Data Bank (PDB) file, interfacing with Jmol, and so on.

Keywords *:* Java programming, Biojava, DNA, protein sequences, bioinformatics

1.INTRODUCTION

Because it is a fairly organised language, java is a suitable computer language for bioinformatician novices. Biojava is a set of java modules for bioinformatics projects, and it is the world's biggest collection of bioinformatics programming tools. Despite its speed and ubiquity, java lacks the flexibility of python or perl. Regardless, Java is one of the greatest beginning languages for bioinformatics researchers who are new to the field [Ninh Laboratory of Computational Biology, 2013].

Java is an object-oriented programming language and software platform that is utilised on billions of devices, including laptop computers, mobile devices, game consoles, medical equipment, and more. Java's principles and grammar are based on the C and C++ programming languages. The mobility of java-based applications is a significant benefit. It's fairly simple to migrate code from a notebook computer to a mobile device after you've developed it in java.["Design and Implementation of ECMAScript Interpreter for VXML Execution," 2003]. The major purpose of java when it was designed in 1991 by Sun Microsystems' James Gosling was being to be able to be only compiled once without the need to recompile every time in order to save resources and run the computer program more efficiently [TechTarget, 2002].

Furthermore, Javascript can only be used in web browsers, but java may be used anywhere. At a breakneck pace, new and improved software development tools are displacing monopolistic products that were once thought to be irreplaceable with java programming to endurance is astonishing in light of this constant turnover; more than two decades after its birth, it is still the most popular language for application software development, with developers preferring it to languages like Python, PHP, Swift, C++, and others [Ninh Laboratory of Computational Biology, 2013]. Thus, java continues to be a crucial prerequisite for job seekers especially the ones with computer science backgrounds [Education, 2021].

Javascript is an ECMAScript-compliant high-level, frequently compiled language where it contains Text, dates, regular expressions, standard data structures, and the Document Object Model are all supported through APIs. There are no input/output (I/O) features in the ECMAScript standard, including graphics, storage and networks. In reality, Javascript APIs for I/O are provided by the web browser or another runtime system. Initially used only for internet browsers, but now Javascript engines are found in a wide range of servers and apps that are compatible with it ["Design and Implementation of ECMAScript Interpreter for VXML Execution," 2003].

Biojava is a preferred choice for technique and application as well as software development in the field of bioinformatics that incorporates java programming because of the resources offered and its cross-platform compatibility. Other well-known Bio projects, such as Bioperl and Biopython which provide an excellent scripting versatility, are included in the java programming language via the JVM-based scripting languages [TechTarget, 2002]. Biojava is a centralised code repository, whereas R/Bioconductor is a collection of decentralised packages that are produced and operated separately. The practicality of Biojava is inextricably linked to the programming language, which is always changing [Pocock et al., 2000].

Biojava is a long-running initiative with an active user and development community. The Open Bioinformatics Foundation is a public organisation without profit that is committed to promoting the practice and concept of open-source software development and open science within the biological research community and is one of the benefactors in the success of the Biojava project [Lafita et al., 2019]. The Biojava library's functionality has evolved over time, from primitives' language for popular biological file formats to cutting-edge tools for sequencing and structure analysis. The Biojava collection is divided into 11 modules that provide users with biological sequences, structures, and bioinformatics functions. Biojava employed automated integration, unit testing, and code review to achieve its software engineering goal hence Biojava is qualified for usage in essential bioinformatics pipelines, databases, and applications [Holland et al., 2008].

Besides that, another functionality within the Biojava for bioinformatics is its interfaces and algorithms for working with protein and nucleotide sequences are provided by the core module. Parsing sequences from local files and distant resources, file format conversion, and gene to protein translation are only a few of the features. This module serves as a foundation upon which additional modules can be built [Lafita et al., 2019]. For pairwise and multiple sequence alignments, the alignment module provides standard techniques and data formats. The structure module offers data formats and methods for parsing, manipulating, and comparing 3D structures of biological macromolecules, while the structure-gui module allows for structure display and alignment in Jmol. Other smaller modules provide more specialised capabilities for certain Bioinformatics disciplines. Genome data is handled via the genome module, which includes memory-efficient parsers for GTF, GFF2, GFF3, and FASTQ file formats [Holland et al., 2008].

2. DISCUSSIONS

IMPORTANCE OF JAVA PROGRAMMING

Java programming language was created specifically to be used for the Internet's dispersed setting. It was created with the C++ programming language in mind, nevertheless easier to navigate and enforce on an object-oriented programming language [R.Rory.,2022b]. Java may be used to build whole apps that can operate on a single machine or be spread across a network of servers and clients. It is also be used to create a tiny application model or applet that can be embedded in a Web page where Applets allow visitors to engage with the content of a Web site [Mercer, 2022].

Java is an object-oriented language, it is much more reliable, secure, and easy to learn as compared to other programming languages like C and C++, which adds to its broad usage and popularity. An object-oriented language makes tough issues manageable by providing more flexibility and practicality[Mercer, 2022]. Java is meant to allow the creation of portable, high-performance programmes for the greatest possible variety of computer systems, supporting the essential notions of encompassing accessibility and cross-platform interaction. Objects are created in java using classes that specify the data's attributes and the object's activities. Java features such as inheritance aid in additional code simplification [JAVA, n.d.]. Comparable but distinct sorts of objects can be made easier by inheriting similar features and actions from the parent class allowing for the use of common code while still allowing differentiation [Appslure.,2021].

Java is preferable to certain other programming languages because it is mature and continues to evolve[JAVA, n.d.]. It is a reliable programming language based on the natural world. Oracle Corporation, on the other hand, often modifies the programming language with the assistance of a thriving society. Each new version of Java brings with it a slew of new features and enhanced performance[Mercer, 2022]. Java is not just an ever-evolving platform with the choice to use a variety of Java APIs to achieve everyday improvement tasks exclusive of writing extra code., but it is also a stand-alone platform. This enables the development of application code once and publishes appliance code to several platforms with minimal effort with programmers can convert java code to bytecode and utilise the bytecode address across several platforms without having to reinstall the system [Team, 2021]. The rules of java syntax on the syntax of C and C++. As a result, novices find it simpler to learn and use it in a shorter period of time.[“Design and Implementation of ECMAScript Interpreter for VXML Execution,” 2003]. Java is a concurrent, class-based, and object-oriented programming language that may be used with a variety of IDEs, frameworks, and development tools. Developers may also choose from a variety of java libraries, frameworks, IDEs, and optimization tools. They may use powerful IDEs like NetBeans and Eclipse to produce understandable and high-quality java code. They can also take advantage of the IDEs' superior debugging capabilities and code finishing point features to keep the Java core maintainable [Team, 2021]. Research and scientific applications will require natural language processing where the utilisation of java is frequently the default choice. The fundamental reason being that it is safer, more portable, and more manageable than C++ or any other language, plus it has stronger integrated services tools.

JUnit is the single most significant contribution of open source to java and the broader field of programming. Smalltalk has already experimented with test-driven development (TDD). TDD, like many other language advances, did not gain significant notice and use until it was made accessible in java. TDD soared quickly from an experimental approach of a few programmers to the conventional way to create software in the twenty-first century when Kent Beck and Erich Gamma released JUnit in 2000.[Pedamkar, 2021]

3.APPLICATION OF JAVA PROGRAMMING

A java application is an independent platform software developed in java that runs whether on a client or a server [Pedamkar, 2021]. The JVM (Java Virtual Machine) executes all commands and runs the programme in the JRE (Java Runtime Environment), [Tyson, 2020] giving java programmes unfettered access to all of the computer system's resources [Pedamkar, 2021].

I. Application in Mobile

The most widely used java application is in the development of a mobile application, especially for mobile running the Android operating system. Several programmes written in the java programming language are used by almost all mobile phones [JAVA, n.d.-a]. Java is the official or basic programming language for developing mobile apps. It's available for Android Studio and Kotlin, which are both used to create mobile apps. Android utilises the Dalvik Virtual Machine (DVK) to execute class files, similar to how java does, where these Android files are packaged as Android Application Packages (APK). Due to java and its OOPs ideas, Android has a higher level of security and simplicity [Pedamkar, 2021].

II. Desktop GUI Applications

Java assists in the development of web services that are platform-independent and language-independent, meaning that an application written in another language might conveniently consume java web services. To support this, java provides Soap (Simple Object Access Protocol) and Rest based web services, each with its own set of benefits and drawbacks [Indeed Editorial Team, n.d.]. Soap has several limitations, although it is more secure than the others; nonetheless, it only accepts XML data formats, which necessitates a lot of convention during construction and the rest offers a variety of data types such as JSON, HTML, and XML, making it more demandable than soap, it uses less bandwidth during web service creation. [Pedamkar, 2021].

III. Web Applications

java is a computer language that allows for both strong security and basic and straightforward development. Java's two features entice web application developers to use it. Struts, Servlets, and JSPs are java paradigms that offer robust and effective support for web applications. The server-side components servlet and JSP assist in the development of the web application's business logic; Jsp is an extension of the servlet; it has more functionality than servlet and is easier to use than servlet due to tags similar to those found in HTML [JAVA, n.d.-a] It also has the capability of developing whatever programmes are necessary. Open-source e-commerce platforms, such as Broadleaf, make designing e-commerce apps a breeze [Pedamkar, 2021].

4. ROLE OF JAVA IN BIOINFORMATICS (Biojava)

The unique field of bioinformatics is seen to be an interdisciplinary field of science that is known to broaden the methods and software tools for analysing biological data, as well as vast and complicated data sets. Bioinformatics analysis entails a variety of tasks and processes that make use of a variety of tools designed for specific bioinformatics applications [Venkataramanan et al., 2016]. A typical bioinformatics application, for example, analysing DNA and protein, reading FASTA files, maintaining

DNA sequences in memory, performing various computing activities on the sequences, and then outputting the results to a file. Text mining and text parsing activities are also common in bioinformatics [Fourment & Gillings, 2008]. To do these activities and make them more efficient, algorithms can be designed using programming languages [Venkataramanan et al., 2016].

Java is a semi-compiled programming language that utilises automatic memory management. This program is compiled into intermediate-level code, which is then run by an interpreter or compiler, such as the Java Virtual Machine (JVM), during runtime. Java is a computer language that may be used to do bioinformatics computations. However, because of its limits and low performance, it is not particularly effective. It can't meet the demands of bioinformatics research, which is difficult to conduct using commercial programming languages such as java [Mangalam, 2002]. There are, however, open-source Bio toolkits dedicated to the processing of biological data. Biojava is an open-source software project that provides tools for processing biological data using the tools from the java programming language. Biojava introduces multiple modules that work independently to provide a framework for creating bioinformatics applications using the java language [Bhongale & Vyas, 2020]. Biojava's broad support for Basic procedures such as interpreting/analysing and modifying sequences and structures enable unique algorithm developers to focus solely on the specific challenge in the field of bioinformatics.

Biojava remains to be an open-source project that implements parsers, data structures, and algorithms to make bioinformatics analysis easier. Common engagements in the genomic study, structural biology, ontology, phylogenetics, and other fields are carried out using these implementations [Lafita et al., 2019]. Figure 1 shows how Biojava is connected with system biology and computational biology. This software, which was first released in the year 2000, concentrates on biological sequences and alignments. Biojava is currently a well-established project with over 65 distinct developers contributing to it. Biojava has embraced software engineering practices such as overtime continual integration, unit testing, and code review [Lafita et al., 2019]. These practices allow Biojava to be more suited for inclusion in bioinformatics databases, software and pipelines. In addition, Biojava provides tools that are utilised for the comparison of protein structure comparison, protein sequence alignment and the analysis of the protein modification as a disordered region [Bhongale & Vyas, 2020]. Figure 2 below shows the number of articles that had utilised the Biojava project from the year 2000 till 2018.

The Biojava library is a library The Biojava library is made up of a number of modules that may be easily accessible by users [Lafita et al., 2019]. Among the modules available are the core module, protein structure modules, genome and sequencing modules, alignment module, ModFinder module, amino acid characteristics module, and protein disorder module. These modules include cutting-edge tools for comparing protein structures, DNA and protein sequences, pairwise and multiple sequence alignments, detecting protein modifications, amino acid properties analysis, predicting disordered protein regions, and interpreters for common file formats, all constructed based on a biologically meaningful data model. The Core module, for instance, provides an interface and procedures that allow users to work on projects involving protein and nucleotide sequences [Prlic et al., 2012]. It also provides classes for modelling nucleotide and amino acid sequences, as well as their interactions. Processing sequences from local and remote files, file format conversion, and gene-to-protein translation are all features of the module [Lafita et al., 2019], unlike the protein structure model, which includes instruments for representing and manipulating 3D biomolecular structures. It particularly focuses on the protein structure comparison and contains an alignment algorithm that is both flexible and rigid body, the FATCAT algorithm.

The software now has a range of important and new capabilities that help with bioinformatics projects. The structure data model, structure file format, multiple sequence alignment, and other critical elements are among the key modifications [Lafita et al., 2019]. In terms of the structure data model, the Biojava software represents biological structures using a hierarchical data model that has been customised to closely reflect the data model provided in the PDBx/mmCIF dictionary. The software's structure module has been updated to include more functions such as reading, writing, and obtaining biological structures in the PDB, PDBx/mmCIF, and MMTF formats. This is beneficial to bioinformaticians since it allows for the representation of big structures and supports rich annotations as well as format conversion. A new custom procedure was implemented in Biojava using the org.Biojava.nbio.structure.align.multiple package which allows merging the result of the pairwise alignment algorithm, to generate multiple structure alignments and the outputs can be stored in an identical hierarchical structure format which is used by Biojava for the pairwise structure alignment algorithm [Lafita et al., 2019]. These are just a few of the Biojava modifications that have been made to increase its bioinformatics functionality and performance.

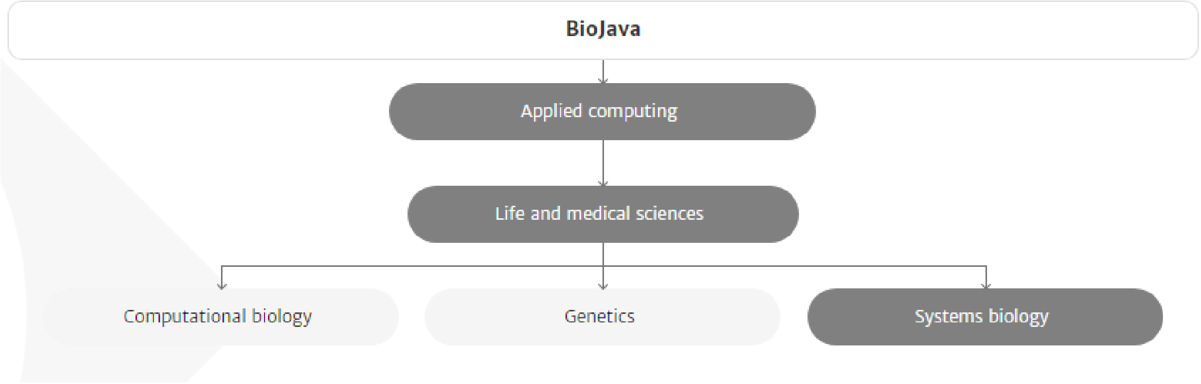


Figure 1 – Classification of Biojava

Source: Prlic, A., Yates, A., Bliven, S. E., Rose, P. W., Jacobsen, J., Troshin, P. V., Chapman, M., Gao, J., Koh, C. H., Foisy, S., Holland, R., Rimsa, G., Heuer, M. L., Brandstatter-Muller, H., Bourne, P. E., & Willis, S. (2012b). BioJava: an open-source framework for bioinformatics in 2012. Bioinformatics, 28(20), 2693–2695. https://doi.org/10.1093/bioinformatics/bts494

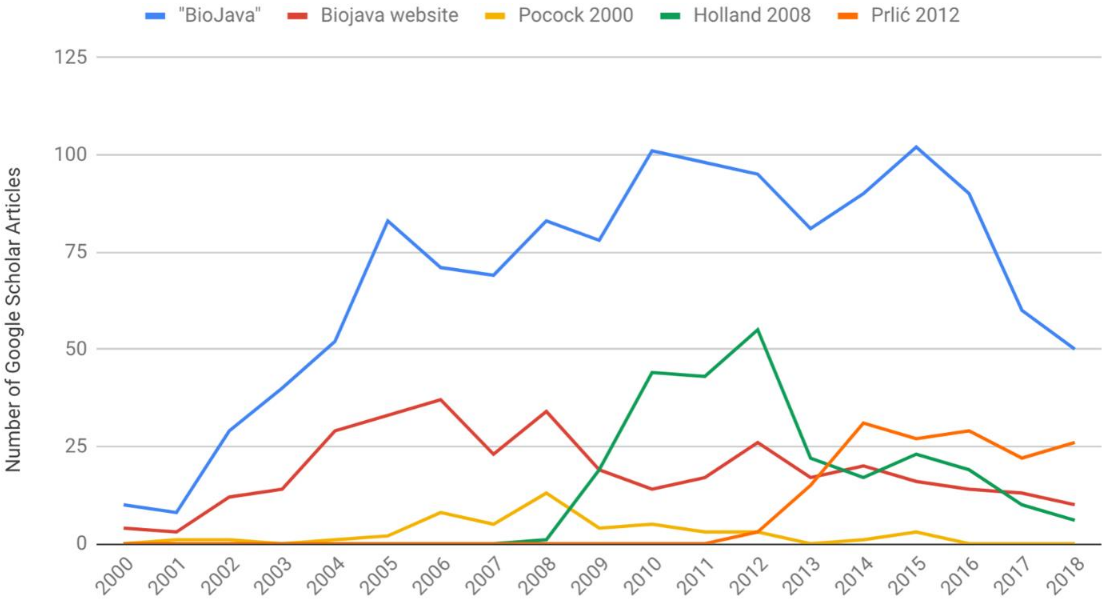


Figure 2 - Yearly count of the number of articles that utilised the Biojava project from 2000 to 2018

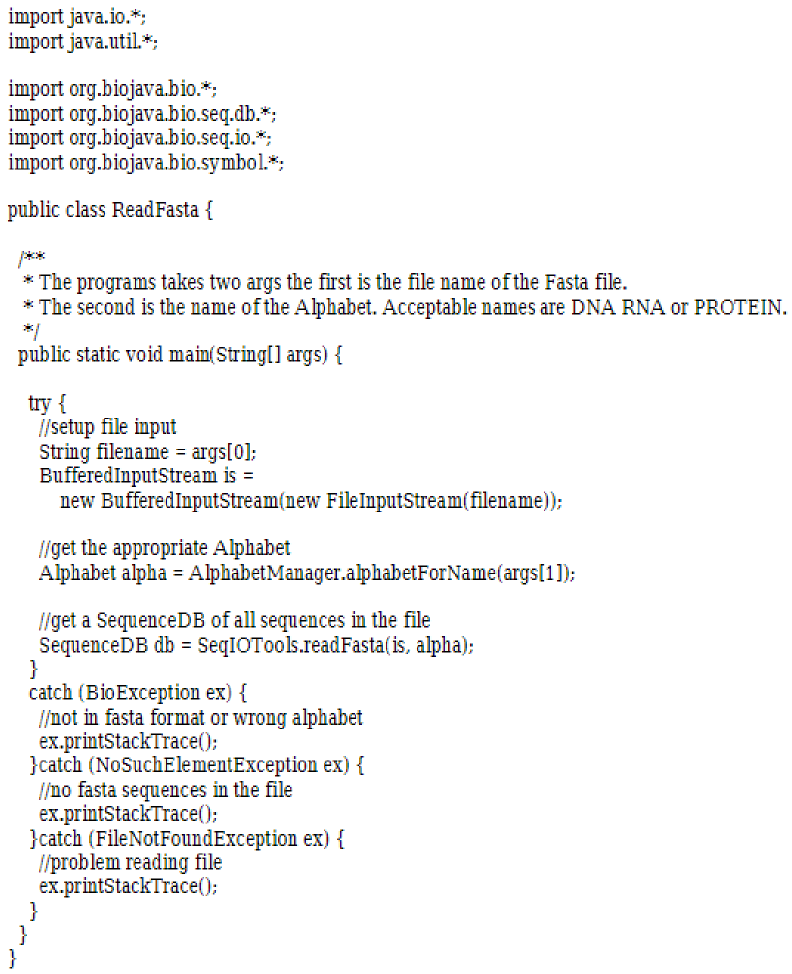
Source: Lafita, A., Bliven, S., Prlić, A., Guzenko, D., Rose, P. W., Bradley, A., Pavan, P., Myers-Turnbull, D., Valasatava, Y., Heuer, M., Larson, M., Burley, S. K., & Duarte, J. M. (2019b). BioJava 5: A community driven open-source bioinformatics library. PLOS Computational Biology, 15(2), e1006791. https://doi.org/10.1371/journal.pcbi.1006791

5. BIOJAVA INTEGRATION IN BIOINFORMATICS

The discipline of molecular biology has advanced dramatically over the previous few decades. Especially after the completion of the Human Genome Project (HGP), molecular biology study conducted in the research facility has also progressed in terms of technological advancement. As a result of this advancement, significant portions of the genomes of various species have been sequenced at a rapid pace. The field of Bioinformatics has significantly progressed by utilising advanced computational methods to solve biological problems [Venkataramanan et al., 2016]. The data commonly expressed in a bioinformatics problem consists of usually 2 major biological components; Gene Data and Protein Structures [Padole, Dr Mamta, 2005]. Sequence analysis in Bioinformatics, which comprises statistical outputs as a theoretical value and result, can be used to manipulate the gene and protein sequences [Venkataramanan et al., 2016]. These biological data undergo operations that would represent the gene data in computational forms. The operations include sequencing DNA structures, organising gene structures to form genomes, analysing genome data, predicting protein structures, describing protein dynamics, and modelling protein structures to unfold the protein structure all performed on biological data [Venkataramanan et al., 2016] [Padole, Dr Mamta, 2005].

Bioinformatics faces a number of obstacles as a field that evolves year after year. Databases, networks, search engines, data mining, pattern matching, data visualisation, modelling and simulations, approaches to bioinformatics and other aspects of information science comprise the challenges faced by the community [Padole, Dr Mamta, 2005]. Tool Building, which creates new programmes and methods to analyse and organise data, Tool Using, which uses existing programmes and data to solve biological problems, and Tool Maintenance, which creates databases, translates biological questions into a programming language, and keeps databases and tools updated, are the three approaches to Bioinformatics [Padole, Dr Mamta, 2005]. A technological tool that has made a significant contribution in the field of molecular biology that can be utilised to address the three bioinformatics methods is the Bio-program, Biojava.

Biojava is the most recent java programming language application development. Biojava is a feature from java that is beneficial in the development of bioinformatics-related applications. The latest version of Biojava is Biojava 6.0.5 which greatly extends its functionality and has fixed null handling in places related to loading PDB, CIF and mmtf files. Biojava is a full-featured programming language that may be used to create applications in a variety of bioinformatics fields. Furthermore, as an open-source technology, anyone is able to contribute to the development of Biojava by writing more packages providing several facilities that are related to the application of AI algorithms for generating gene sequences, gene data expression and capturing the data from a variety of file formats [Padole, Dr Mamta, 2005]. The following is an example of the Biojava code. The purpose of this program is to read the gene sequences from FASTA format files.



```
import java.io.*;
import java.util.*;

import org.biojava.bio.*;
import org.biojava.bio.seq.db.*;
import org.biojava.bio.seq.io.*;
import org.biojava.bio.symbol.*;

public class ReadFasta {

  /**
   * The programs takes two args the first is the file name of the Fasta file.
   * The second is the name of the Alphabet. Acceptable names are DNA RNA or PROTEIN.
   */
  public static void main(String[] args) {

    try {
      //setup file input
      String filename = args[0];
      BufferedInputStream is =
          new BufferedInputStream(new FileInputStream(filename));

      //get the appropriate Alphabet
      Alphabet alpha = AlphabetManager.alphabetForName(args[1]);

      //get a SequenceDB of all sequences in the file
      SequenceDB db = SeqIOTools.readFasta(is, alpha);
    }
    catch (BioException ex) {
      //not in fasta format or wrong alphabet
      ex.printStackTrace();
    }catch (NoSuchElementException ex) {
      //no fasta sequences in the file
      ex.printStackTrace();
    }catch (FileNotFoundException ex) {
      //problem reading file
      ex.printStackTrace();
    }
  }
}
```

Source: [Padole, Dr. Mamta. (2005). Biojava : Tool for BioInformatics.]

Biojava has been used in a number of bioinformatics research papers with great success. An example is a recent paper that was published by the Department of Laboratory Medicine, Seoul National

University Hospital on the 1st of March that used Biojava to was used to read sequence trace files [Kim et al., 2022]. The researchers have developed a computational method (called SnackNTM) using programming languages such as java – Biojava for the identification of nontuberculous mycobacteria (NTM) at the species level. The software, SnackNTM is a fully automated software that enables species-level identification of nontuberculous mycobacteria from trace files. The use of Biojava in the development of the software allows it to read multiple trace files which is the first step of the workflow of SnackNTM [Kim et al., 2022].

A different case where Biojava has aided the study is a protein structure search study that was published on February 22nd [van Kempen et al., 2022] This study revolves around fast and sensitive comparisons of large protein structure sets. The Foldseek open-source software reduces structural alignments to 3Di sequence alignments by encoding structures as sequences across the 20-state 3Di alphabet [van Kempen et al., 2022]. Biojava was implemented into the combinatorial extension (CE) alignment algorithm of the software. The modules of Biojava were utilised to calculate the CE values of the large structure sets.

There are many other projects in the field of bioinformatics which utilises Biojava. Some include the following software and web servers; Metabolic Pathway Builder, Dazzle, BioSense, Cytoscape, BioWeka, Jstacs, GenBeans and many more. To summarize the information that has been mentioned above, Biojava remains to be one of the largest open-source APIs for the development of bioinformatics software. It offers a wide range of tools that are used in many bioinformatics tasks and problems. The libraries and the modules of Biojava come with a comprehensive API for examining DNA, RNA, and proteins [Holland et al., 2008]. It includes cutting-edge algorithms for performing various calculations and a versatile platform for bioinformatics application development. As a result, we conclude that Biojava is a useful and important tool in bioinformatics.

6. CONCLUSION

Though it has been around for several decades, java programming is still one of the most widely used programming languages in a range of industries. This programming language is so powerful that it can be used to build practically anything, from small apps to web servers. As a result, java is readily incorporated into a wide range of software and tools and can be used in a wide range of fields. In addition to its application, the development of Biojava in java programming has made a big effect on the developing field of bioinformatics. Biojava is a well-established open-source project that supports a diverse set of bioinformatics applications. Recent Biojava work has centred on overcoming issues with complicated structural bioinformatics data. As a result, it can be argued that java has the broadest general programming support, with Biojava providing excellent support in the Bio sector.

REFERENCES

1. Education, I. C. (2021, April 12). *What is Java?* IBM. Retrieved April 26, 2022, from https://www.ibm.com/cloud/learn/java-explained
2. *Programming Languages of Bioinformatics*. (2013, February 24). Ninh Laboratory of Computational Biology. Retrieved April 26, 2022, from https://nlcb.wordpress.com/2013/02/23/programming-languages-of-bioinformatics-2/
3. *Wondering What Java Is? Here's the Answer*. (2019, July 3). ThoughtCo. Retrieved April 26, 2022, from https://www.thoughtco.com/what-is-java-2034117
4. Shu-yu Guo,Michael Ficarra, Kevin Gibbons. (2022). ECMAScript® 2023 Language Specification. *ECMAScript® 2023 Language Specification*. https://tc39.es/ecma262/#sec-overview
5. TechTarget. (2002, May 2). *Write once, run anywhere?* ComputerWeekly.Com. Retrieved April 26, 2022, from https://www.computerweekly.com/feature/Write-once-run-anywhere
6. Pocock, M., Down, T., & Hubbard, T. (2000). BioJava. *ACM SIGBIO Newsletter*, *20*(2), 10–12. https://doi.org/10.1145/360262.360266
7. Lafita, A., Bliven, S., Prlić, A., Guzenko, D., Rose, P. W., Bradley, A., Pavan, P., Myers-Turnbull, D., Valasatava, Y., Heuer, M., Larson, M., Burley, S. K., & Duarte, J. M. (2019). Biojava 5: A community driven open-source bioinformatics library. *PLOS Computational Biology*, *15*(2), e1006791. https://doi.org/10.1371/journal.pcbi.1006791
8. Holland, R. C. G., Down, T. A., Pocock, M., Prlic, A., Huen, D., James, K., Foisy, S., Drager, A., Yates, A., Heuer, M., & Schreiber, M. J. (2008). Biojava: an open-source framework for bioinformatics. *Bioinformatics*, *24*(18), 2096–2097. https://doi.org/10.1093/bioinformatics/btn397
9. R. (2022, April 26). *What is Java and why is it important?* Code Institute NL. Retrieved April 27, 2022, from https://codeinstitute.net/nl/blog/what-is-java/
10. Mercer, J. (2022, March 13). *Why is Java so popular for developers and programmers?* FRG Technology Consulting. Retrieved April 27, 2022, from https://www.frgconsulting.com/insights/why-is-java-so-popular-developers
11. Appslure (2021, December 27). *What Is The Importance Of Java Programming Language?* Appslure. Retrieved April 27, 2022, from https://www.appslure.com/what-is-the-importance-of-java-programming-language.
12. JAVA. (2022, April 8). *Java Platform Evolution - Dev.java*. Dev.Java: The Destination for Java Developers. Retrieved April 27, 2022, from https://dev.java/evolution/
13. Team, J. H. (2021, June 5). *What is Java and Why it is Important?* Java Assignment Help. Retrieved April 27, 2022, from https://www.javaassignmenthelp.com/blog/what-is-java-and-why-it-is-important/#what-are-the-three-main-components-of-java
14. Tyson, M. (2020, January 17). *What is the JRE? Introduction to the Java Runtime Environment*. InfoWorld. Retrieved April 27, 2022, from https://www.infoworld.com/article/3304858/what-is-the-jre-introduction-to-the-java-runtime-environment.html
15. Pedamkar, P. (2021, April 19). *Applications of Java*. EDUCBA. Retrieved April 27, 2022, from https://www.educba.com/applications-of-java/
16. JAVA. (n.d.). *Java Apps - Javatpoint*. Dev.Java: The Destination for Java Developers. Retrieved April 27, 2022, from https://www.javatpoint.com/java-apps.

17. Indeed Editorial Team. (2021, August 28). *What Is SOAP API? (And How It Differs From REST API)*. Indeed Career Guide. Retrieved April 27, 2022, from https://www.indeed.com/career-advice/career-development/what-is-soap-api
18. Venkataramanan, S., Sivakumar, C., & MD. JOHAR, P. D. M. G. (2016). <span>A prototype web application package for basic DNA and protein analysis using R language</span> Proceedings of MOL2NET 2016, International Conference on Multidisciplinary Sciences, 2nd Edition. https://doi.org/10.3390/mol2net-02-02003
19. Fourment, M., & Gillings, M. R. (2008). A comparison of common programming languages used in bioinformatics. *BMC Bioinformatics*, *9*(1). https://doi.org/10.1186/1471-2105-9-82
20. Mangalam, H. (2002). The Bio* toolkits -- a brief overview. *Briefings in Bioinformatics*, *3*(3), 296–302. https://doi.org/10.1093/bib/3.3.296
21. Bhongale, M. P., & Vyas, R. (2020). Role of Biojava in the Department of Bioinformatics Tools. *Advances in Bioengineering*, 29–49. https://doi.org/10.1007/978-981-15-2063-1_2
22. Andreas Prlić, Andrew Yates, Spencer E. Bliven, Peter W. Rose, Julius Jacobsen, Peter V. Troshin, Mark Chapman, Jianjiong Gao, Chuan Hock Koh, Sylvain Foisy, Richard Holland, Gediminas Rimša, Michael L. Heuer, H. Brandstätter–Müller, Philip E. Bourne, Scooter Willis, Biojava: an open-source framework for bioinformatics in 2012, Bioinformatics, Volume 28, Issue 20, 15 October 2012, Pages 2693–2695, https://doi.org/10.1093/bioinformatics/bts494
23. Padole, Dr. Mamta. (2005). Biojava : Tool for BioInformatics.
24. Kim, Y. G., Jung, K., Kim, S., Kim, M. J., Lee, J. S., Park, S. S., & Seong, M. W. (2022). SnackNTM: An Open-Source Software for Sanger Sequencing-based Identification of Nontuberculous Mycobacterial Species. *Annals of Laboratory Medicine*, *42*(2), 213–248. https://doi.org/10.3343/alm.2022.42.2.213
25. van Kempen, M., Kim, S. S., Tumescheit, C., Mirdita, M., Söding, J., & Steinegger, M. (2022). Foldseek: fast and accurate protein structure search. *Foldseek: Fast and Accurate Protein Structure Search*. https://doi.org/10.1101/2022.02.07.479398
26. Holland, R. C. G., Down, T. A., Pocock, M., Prlic, A., Huen, D., James, K., Foisy, S., Drager, A., Yates, A., Heuer, M., & Schreiber, M. J. (2008). Biojava: an open-source framework for bioinformatics. *Bioinformatics*, *24*(18), 2096–2097. https://doi.org/10.1093/bioinformatics/btn397