IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

ISO 3297:2007 Certified

Vol. 5, Issue 12, December 2017

Java Based Genome Sequence Detection

Mr.Shreenath Waramballi[1], Dr.Gurumurthy[2], Dr.Guruprakash C.D[3]

Assistant Professor, Department of Computer Science, Sahyadri College of Engineering, Mangalore, India [1]

Professor, Department of Biotechnology, GM Institute of Technology, Davangere, India[2]

Professor, Department of Computer Science, Sri Siddhartha Institute of Technology, Tumkur, India [3]

Abstract: Genome Sequencing is the way of figuring out the sequence of genome.The genome sequence will represent a valuable shortcut, helping scientists find genes much more easily and quickly. A genome sequence contains some clues about where genes are, even though scientists are just learning to interpret these clues.Scientists also hope that being able to study the entire genome sequence will help them understand how the genome as a whole works—how genes work together to direct the growth, development and maintenance of an entire organism.knowing the entire genome sequence will help scientists study the parts of the genome outside the genes.Java is a simple robust,effective programming language which has been used by many programers,industries,developers all over the world.Using the platform independent nature of java we can develop various applications according to the specification given by the user in an efficient manner.our idea is to use the java features to predict the genome sequence of various individuals.With the help of various genome sequences copied from EBI and NCBI the given input sequence is examined and figured out.

Keywords: Genome Sequence, Platform Independent, Java Feautures, Platform Independent, Javafeauture.

I. INTRODUCTION

A genome is an organism's complete set of DNA, including all of its genes. Each genome contains all of the znformation needed to build and maintain that organism. In humans, a copy of the entire genome—more than 3 billion DNA base pairs—is contained in all cells that have a nucleus. The main goals of the Human Genome Project were to provide a complete and accurate sequence of the 3 billion DNA base pairs that make up the human genome and to find all of the estimated 20,000 to 25,000 human genes. The Project also aimed to sequence the genomes of several other organisms that are important to medical research, such as the mouse and the fruit fly.In addition to sequencing DNA, the Human Genome Project sought to develop new tools to obtain and analyze the data and to make this information widely available. Also, because advances in genetics have consequences for individuals and society, the Human Genome Project committed to exploring the consequences of genomic research through its Ethical, Legal, and Social Implications (ELSI) program. **Biological databases** are libraries of life sciences information, collected from scientific experiments, published literature, high-throughput experiment technology, and computational analysis. They contain information from research areas including genomics, proteomics, metabolomics, microarray gene expression, and phylogenetics. Information contained in biological databases includes gene function, structure, localization clinical effects of mutations as well as similarities of biological sequences and structures. The National Center for Biotechnology Information advances science and health by providing access to biomedical and genomic information. The NCBI houses a series of databases relevant to biotechnology and biomedicine and is an important resource for bioinformatics tools and services. Major databases include GenBank for DNA sequences and PubMed, a bibliographic database for the biomedical literature. Other databases include the NCBI Epigenomics database. All these databases are available online through the Entrez search engine. Java is a programming language created by James Gosling from Sun Microsystems in 1991. The target of Java is to write a program once and then run this program on multiple operating systems. The first publicly available version of Java was released in 1995. Sun Microsystems was acquired by the Oracle Corporation in 2010. In 2006 Sun started to make Java available under the GNU General Public License (GPL). Oracle continues this project called OpenJDK. Java is defined by a specification and consists of a programming language, a compiler, core libraries and a runtime (Java virtual machine) The Java runtime allows software developers to write program code in other languages than the Java programming language which still runs on the Java virtual machine. The Java platform is usually associated with the Java virtual machine and the Java core libraries.

The Java language was designed with the following properties:

1. Platform independent: Java programs use the Java virtual machine as abstraction and do not access the operating system directly. This makes Java programs highly portable. A Java program (which is standard-compliant and follows certain rules) can run unmodified on all supported platforms Ex:Windows or Linux.
2. Object-orientated programming language: Except the primitive data types, all elements in Java are objects.

IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 12, December 2017

3. Strongly-typed programming language: Java is strongly-typed, e.g., the types of the used variables must be pre-defined and conversion to other objects is relatively strict, e.g., must be done in most cases by the programmer.
4. Interpreted and compiled language: Java source code is transferred into the byte code format which does not depend on the target platform. These bytecode instructions will be interpreted by the Java Virtual machine (JVM). The JVM contains a so called Hotspot-Compiler which translates performance critical byte code instructions into native code instructions.
5. Automatic memory management: Java manages the memory allocation and de-allocation for creating new objects. The program does not have direct access to the memory. The so-called garbage collector automatically deletes objects to which no active pointer exists.Structured Query language (SQL) is standard language for dealing with Relational Databases. SQL programming can be effectively used to insert, search, update, delete database records. SQL databases support stored procedure sql which allow database developers to implement part of the business logic into the database.

II. JAVA APIS FOR BIOINFORMATICS

Java provides a rich source of APIs for bioinformatics study of DNA and protein sequences. However, design and maintenance of these APIs present their own challenges .The development of APIs that couple biological and computational knowledge to formally describe complex biological data types significantly reduces the number of conflicting formats and the time required to access and meaningfully analyzes biological data.

Some of the Java-based bioinformatics APIs that are being developed to reduce the programming task are listed below.

1. **BioJava** - An open source project dedicated to provide Java tools for processing biological data. BioJava's goal is to create an API that automates common bioinformatics tasks while providing a foundation for bioinformatics-based software projects.

2. **caBio (Cancer Bioinformatics Infrastructure Objects)** -is one component of the National Cancer Institute's Centre for Bioinformatics (NCICB), caCORE research management system. The caBio API contains the implementations of various biomedical objects to facilitate consistent data representation and data integration projects.

3. **ENSJ**- is the Java implementation of the EnsEMBL driver and data adaptors. ENSJ allows a developer to access sequence or annotation information stored in the EnsEMBL database. Recently, a new prototype API, called MartJ, has been developed and allows the developers to access EnsEMBL's Mart database that focuses on the fast and flexible multi-organism data-mining.

4. **Phylogenetic Analysis Library (PAL)** -is a Java API dedicated to the subset of bioinformatics analysis that pertains to the evolutionary development of genomes (DNA and protein sequence).

5. **KDOM** - The Knowledge Discovery Object Model (KDOM) is a bioinformatics-based API designed to represent and manage biological knowledge during application development.

6. **MAGE**-stk is an example of a bioinformatics-based API that provides a Java representation of the information required to describe a particular experiment such as a microarray experiment.

Advances in biology and medicine have reached the stage where it is now possible to acquire a thorough and very detailed understanding of human biology and inheritance at the molecular level. This understanding will require mapping and sequencing of DNA on a massive scale, a task which cannot be accomplished efficiently with current technologies.

Two major tools are needed:
1. The sequence of a reference human genome
2. Efficient methods for obtaining and interpreting the large amount of additional sequence data needed for a wide variety of biological and medical studies.

Consider the following Escherichia coli str.k12 substr MG 1655 chromosome, This reference sequence can be stored using database.Similiarly we can even store millions of individual varieties of genome sequences.The reference sequence can be collected using BLAST or FASTA reference sequences.

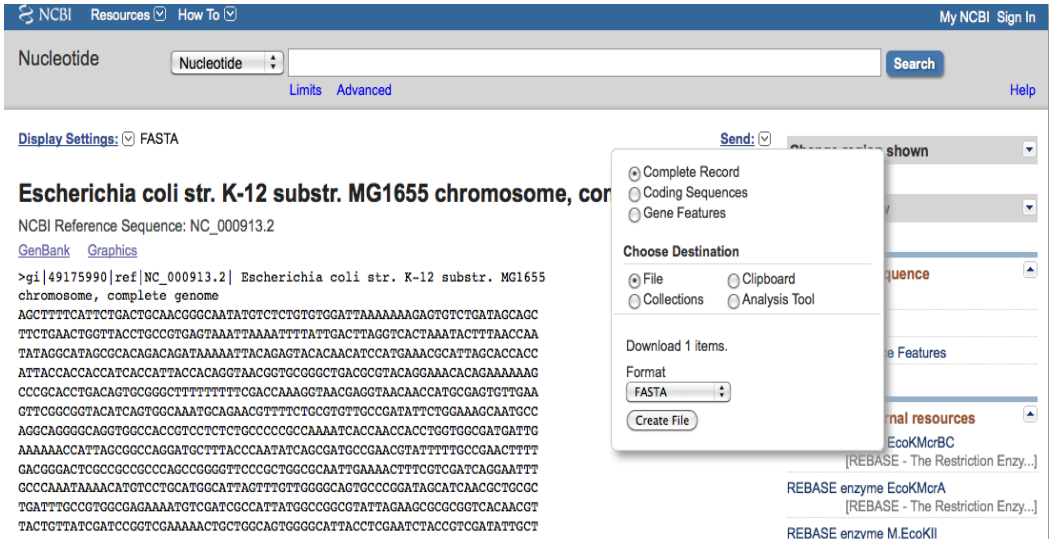
IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

ISO 3297:2007 Certified

Vol. 5, Issue 12, December 2017



The above DNA reference sequence can be read and processed using the java program. Java provides various features to read and process the data .Here we can create our own package called symbol to read and process the nucleotide sequences. The symbol can be stored using the variable s as and when the individual symbols are read from the given input sequences. Thus we can retrieve the input genome sequences easily and effectively.

```
package symbols;
import java.util.Iterator;
import org.biojava.bio.seq.DNATools;
import org.biojava.bio.symbol.FiniteAlphabet;
import org.biojava.bio.symbol.Symbol;
public class TrialSymbol {
  public static void main(String[] args) {
      FiniteAlphabet dna = DNATools.getDNA();
      Iterator dnaSymbols = dna.iterator();
      while (dnaSymbols.hasNext()) {
         Symbol s = (Symbol) dnaSymbols.next();
         System.out.println(s.getName());
      }
    }
  }
```

The RNA sequences can be read as shown below:

```
package symbols;
import java.util.Iterator;
import org.biojava.bio.seq.RNATools;
import org.biojava.bio.symbol.FiniteAlphabet;
import org.biojava.bio.symbol.Symbol;

public class TrialSymbolModified {

   public static void main(String[] args) {
      FiniteAlphabet rna = RNATools.getRNA();
      Iterator rnaSymbols = rna.iterator();
      while (rnaSymbols.hasNext()) {
         Symbol s = (Symbol) rnaSymbols.next();
         System.out.println(s.getName());
      }
   }
}
```


DOI 10.17148/IJIREEICE.2017.51204

IJIREEICE
ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526
International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 12, December 2017

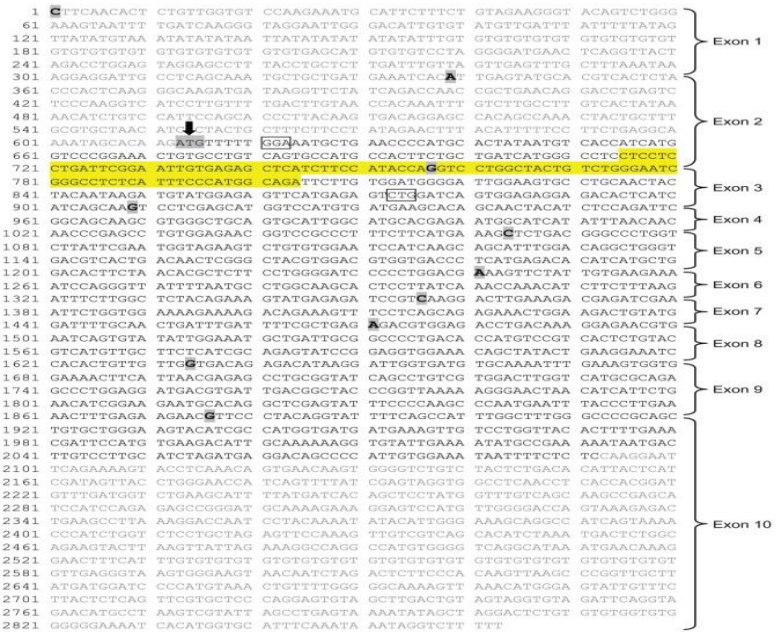
The sequence thus read can also be grouped as meaningful groups called tokens.Using getName() we can read the individual symbols.Built in function length() may be used to find out the length of given sequence.Using SymbolAt() function it is also possible to retrieve the first symbol of the sequence.

```
package symbols;

import org.biojava.bio.BioException;
import org.biojava.bio.seq.DNATools;
import org.biojava.bio.seq.io.SymbolTokenization;
import org.biojava.bio.symbol.Alphabet;
import org.biojava.bio.symbol.SimpleSymbolList;
import org.biojava.bio.symbol.SymbolList;
public class TrialSymbolList {

   public static void main(String[] args) throws BioException {
      String seqString = "GATTACA";
      Alphabet dna = DNATools.getDNA();
      SymbolTokenization dnaToke = dna.getTokenization("token");
      SymbolList seq = new SimpleSymbolList( dnaToke, seqString );
      System.out.println("Alphabet = " + seq.getAlphabet().getName());
      System.out.println("Length = " + seq.length());
      System.out.println("First symbol = " + seq.symbolAt(1).getName() );
} }
```

Thus we can effectively read and store the genome sequences using java program. Our next step is to store the data in database for future reference. This can be done using structural query languages. Consider a sequence from Exon2 which is to be examined. We can read and store the sequence in database. If we refer the same sequence we need not perform the lengthy operations of reading and storing the data. Rather we would get a quick reference to the stored sequence in less time. Thus we can store million of database sequences.



III. CONCLUSION

The advantages of java can be applied to the field of genomics effectively.Genome prediction technique using java is a way to retrieve ,process the data in an efficient manner. Millions of genome sequences can be stored for future reference.If any user comes across same sequence,the sequence can be retrieved and processed with few seconds because of stored genome sequences.Thus the performace of the operation will be high.Thus using the java programming concepts we can add more feautures to theHuman Genome Project.

DOI 10.17148/IJIREEICE.2017.51204

IJIREEICE

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

ISO 3297:2007 Certified

Vol. 5, Issue 12, December 2017

REFERENCES

[1].Altschul SF, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. , 1997, vol. 25 (pg. 3389-3402)
[2].Ewing B, et al. Base-calling of automated sequencer traces using phred, Genome Res. , 1998, vol. 8 (pg. 175-185)
[3].Mangalam H. The Bio* toolkits - a brief overview, Brief. Bioinform. , 2002, vol.3(pg. 396-302)
[4].Stajich JE, et al. The Bioperl toolkit: Perl modules for the life sciences, Genome Res. , 2002, vol. 12 (pg. 1611-1618)
[5].Bliven S., Prlić A. Circular permutation in proteins. PLoS Comput. Biol. 2012;8:e1002445.
[6].Finn R.D., et al. HMMER web server: interactive sequence similarity searching. Nucleic Acids Res. 2011;39(Web Server issue):W29–W37.
[7]. Garavelli J.S. The RESID Database of Protein Modifications as a resource and annotation tool.Proteomics. 2004;4:1527–1533.
[8].Montecchi-PalazziL.,etal.The PSI-MOD community standard for representation of protein modification data. http://www.ncbi.nlm.nih.gov/pubmed/18688235. 2008
[9]. Stein L.D., et al. The Generic Genome Browser: a building block for a model organism system database. Genome Res. 2002;12:1599–1610
[10].Velankar S., et al. E-MSD: an integrated data resource for bioinformatics. Nucleic Acids Res. 2005;33(Database issue):D262–D265.
[11].National Center for Biotechnology Information (NCBI), http://www.ncbi.nlm.nih.gov
[12].Kalpana Raja.,et al. Bio programming prospects of java:a computational move towards the understanding of the biological aspects of genes and proteins
[13].Durbin, R.; Eddy, S.; Krogh, A.; Mitchison, G. Biological sequence analysis: Probabilistic models of proteins and nucleic acids.
[14].Rouchka, E. C. (1999): Pattern Matching Techniques and Their Applications to Computational Molecular
[15].Biology - A Review, bioinformatics.louisville.edu/localresources/papers/WUCS-99-09.pdf (13 Nov 2011)
[16].Regular Expressions in Java, http://www.tutorialspoint.com/java/java_regular_expressions.htm
[17].H. M. Deitel - Deitel & Associates, Inc., P. J. Deitel - Deitel & Associates, Inc. JavaTM How to program. Sixth Edition
[18].Kalpana Raja / Indian Journal of Computer Science and Engineering (IJCSE) ISSN : 0976-5166 Vol. 2 No. 6 Dec 2011-Jan 2012 955
[19].Stajich JE, Block D, Boulez K, et al. (October 2002). "The Bioperl toolkit: Perl modules for the life sciences"
[20].Rice P, Longden I, Bleasby A (June 2000). "EMBOSS: the European Molecular Biology Open Software Suite"
[21].Chen K, Jung YS, Bonagura CA, et al. (February 2002). "Azotobacter vinelandii ferredoxin I: a sequence and structure comparison approach to alteration of [4Fe-4S]2+/+ reduction potential"