International Journal of Applied Science and Engineering Review

ISSN: 2582-6271

Vol.1 No.6; Dec. 2020

PROCESSING GENETIC DATA USING BIOJAVA

Eziechina Malachy A.[1], Esiagu Ugochukwu E.[2] & Ojinnaka, Ebuka R.[3] and Okechukwu Oliver[4]
[1,2] Department of Computer Science, Akanu Ibiam Federal Polytechnic Unwana
[3]Department Of Science Education, Michael Okpara University of Agriculture, Umudike
[4]Department Of Mathematics Education, Enugu State University of Science and Technology, Enugu

ABSTRACT

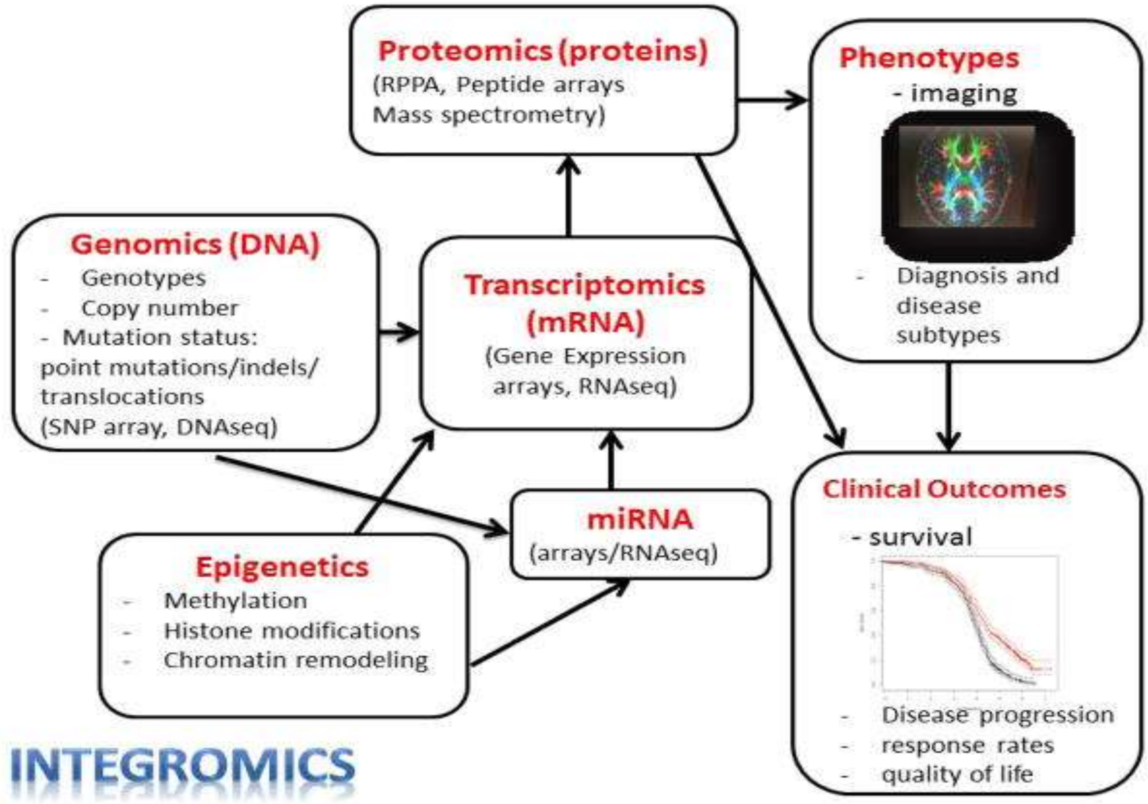
The study, analysis and processing of biological data using computer is known as bioinformatics. Bioinformatics is an interdisciplinary science that develops and applies techniques to generate, organize, analyze, store, and retrieve biological data. Bioinformatics make use of expertise from the fields of computer science, statistics, and biology. The fundamental element of bioinformatics is the genetic data and the associated gene expression. Genetic data is the entire Deoxyribonucleic Acid (DNA) properties of an organism, both heritable and inheritable. BioJava is a computer solution system that is equipped with powerful functionalities to handle genetic data and its complexities. This paper therefore dwells on the application of computer tools in the study and computation of genetic data. Statistical and computer models are used to illustrate some basic genetic and computer concepts. Samples of data generated from genomic studies were manipulated using Biojava. From the result it was revealed that BioJava proved very effective in analysis and computation of genetic data. Biojava supports standards-based middleware technologies to provide seamless access to remote data, annotation and computational servers, thereby, enabling researchers with limited local resources to benefit from available public infrastructure.

KEYWORDS: Genetics, Database, DNA, Biojava, Bioinformatics, geWorkbench.

INTRODUCTION

The advent of high-throughput multi-platform genomics technologies providing whole-genome molecular summaries of biological samples has revolutionalized biomedical research. These technologies yield highly structured big data, whose analysis poses significant quantitative challenges. The field of Bioinformatics has emerged to deal with these challenges, and is comprised of many quantitative and biological scientists working together to effectively process these data and extract the treasure trove of information they contain. The name “bioinformatics” was used for the first time by Paulien Hogeweg in 1970, implying the study of information processes in biotic systems (Hogeweg, 2011). Starting from 2001, the field accelerated with the announcement of Human Genome Project. Bioinformatics was hardly distinguishable from molecular biology in 1970’s; it was established as a separate science with the need of increasing amount of data, and with the opportunities computer science offered.

A large number of bioinformatics techniques have been developed in recent years to serve the needs of biomedical research (see Fig 1).



(Dolinski, 2012)

Fig.1 Types of Multi-platform Genomics Data and Their Interrelationships

The field is moving rapidly, with new and improved approaches appearing frequently. The fast pace of change and the technical sophistication of these approaches creates a barrier of adoption for ordinary biologists. The problem is exacerbated by the integrative nature of biomedical research, which requires combining data from multiple genomic/biomedical databases and using an array of advanced analyses, often available only in the form of command line programs (Kumar and Dudley, 2007; Reich et al., 2006). Additionally, due to their sheer size and dimensionality, analysis of genomic data sets can be computationally very demanding. It is unlikely that every biomedical researcher that would like to utilize such analyses will have access to local/institutional hardware resources capable of supporting their execution. It is then important to facilitate sharing of public infrastructure through the use of technologies such as grid computing (Kurc et al., 2009; Stevens et al., 2003).

Rapid technological advances accompanied by a sharp decline in experimental costs have led to a proliferation of genomic data across many scientific disciplines and virtually all disease areas. These include high-throughput technologies that can profile genomes, transcriptomes, proteomes and metabolomes at a comprehensive and detailed resolution unimaginable only a couple of decades ago. This has led to data generation at an unprecedented scale in various formats, structure and sizes (Bild, 2014). The general term Bioinformatics refers to a multidisciplinary field involving computational biologists, computer scientists, mathematical modellers, systems biologists, and statisticians exploring different facets of the data ranging from storing, retrieving, organizing and subsequent analysis of biological data. Given the myriad challenges posed by this complex field, bioinformatics is necessarily interdisciplinary in nature, as it is not feasible for any single researcher in themselves to possess all clinical, biological, and computational, data management, mathematical modelling, and statistical knowledge and skills necessary to optimally discover and validate the vast scientific knowledge contained in the outputs from these technologies.

The large pool of data generated is manipulated with the aid of software tools such as Structural Query Language (SQL) and JAVA programming language. SQL is responsible for storage, organising and retrieval of the genetic data, while JAVA programming language is the platform upon which the computation and linking would be done.

BioJava is an established open-source project driven by an active developer community (Holland, 2008). It provides a frame-work for processing commonly used biological data. The geworkbench supports data range in scope from DNA and protein sequence information up to the level of 3D protein structures. BioJava5 as a geworkbench platform provides various file parsers, data models and algorithms to facilitate working with the standard data formats and enables rapid application development and analysis. The project is hosted by the Open Bioinformatics Foundation (OBF, http://www.open-bio.org, 2008), which provides the source code repository, bug tracking database and email mailing lists.

Statisticians have a unique perspective and skill set that places them at the canter of this process. One of the key attributes that sets statisticians apart from other quantitative scientists is their understanding of variability and uncertainty quantification. These are essential considerations in building reproducible methods for biological discovery and validation, especially for complex, high-dimensional data as encountered in genomics. Statisticians are “data scientists” who understand the profound effect of sampling design decisions on downstream analysis, potential propagation of errors from multi-step processing algorithms, and the potential loss of information from overly reductionist feature extraction approaches. They are experts in inferential reasoning, which equips them to recognize the importance of multiple testing adjustments to avoid reporting spurious results as discoveries, and to properly design algorithms to search high-dimensional spaces and build predictive models while obtaining accurate measures of their predictive accuracy.

While statisticians have been involved in many aspects of bioinformatics, they have been hesitant to get heavily involved in other aspects. For example, many statisticians are primarily interested in end-stage modelling after all of the data already been collected and pre-processed. Statistical expertise in the experimental design and low-level processing stages are equally if not more important than end-stage modelling, since errors and inefficiencies in these steps propagate into subsequent analyses, and can preclude the possibility of making valid discoveries and scientific conclusions even with the best constructed end-stage modelling strategies. This has resulted in a missed opportunity for the statistical community to play a larger leadership role in bioinformatics that in many cases has been instead assumed by other quantitative scientists and computational biologists, and congruously, a missed opportunity for biologists as well, to more efficiently learn true reproducible biological insights from their data.

Designing a Computer model

Computer simulation is the reproduction of the behaviour of a system using a computer to simulate the outcomes of a mathematical model associated with the said system (see Fig. 2).

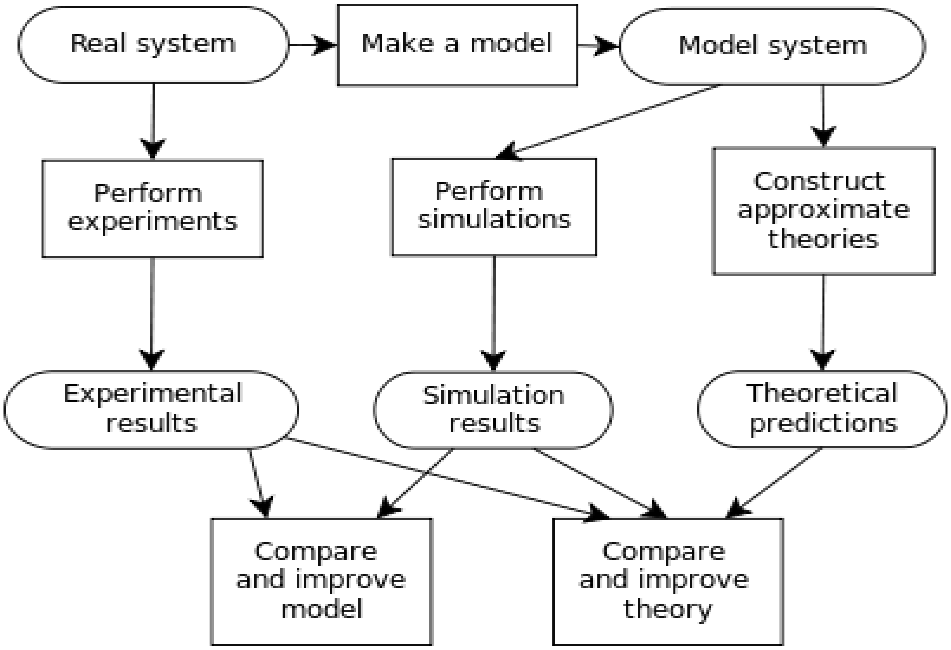


Fig. 2 The Process of building a computer model, and the interplay between experiment, simulation, and theory

Since they allow checking the reliability of chosen mathematical models, computer simulations have become a useful tool for the mathematical modelling of many natural systems in physics (computational physics), astrophysics, climatology, chemistry, biology and manufacturing, human systems in economics, psychology, social science, health care and engineering. Simulation of a system is represented as the running of the system's model. It can be used to explore and gain new insights into new technology and to estimate the performance of systems too complex for analytical solutions.

Computer simulations are realized by running computer programs that can be either small, running almost instantly on small devices, or large-scale programs that run for hours or days on network-based groups of computers. The scale of events being simulated by computer simulations has far exceeded anything possible (or perhaps even imaginable) using traditional paper-and-pencil mathematical modelling. Over 10 years ago, a desert-battle simulation of one force invading another involved the modelling of 66,239 tanks, trucks and other vehicles on simulated terrain around Kuwait, using multiple supercomputers in the DoD High Performance Computer Modernization Program. Other examples include a 1-billion-atom model of material deformation; a 2.64-million-atom model of the complex protein-producing organelle of all living organisms, the ribosome, in 2005.

Because of the computational cost of simulation, computer experiments are used to perform inference such as uncertainty quantification.

The Domain of Genetic Data

One of the world’s most promising fields is biotechnology, making use of many biological sciences. Imagine that you take sample from a hot and acidic crater, which contains many different organisms. Sequencing the total DNA of the sample (metagenomic analysis) will give very important clues about how different species in this niche manage the high temperature/low pH problem, using which genes; you may even discover novel genes responsible for these organisms to survive in extreme conditions. Finding novel genes is extremely important in biotechnology. The governments and biotechnology companies invest a lot for gene discovery and gene classification. Also, the conversion of alcohol to bio fuel is another great area of interest.

Genetic information differs between individuals of specie. Although this variation helps the species adapt and survive, it also causes some individuals to be prone to specific diseases or to have genetic diseases. Advanced genetic studies are carried out to link the diseases to the responsible genes or mutations. Analyzing someone’s genome, you can plot which diseases s/he carries or have risk to develop. Geneticists might use these bioinformatics tools themselves, but most of the time a bioinformatician gets involved.

Sequencing techniques getting widely applied has become a very common approach for genetic diagnosis. The genetic diagnosis field is highly developed in Turkey, and highly demanded from the neighbouring countries. Yet, as we have implied before, holding a data of 3 billion base has no practical use. We need bioinformaticians to select out meaningful knowledge out of this huge data.

Involving Computer Technology and Implementation

geWorkbench integrates many computational resources and makes them available through a unified user interface. For biomedical researchers with little or no computational training, this approach facilitates adoption by eliminating many steps that require programming skills (e.g. transformations from one file format to another; staging of databases and programs; dealing with operating system (OS) shells in order to execute command line programs). For software developers, geWorkbench provides an open source, component-based architecture that enables the addition of new functionality in the form of plug-in modules (which can further leverage existing tooling that streamlines access to server-side components). Extensive documentation is available (manuals, online help and tutorials) to guide users in the proper use of the application. An innovative logging framework collects and mines data about how various modules are being utilized, offering the possibility for novice users to learn from their more advanced peers. geWorkbench comprises at present more than 70 distinct modules, supporting the integrated analysis and visualization of many types of genomic data.

Users of geWorkbench can:

1. Load data from disk or from remote data sources (such as the microarray data repository at the National Cancer Institute, (https://array.nci.nih.gov); the protein structure database; and sequence databases at the University of Santa Cruz and the European Molecular Biology Laboratory).
2. Visualize gene expression, molecular interaction networks, protein sequence and protein structure data in a variety of ways.
3. Access client- and server-side computational analysis tools such as t-test analysis, hierarchical clustering, self-organizing maps, analysis of variance (ANOVA), regulatory and signalling network reconstruction, basic local alignment search tool (BLAST), pattern/motif discovery, etc.
4. Validate computational hypothesis through the integration of gene and pathway annotation information from curetted sources as well as through enrichment analyses.

Several modules have been developed in collaboration with investigators from the Centre for the Multi-scale Analysis of Genomic and Cellular Networks (MAGNet), (http://magnet.c2b2.columbia.edu), one of seven National Centres for Biomedical Computing (NCBCs, http://www.ncbcs.org); The mission of the MAGNet Centre is to provide the research community with novel, structural and systems biology methods and tools for the dissection of molecular interactions in the cell and for the interaction-based elucidation of cellular phenotypes. Other geWorkbench modules are wrapped versions of pre-existing third party software tools such as Cytoscape , and several analysis modules from the Microarray Experiment Viewer and GenePatt (with screenshots) (http://www.geworkbench.org), under the 'Plug-ins' page.

BioJava is an established open-source project driven by an active developer community. It provides a frame- work for processing commonly used biological data and has seen contributions from460 developers in the 12 years since its creation. They supported data range in scope from DNA and protein sequence information up to the level of 3D protein structures. BioJava provides various file parsers, data models and algorithms to facilitate working with the standard data formats and enables rapid application development and analysis. The project is hosted by the Open Bioinformatics Foundation (OBF, http://www.open-bio.org), which provides the source code repository, bug tracking database and email mailing lists. It also supports projects such as BioPerl, BioPython, BioRuby, etc.

Over the last 2 years, large parts of the original code base have been rewritten. BioJava now consists of several independent modules built using Maven (http:// maven.apache.org). The original code has been moved into a separate biojava-legacy project, which is still available for backwards compatibility.

BioJava observes modular programming core module. The core module provides classes to model nucleotide and amino acid sequences and their inherent relationships. Emphasis was placed on using Java classes and method names to describe sequences that would be familiar to the biologist and provide a concrete representation of the steps in going from a gene sequence to a protein sequence to the computer scientist. BioJava leverages recent innovations in Java. A sequence is defined as a generic interface, allowing the framework to build a collection of utilities which can be applied to any sequence such as multiple ways of storing data. In order to improve the framework's usability to biologists, we also define specific classes for common types of sequences, such as DNA and proteins. One area that highlights this work is the translation engine, which allows the inter-conversion of DNA, RNA and amino acid sequences. The storage of sequences is designed to minimize memory usage for large collections using a 'proxy' storage concept. Various proxy implementations are provided which can store sequences in memory, fetch sequences on demand from a web server file as needed. The two approaches save memory by not loading sequence data until it is referenced in the application. This concept can be extended to handle very large genomic datasets, such as NCBI GenBank or a proprietary database.

Protein structure module

The protein structure module provides tools for representing and manipulating 3D bimolecular structures, with particular focus on protein structure comparison. It contains Java ports of the FATCAT algorithm for flexible and rigid body alignment, a version of the standard Combinatorial Extension (CE) algorithm as well as a new version of CE that can detect circular permutations in proteins. These algorithms are used to provide the Protein Data Bank (PDB), Protein Comparison Tool as well as systematic comparisons of all proteins in the PDB on demand.

Genome and sequencing module

The genome module is focused on the creation of gene sequence objects from the core module by supporting the parsing of GTF files generated by GeneMark, GFF2 files generated by GeneID and GFF3 files generated by Glimmer. The gene sequences can then be written out as a GFF3 format for importing into GMOD.

Alignment module

The alignment module supplies standard algorithms for sequence alignment and establish a foundation to perform progressive multiple sequence alignments. For pairwise alignments, an implementation of the Needleman–Wunsch algorithm computes the optimal global alignment, and the Smith–Waterman algorithm calculates local alignments. This routine also allows predefined anchors to be manually specified that will be included in the alignment produced. Any of the pairwise routines can also be used to perform progressive multiple sequence alignment. Both pairwise and multiple sequence alignments output to standard alignment formats for further processing or visualization.

ModFinder module

The ModFinder module provides new methods to identify and classify protein modifications in protein 3D structures. The module provides an API for detecting protein modifications within protein structures.

Amino acid properties module

The goal of the amino acid properties module is to provide a range of accurate physicochemical properties for proteins. The following peptide properties can currently be calculated: molecular weight, extinction coefficient, instability index, aliphatic index, grand average of hydropath, isoelectric point and amino acid composition. To aid proteomic studies, the module includes precise molecular weights for common isotopically labelled or post-translational modified amino acids. Additional types of PTMs can be defined using simple XML configuration files. This flexibility is especially valuable in situations where the exact mass of the peptide is important, such as mass spectrometry experiments.

Protein disorder module

BioJava now includes a port of the Regional Order Neural Network (RONN) predictor, for predicting disordered regions of proteins. BioJava's implementation supports multiple threads, making it 3-times faster than the original C implementation on a modern quad-core machine. The protein disorder module is distributed both as part of the BioJava library and as a standalone command line executable. The executable is optimized for use in automated analysis pipelines to predict disorder in multiple proteins. It can produce output optimized for either human readers or machine parsing.

Web service access module

More and more bioinformatics tools are becoming accessible through the web. As such, BioJava now contains a web services module that allows bioinformatics services to be accessed using REST protocols. Currently, two services are implemented: NCBI Blast through the Blast URLAPI (previously known as QBlast) and the HMMER web service.

CONCLUSIONS

The BioJava library as an implementation platform for gworkbench provides a powerful API for analyzing DNA, RNA and proteins. It contains state-of-the-art algorithms to perform various calculations and provides a flexible framework for Rapid Application Development in bioinformatics. The library also provides lightweight interfaces to other projects that specialize in visualization tools. The BioJava project site provides an online cookbook which demonstrates the use of all modules through short recipes of common tasks. We are looking forward to extending the BioJava library with more functionality over the coming years.

RECOMMENDATIONS

Based on the conclusions above, the following recommendations are made;

1. The big data that results from genomic studies requires a suitable and dynamic programming platform for effective processing.
2. Gworkbench with its multiplatform and interoperability is the right way to go for the bioinformaticians.
3. Any society that wishes to breakthrough in quality healthcare delivery, should embrace bioinformatics

REFERENCES

Berman, H.M. (2012) The Protein Data Bank & Nucleic Acids, Journal of Biological science 28,335-342.

Bild, A.H. (2014) A field guide to genomics research. PMC free article, Google Scholar.

Dolinski, K. (2010) Gene ontology: tool for the unification of biology. Nature. PMC free article, Google Scholar.

Hamid, J.S.(2009) Data integration in genetics and genomics: methods and challenges.

Prolic,A.(2013)Bioinformatics .http://bioinformatics.oxfordjournals.org Retrieved September 14, 2019.

https://array.nci.nih.gov (2013) Remote data sources. Retrieved July 5. 2019.

http://www.geworkbench.org (2010) Genomic Workbench. Retrieved August National Centres for Biomedical Computing (NCBCs, http://www.ncbcs.org (2014) modules in Biojava. Retrieved September

4, 2019.

OBF, http://www.open-bio.org (2018) Open Biojava. Retrieved July 6. 2019.