UNIT-1- Biology and Computer: Scope and Challenges

1.1. Basic of computer, Internet: Biological Data

1.1.1. Internet

Internet is a system that interconnects the different computer systems across the world. It uses the Internet protocol suite to link devices located in different corners of the world. The Internet system carries an extensive range of information resources and services including World Wide Web (WWW), telephony, electronic mail, etc. It uses standard internet protocols, such as TCP/IP and HTTP, etc.

1.1.2. History of Internet:

-ARPANET is the first Internet network. ARPANET stands for Advanced Research Projects Agency Networks.

- **ARPANET** was introduced by the United States. ARPANET has made the TCP/IP correspondences standard, which characterizes information exchange on the web. In 1972, E-mail was adapted by Ray Tomlinson of BBN to ARPANET. In this, Ray has included @ symbol as address. TCP/IP was introduced in 1982.

In 1978, the British post office telenet, **DATAPAC**, and **TRANSPAC** teamed up to make the main worldwide packet-switched system service, and this was referred to as the IPSS.

- The abbreviation of IPSS is International Packet Switched Service. Network control program was developed by a group called “the network working group”.
- **NSF**, abbreviated as the National Science Foundation, is mainly used to create the similar and parallel network called NSFnet.

--World Wide Web: A web server is a computer that provides web services to the client. A page hosted on the internet is known as web page. It can be viewed by a browser. A browser can help locate a website on the internet. The World Wide Web (WWW) permits user to view multi-media based documents like graphics, animations, audios and/or videos and any subject. In 1990, the World Wide Web was introduced by Tim Berners-Lee of CERN.

--E-Mail: Email is an electronic mail. It is used to send and receive the messages. It consists of two components like message header and message body. The message header contains added addresses and the body contains any information and sends any attached contents. The Internet makes your work easy by communication technologies.

1.1.3. Features of Internet

Let us now discuss the features of Internet. The features are described below −

a.Accessibility

An Internet is a global service and accessible to all. Today, people located in a remote part of an island or interior of Africa can also use Internet.

b.Easy to Use

The software, which is used to access the Internet (web browser), is designed very simple; therefore, it can be easily learned and used. It is easy to develop.

c. Interaction with Other Media

Internet service has a high degree of interaction with other media. For example, News and other magazine, publishing houses have extended their business with the help of Internet services.

d.Low Cost

The development and maintenance cost of Internet service are comparatively low.

Extension of Existing IT Technology

This facilitates the sharing of IT technology by multiple users in organizations and even facilitates other trading partners to use.

e.Flexibility of Communication

Communication through Internet is flexible enough. It facilitates communication through text, voice, and video too. These services can be availed at both organizational and individual levels.

f.Security

Last but not the least, Internet facility has to a certain extent helped the security system both at the individual and national level with components such as CCTV camera, etc.

1.1.4. Biological Data

Twenty-first century biology will be a data-intensive enterprise. Laboratory data will continue to underpin biology's tradition of being empirical and descriptive. In addition, they will provide confirming or disconfirming evidence for the various theories and models of biological phenomena that researchers build. Also, because 21st century biology will be a collective effort, it is critical that data be widely shareable and interoperable among diverse laboratories and computer systems.

1.2.Data Integration, and Challenges Faced in the Integration of Biological data

1.2. 1. Data integration

Data driven biological research has made data integration strategies crucial for the advancements and discovery in a plethora of fields (e.g. genomics, proteomics, metabolomics, environmental sciences, clinical research to name a few) Technically, solutions for data integration have been developed and applied in both corporate and academic sectors. When it comes to biological research, there are different interpretations and levels of data integration people seem to consider ranging from genomic data to protein-protein interactions.

Together with data production, there is no doubt that data management, storage and consequently retrieval, analysis and interpretation are at the core of any biological research project. Moreover, the ability to have access to the actual data sets used in a particular study is often crucial for reproducibility and expansion of such study, hence the emphasis in recent years on Open Science and the various initiatives associated Noticeably, in biological research, the difficulties associated with data integration have only expanded with the advent of high throughput technologies Anyone working with Next Generation Sequencing (NGS) faces challenges associated with a variety of aspects this type of data brings, one of the major being: the volume of the data

Here, we refer to data integration as the computational solution allowing users, from end user (GUI) to power users (API), to fetch data from different sources, combine, manipulate and re-analyse them as well as being able to create new datasets and share these again with the scientific community.

With this definition in mind, it is clear that data integration solutions are imperative for the advancement of research in biological sciences as well as the mechanisms to make such processes traceable, shareable hence “integrable” Here, we provide an overview of the strategies most commonly adopted by the biological research community, current challenges and future directions.

Key concepts and terminology

Data integration should not just rely on software engineers and computational scientists, but needs to be driven by the actual users whose communities need to define, adopt and use standards, ontologies and annotation best practice. Therefore, it is particularly important for the biological research community to get acquainted with the conceptual basis of data integration, its limitations, challenges and actual terminology.

In order to familiarise the experimental biology community of readers, in Table Table11 we present key concepts, definitions and terms used by bioinformaticians and computer scientists

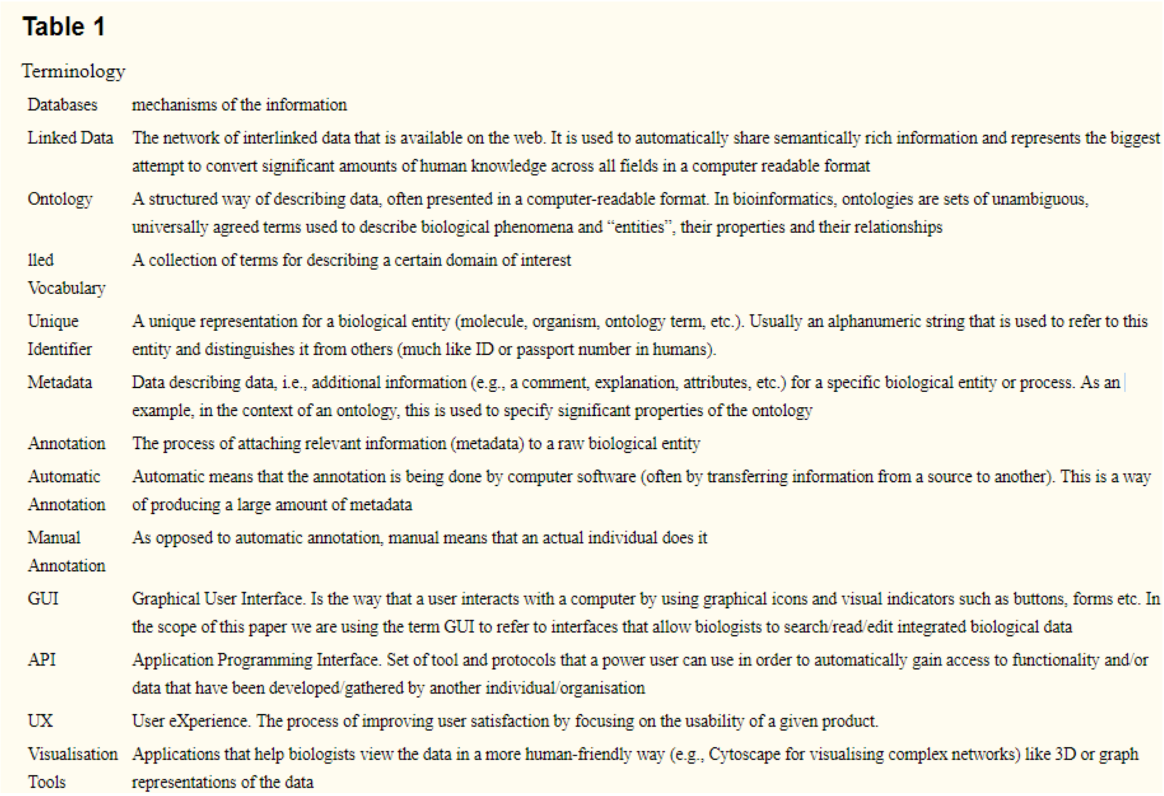

Table 1

Terminology

Databases	mechanisms of the information
Linked Data	The network of interlinked data that is available on the web. It is used to automatically share semantically rich information and represents the biggest attempt to convert significant amounts of human knowledge across all fields in a computer readable format
Ontology	A structured way of describing data, often presented in a computer-readable format. In bioinformatics, ontologies are sets of unambiguous, universally agreed terms used to describe biological phenomena and "entities", their properties and their relationships
lled Vocabulary	A collection of terms for describing a certain domain of interest
Unique Identifier	A unique representation for a biological entity (molecule, organism, ontology term, etc.). Usually an alphanumeric string that is used to refer to this entity and distinguishes it from others (much like ID or passport number in humans).
Metadata	Data describing data, i.e., additional information (e.g., a comment, explanation, attributes, etc.) for a specific biological entity or process. As an example, in the context of an ontology, this is used to specify significant properties of the ontology
Annotation	The process of attaching relevant information (metadata) to a raw biological entity
Automatic Annotation	Automatic means that the annotation is being done by computer software (often by transferring information from a source to another). This is a way of producing a large amount of metadata
Manual Annotation	As opposed to automatic annotation, manual means that an actual individual does it
GUI	Graphical User Interface. Is the way that a user interacts with a computer by using graphical icons and visual indicators such as buttons, forms etc. In the scope of this paper we are using the term GUI to refer to interfaces that allow biologists to search/read/edit integrated biological data
API	Application Programming Interface. Set of tool and protocols that a power user can use in order to automatically gain access to functionality and/or data that have been developed/gathered by another individual/organisation
UX	User eXperience. The process of improving user satisfaction by focusing on the usability of a given product.
Visualisation Tools	Applications that help biologists view the data in a more human-friendly way (e.g., Cytoscape for visualising complex networks) like 3D or graph representations of the data

In computational sciences the theoretical frameworks for data integration have been classified into two major categories namely "eager" and "lazy". The difference between the two approaches is the way the data get integrated. In the eager approach (warehousing), the data are being copied over to a global schema and stored in a central data warehouse; whereas in the lazy approach the data reside in distributed sources and are integrated on demand based on a global schema used to map the data between sources.

Each of the two main categories of data integration has to deal with its own challenges in order to provide the user with a unified view of the data. In the eager approach, researchers face challenges to keep data updated and consistent, and protect the global schema from having corrupted data . In the lazy approach, data are queried at sources and the scientific community is trying to find ways of improving the answering query process and source completeness . Which approach should be used and when depends on amount of data, who owns them and the existing infrastructure.

In biology we see a diversity of implementations across these two approaches being used at a variety of levels and forms like data centralisation, federated databases and linked data. Figure shows the most common schemata used to integrate data in biology.

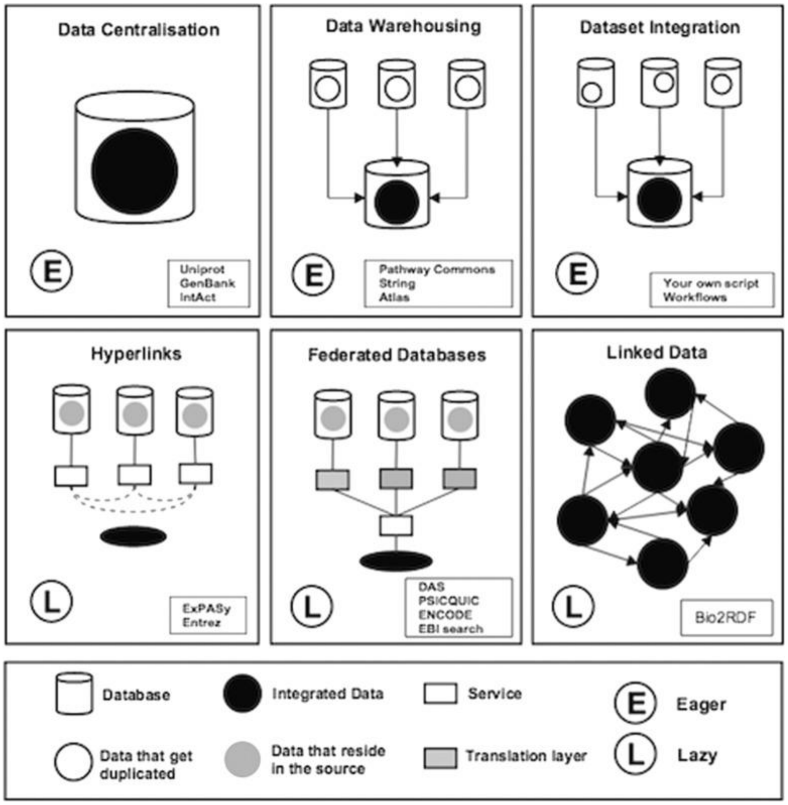

Fig.1. Data integration

Data integration in biological research has its challenges associated to a variety of factors such as standards adoption or easy conversion between data/file formats

Figure illustrates a simplified schematic view of the current state of biological research data integration components. Various attempts to integrate the data rely on translation layers that, by applying agreed standards, transform the data in a unified format in order to integrate them. In other words, different formats for the same type of data (e.g. NGS) need to be “translated” into a unified format by applying shared rules. On top of the integration layer, there are various GUIs that make it possible to utilise (download, analyse, represent, etc) the integrated data. Furthermore, there is a myriad of resources and visualisation tools generated that fail to comply with standards and/or are not compatible with each other On the other hand, controlled vocabularies and ontologies to ease data integration are available for an increasing number of biological domain areas. Some of them can be found at the websites of the OBO (Open Biological and Biomedical Ontologies) foundry], the NCBO (National Center for Biomedical Ontology) BioPortal and the OLS (Ontology Lookup Service). One successful example is the XML-based proteomic standards defined by the HUPO-PSI (Human Proteome Organisation-Proteomics Standards Initiative) consortium (see Table Table2).2. The rest of the paper will discuss key aspects of standards: ontologies, data formats, identifiers, reporting guidelines, consortiums and standard initiatives which will be followed by a section on visualisation.

1.2.2. Challenges Faced in the Integration of Biological data

Computational challenges of data integration

The main goal of any data integration methodology is to extract additional biological knowledge from multiple datasets that cannot be gained from any single dataset alone. To reach this goal, data integration methodologies have to meet many computational challenges. These challenges arise owing to different sizes, formats and dimensionalities of the data being integrated, as well as owing to their complexity, noisiness, information content and mutual concordance (i.e. the level of agreement between datasets).

A number of current data integration methods meet some of these challenges to some extent, whereas the majority of them hardly meet any of them. A reason is that many data integration approaches are based on the methods designed for analysing one data type, and they are further adopted to deal with multiple data types. Thus, these methods often suffer from various limitations when applied to multiple data types. For example, in terms of network integration, standard methods for network analysis fail to simultaneously take into account connectivity structure (topology) of multiple different networks along with capturing the biological knowledge contained in them. They are based on different types of *transformation methods* to project, or merge multiple networks into a single, integrated network on which further analysis is performed Their limitations will be explained later in this article. However, more sophisticated network-based (NB) methods use either *random walk* or *diffusion* processes to simultaneously explore connectivity structures (topologies) of multiple different networks and to infer integrated biological knowledge from all networks concurrently.

However, a majority of data integration studies are based on methods from *machine learning* (*ML*) owing to their ability to integrate diverse biological networks along with other biological data types. Namely, the basic strategy has been to use standard ML methods and extend them to incorporate disparate data types.

Methodologies for biological data integration and highlight their applications in various areas of biology and medicine

different size, format and dimensionality of datasets, —

- presence of noise and data collection biases in datasets, —
- effective selection of informative datasets, —
- effective incorporation of concordant and discordant datasets, and —
- scalability with the number and size of datasets. —

computational challenges to be the most important, as every data integration methodology aims to address them at least to some extent.

1.3.Data sources in Life Sciences, Challenges in information integration.

1.3.1. Data sources in life science

Modern life science research is very data-intensive: to understand the functions of biological systems, scientists rely on a variety of data about the systems and their functions. These data are very heterogeneous in scope, ranging from molecular mechanisms to phenotypes and beyond. The ways it is generated and collected are also varied, as they have been assembled over time by large and small scientific investigations, deposited in large institutional repositories and conveyed in publications.

The current life science literature includes about 27 million papers in PubMed, genomic databases storing 200 million sequences in GenBank (https://www.ncbi.nlm.nih.gov/genbank/statistics/), and pathway data scattered over at least 165 databases (http://www.oxfordjournals.org/nar/database/cap/). This is a very large and complex information landscape whose exploitation is one of the keys to a data-driven approach to science. Exploiting such scattered information requires taking an integrated view of the data, which in turn requires locating relevant data and making it accessible in a way that facilitates integration. This is a complex task, not only because of the intrinsic nature of the information itself, but also because the same information can be delivered by a variety of providers, with different data formats, terminologies, and update policies. In addition, many datasets aggregate other data sources, in more or less indirect ways, so the provenance of the dataset itself can be hard to delineate.

In order to cope with such a wide range of representations and formats, approaches based on Linked Data have been proposed (1) and now partially adopted (2). These approaches (presented in more detail later) provide means of using ontologies to describe data, as well as a standard language (the Resource Description Framework, or RDF) and access protocol (SPARQL). In addition to Linked Data, the Findable, Accessible, Interoperable and Reusable (FAIR) initiative (3) has proposed a broader set of requirements to enable life science data interoperability.

The RDF and Linked Data formats have significant roles to play in allowing heterogeneous databases, scattered around the world, to be used in an integrated manner. In RDF, entities are identified via global identifiers that can be resolved over the Internet using Web technologies. Once resolved, relevant information is provided in standard formats, making use of standard predicates that also provide explicit links to other entities and datasets. This combination enables information to easily be retrieved and merged according to a unified (albeit schema-light) model. An example of this approach that combines information about a gene, the protein it produces, related diseases, and references to relevant papers can be found in (2).

The RDF and Linked Data formats are being adopted by variety of large and small dataset providers. To cite a few relevant examples, the European Bioinformatics Institute (EBI) provides their major databases including UniProt in RDF (4), in addition to legacy data representations such as text files. Similarly, the National Center for Biotechnology Information provides RDF versions of MeSH (5) and PubChem (6). The Database Center for Life Science (DBCLS) has constructed several RDF datasets in cooperation with the National Bioscience Database Center (NBDC) and has set up a portal site called the NBDC RDF Portal (https://integbio.jp/rdf/) to disseminate them.

That said, the impact of such Linked Data approaches is being severely limited by datasets being published and republished without any real quality control. To illustrate this, imagine that we are interested in finding a representation of an apoptosis pathway. This information is available in different representations from different providers, such as GO (7), Reactome (8) and BioCyc (9), which in turn often derive their datasets from the literature, through more or less curated processes.

The datasets delivered by these providers are then further integrated and republished by systems such as UniProt (10) and PathwayCommons (11), which may introduce additional data or normalization steps. A single dataset can be published by multiple providers, in various (not always explicitly distinguished) versions. For instance, PathwayCommons, the EBI RDF platform (4), OpenPHACTS (12) and Linked Life Data (http://linkedlifedata.com/) provide different versions of data from the Reactome database. Most of these providers deliver datasets in RDF and even release SPARQL access points, which would be easy to process if we were interested in retrieving all genes associated with apoptosis. However, if the sites provide different data, which one should we use? Short of just merging all the data, there is no easy way to determine which dataset to use.

1.3.2. Computational Biology: Challenges and Opportunities

The current issue of the quarterly publication, CTWatch, focuses on the issues and challenges facing the field of computational biology today and in the future. A recurring theme throughout all of the articles is that the field of biology is becoming increasingly data driven and is producing data faster than computers can process it. The authors address the limitations of our current cyberinfrastructure and suggest strategies to overcome these challenges.

In his introduction, “Trends in Cyberinfrastructure for Bioinformatics and Computational Biology,” Rick Stevens, Associate Laboratory Director, Computing and Life Sciences of Argonne National Laboratory and Professor, Computer Science Department of The University of Chicago, outlines three major trends in biology research: the increasing availability of high-throughput data, the acceleration of the pace of questions whose answers rely on increasing computation resources, and simulation and modeling technologies that will eventually lead to predictive biological theory.

Stevens addresses the role of petascale computing with regard to fundamental biological problems, such as the evolutionary history of genes and genomes. This is significant, as the number of completed genome sequences will reach 1,000 in the next few years. He provides a list of multiple “problem areas” and their estimated time to completion at three levels of computing power (360, 1000, and 5000 teraflops). For example, on the IBM Blue Gene/L, screening “all known microbial drug targets against the public and private databases of chemical compounds to identify potential new inhibitors and potential drugs," would take one year for all microbial targets at 360 teraflops, a one month for all microbial targets at 1000 teraflops, and one machine year for all known human drug targets at 5000 teraflops.

Eric Jakobsson of the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign discusses the initiatives that will be required between applications scientists and system architects in order to come up with a suitable cyberinfrastructure for biology in "Specifications for the Next-Generation Computational Biology Infrastructure." One of the five integration models Jakobsson outlines is "Integration of algorithmic development with computing architecture design." He says:

"The different types of biological computing have vastly different patterns of computer utilization. Some applications are very CPU-intensive, some require large amounts of memory, some must access enormous data stores, some are much more readily parallelizable than others, and there are highly varied requirements for bandwidth between hard drive, memory, and processor."

Jakobsson suggests that more extensive mutual tuning of computer architecture to applications software would make existing and projected computational resources more productive. One case of such tuning is the molecular simulation code Blue Matter, designed to leverage the architecture of the IBM Blue Gene supercomputer. Jakobsson praises the Blue Matter-Blue Gene combination, declaring that it has enabled important new discoveries.

Jakobsson also calls for better training in the area of computational biology at the undergraduate and graduate levels. He points to the University of California at Merced as one institution that has fully integrated computing into all levels of its biology curriculum as called for in the National Academy of Sciences BIO 2010 report.

1.4.Aims and tasks of computational biology, challenges and opportunities

1.4.1. Genome sequencing

In "Genome Sequencing vs. Moore's Law: Cyber Challenges for the Next Decade" Folker Meyer of the Argonne National Laboratory addresses the challenge of the number of sequenced genomes growing faster than Moore's Law. He states that the number of available complete genomic sequences is doubling every 12 months, faster than Moore's 18 months. "The analysis of genomic sequences requires serious computational effort: most analysis techniques require binary comparison of genomes or the genes within genomes. Since the number of binary comparisons grows as the square of the number of sequences involved, the computational overhead of the sequence comparisons alone will become staggering."

As the number of sequences grows so do the number of algorithms to study them, requiring additional computer power. For example, using Hidden Markov Models to search for sequence similarities not visible with the traditionally used BLAST algorithm requires greater computing resources. The author states that the TeraGrid is one of the few resources that can handle the computational requirement. We need to overcome these limitations in order to study and better understand "crop plants, pathogens and ultimately human beings." To resolve the gap between data and resource, the author calls for new bioinformatics techniques as well as high-throughput computing, concluding that "biology is in the middle of a paradigm shift towards becoming a fully data driven science."

In "Computing and the "Age of Biology,' " Natalia Maltsev of the Argonne National Laboratory calls for the "development of high-throughput computational environments that integrate (i) large amounts of genomic and experimental data, (ii) comprehensive tools and algorithms for knowledge discovery and data mining, and (iii) comprehensive user interfaces that provide tools for easy access, navigation, visualization, and annotation of biological information." For achieving this integrated environment, Maltev makes four recommendations.

First, she calls for large, public, scalable computational resources to handle the exponential growth of biological data. For example, the largest genomic database, GenBank, contains 56 billion bases, from 52 million sequences; and as the cost of sequencing new genomes drops, the rate of growth of GenBank is expected to increase dramatically.

Second, Maltev proposes a new model to handle the increasing complexity of biological data. She states that biology is becoming increasingly multi-disciplinary, "using information from different branches of life sciences; genomics, physiology, biochemistry, biophysics, proteomics, and many more." The model needs to incorporate various classes of biological information as well as similar classes of data from different resources. According to Maltev, the difficulty with an integrated model is due to "the large volume and complexity of data, the distributed character of this information residing in different databases, shortfalls of current biological ontologies, and generally poor naming conventions for biological objects."

Maltsev's third recommendation is algorithm development. The current bioinformatic tools (for example, BLAST and FASTA) are not adequate to handle the exponential growth of sequence data. Maltev says "bioinformatics will significantly benefit from the development of a new generation of algorithms that will allow efficient data mining and identification of complex multidimensional patterns involving various classes of data."

Maltev's fourth and final recommendation is the development of collaborative environments that will allow researches in different locations to view and analyze the data. Maltev claims that storing data and its analysis in one location will not meet the needs of biology in the future. She also calls for visualization of information to reduce its complexity.

Maltev's article provides an accessible framework for understanding the challenges of computational biology. In the "age of biology," computing and biology will unite to solve major global problems such as curing deadly diseases and ending world hunger.

The message in all of these articles is that biology has become a data-driven discipline and is becoming increasingly more so. Computational resources cannot keep up with the data, and questions are piling up faster than answers. Remedying this situation is essential for progress.

1.5.Internet basics: HTML; Database management system; Database browsing,

1.5.1.Internet Basics,

The Internet is a worldwide telecommunications system that provides connectivity for millions of other, smaller networks; therefore, the Internet is often referred to as a network of networks. It allows computer users to communicate with each other across distance and computer platforms.

The Internet began in 1969 as the U.S. Department of Defense's Advanced Research Project Agency (ARPA) to provide immediate communication within the Department in case of war. Computers were then installed at U.S. universities with defense related projects. As scholars began to go online, this network changed from military use to scientific use. As ARPAnet grew, administration of the system became distributed to a number of organizations, including the National Science Foundation (NSF). This shift of responsibility began the transformation of the science oriented ARPAnet into the commercially minded and funded Internet used by millions today.

The Internet acts as a pipeline to transport electronic messages from one network to another network. At the heart of most networks is a server, a fast computer with large amounts of memory and storage space. The server controls the communication of information between the devices attached to a network, such as computers, printers, or other servers.

An Internet Service Provider (ISP) allows the user access to the Internet through their server. Many teachers use a connection through a local university as their ISP because it is free. Other ISPs, such as America Online, telephone companies, or cable companies provide Internet access for their members.

You can connect to the Internet through telephone lines, cable modems, cellphones and other mobile devices.

1.5.2.HTML

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

HTML is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning HTML:

Create Web site - You can create a website or customize an existing web template if you know HTML well.

Become a web designer - If you want to start a carrer as a professional web designer, HTML and CSS designing is a must skill.

Understand web - If you want to optimize your website, to boost its speed and performance, it is good to know HTML to yield best results.

Learn other languages - Once you understands the basic of HTML then other related technologies like javascript, php, or angular are become easier to understand.

1.5.3. Database Management System

a..Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information.

Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks.

b.A **database management system** stores data in such a way that it becomes easier to retrieve, manipulate, and produce information.

c. Characteristics

Traditionally, data was organized in file formats. DBMS was a new concept then, and all the research was done to make it overcome the deficiencies in traditional style of data management. A modern DBMS has the following characteristics −

- **Real-world entity** − A modern DBMS is more realistic and uses real-world entities to design its architecture. It uses the behavior and attributes too. For example, a school database may use students as an entity and their age as an attribute.
- **Relation-based tables** − DBMS allows entities and relations among them to form tables. A user can understand the architecture of a database just by looking at the table names.
- **Isolation of data and application** − A database system is entirely different than its data. A database is an active entity, whereas data is said to be passive, on which the database works and organizes. DBMS also stores metadata, which is data about data, to ease its own process.

- **Less redundancy** − DBMS follows the rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Normalization is a mathematically rich and scientific process that reduces data redundancy.
- **Consistency** − Consistency is a state where every relation in a database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state. A DBMS can provide greater consistency as compared to earlier forms of data storing applications like file-processing systems.
- **Query Language** − DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and as different filtering options as required to retrieve a set of data. Traditionally it was not possible where file-processing system was used.
- **ACID Properties** − DBMS follows the concepts of **A**tomicity, **C**onsistency, **I**solation, and **D**urability (normally shortened as ACID). These concepts are applied on transactions, which manipulate data in a database. ACID properties help the database stay healthy in multi-transactional environments and in case of failure.
- **Multiuser and Concurrent Access** − DBMS supports multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when users attempt to handle the same data item, but users are always unaware of them.
- **Multiple views** − DBMS offers multiple views for different users. A user who is in the Sales department will have a different view of database than a person working in the Production department. This feature enables the users to have a concentrate view of the database according to their requirements.
- **Security** − Features like multiple views offer security to some extent where users are unable to access data of other users and departments. DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers many different levels of security features, which enables multiple users to have different views with different features. For example, a user in the Sales department cannot see the data that belongs to the Purchase department. Additionally, it can also be managed how much data of the Sales department should be displayed to the user. Since

a DBMS is not saved on the disk as traditional file systems, it is very hard for miscreants to break the code.

A typical DBMS has users with different rights and permissions who use it for different purposes. Some users retrieve data and some back it up. The users of a DBMS can be broadly categorized as follows −

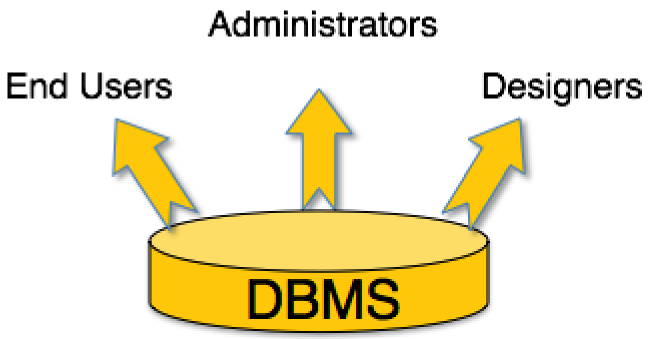

Fig.2. DBMS

- **Administrators** − Administrators maintain the DBMS and are responsible for administrating the database. They are responsible to look after its usage and by whom it should be used. They create access profiles for users and apply limitations to maintain isolation and force security. Administrators also look after DBMS resources like system license, required tools, and other software and hardware related maintenance.
- **Designers** − Designers are the group of people who actually work on the designing part of the database. They keep a close watch on what data should be kept and in what format. They identify and design the whole set of entities, relations, constraints, and views.
- **End Users** − End users are those who actually reap the benefits of having a DBMS. End users can range from simple viewers who pay attention to the logs or market rates to sophisticated users such as business analysts

1.5.4. Advantages of DBMS

- Segregation of applicaion program.
- Minimal data duplicacy or data redundancy.
- Easy retrieval of data using the Query Language.

- Reduced development time and maintainance need.
- With Cloud Datacenters, we now have Database Management Systems capable of storing almost infinite data.
- Seamless integration into the application programming languages which makes it very easier to add a database to almost any application or website.

1.5.5.

1.5.6. Data browsing and data retrieval

In databases, data retrieval is the process of identifying and extracting data from a database, based on a query provided by the user or application.

It enables the fetching of data from a database in order to display it on a monitor and/or use within an application.

Data retrieval means obtaining data from a database management system such as ODBMS. In this case, it is considered that data is represented in a structured way, and there is no ambiguity in data.

In order to retrieve the desired data the user present a set of criteria by a query. Then the Database Management System (DBMS), software for managing databases, selects the demanded data from the database. The retrieved data may be stored in a file, printed, or viewed on the screen.

A *query language*, such as *Structured Query Language* (SQL), is used to prepare the queries. SQL is an American National Standards Institute (ANSI) standardized query language developed specifically to write database queries. Each DBMS may have its own language, but most relational .

How the data is presented

Reports and queries are the two primary forms of the retrieved data from a database. There are some overlaps between them, but queries generally select a relatively small portion of the database, while reports show larger amounts of data. Queries also present the data in a standard format and usually display it on the monitor; whereas reports allow formatting of the output however you like and is normally printed.

Reports are designed using a report generator built into the DBMS.

1.6. Data Retrieval

1.6.1. Basic Example

The following example discusses how a single entity (file/object/attribute/public folder/mailbox) is restored.

The hypothetical entities shown in the figure contains six entities each. Assume that backups are scheduled daily with the first backup occurring on May 10. (The clock times of the backups are unimportant for our purposes.) The figure shows which entities have changed and consequently have been backed up over time.

Assume that on May 16, we request the most recent version (i.e., the default) of entity F. In response, the system retrieves the most recent index file, which was generated by the 5/15 backup. It searches the index for the most recent version of the entity, which is found in the 5/14 backup. The system then retrieves the entity from the backup media and restores it to your client computer.

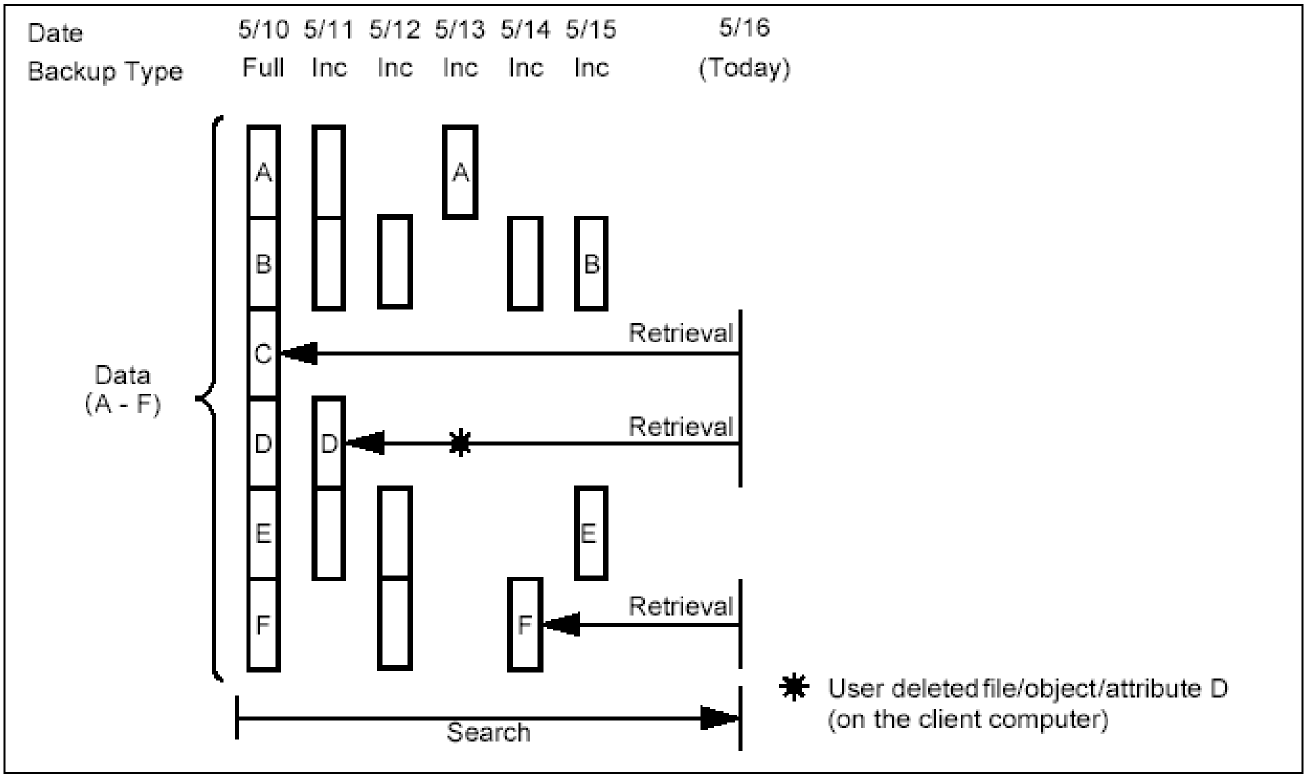

Fig.3.Data retrieval

If we requested entity C instead, the system would search the index and restore the version that was backed up with the full backup that occurred on 5/10.

If you decided to restore the 5/10 backup of Document B rather than the 5/15 backup, during the browse, you would select that version from the browse window. The earliest version appears first in the list and the most recent version appears further down the list.

Finally, if we requested entity D, which was deleted at some time between the 5/12 and 5/13 backups, the search would end with the 5/11 backup and the system would send the entity to the client computer.

1.6.2. Data Retrieval - Example 2

The following example discusses how a folder is restored.

Assume that on May 16 we request the restoration of an entire public folder/mailbox as it existed in its most recent state (i.e., the default). Using the latest index file, which was generated by the 5/15 backup, the browse function retrieves the most recent copy of each folder until all the folders have been restored. Note, in the example, the term Folder refers to a public folder or a mailbox folder.

In this case, the operation would return:

- Folders B and E from 5/15
- Folder F from 5/14
- Folder A from 5/13
- Folder C from 5/10

Folder D is not restored since it did not exist in the backup set on the date that the restore was effective, 5/16.

Message/Item-Level Retrieval

Assume that on May 16, we request the most recent version of message/item B. In response, the browse function retrieves the most recent index file, which was generated by the 5/15 backup. It searches the index for the most recent version of the message/item, which is found in the 5/14 backup. The message/item is retrieved from the backup media and is restored to the client computer.

If we requested message/items C and D instead, the system would search the index and restore those versions from the backups that occurred on 5/10 and 5/11, respectively.

1.6.4. Human genome project

The Human Genome Project was a 13-year-long, publicly funded project initiated in 1990 with the objective of determining the DNA sequence of the entire euchromatic human genome within 15 years. In its early days, the Human Genome Project was met with skepticism by many people, including scientists and nonscientists alike. One prominent question was whether the huge cost of the project would outweigh the potential benefits. Today, however, the overwhelming success of the Human Genome Project is readily apparent. Not only did the completion of this project usher in a new era in medicine, but it also led to significant advances in the types of technology used to sequence DNA.

1.5.6. Initial Principles and Goals of the Human Genome Project

From its inception, the Human Genome Project revolved around two key principles (International Human Genome Sequencing Consortium, 2001). First, it welcomed collaborators from any nation in an effort to move beyond borders, to establish an all-inclusive effort aimed at understanding our shared molecular heritage, and to benefit from diverse approaches. The group of publicly funded researchers that eventually assembled was known as International Human Genome Sequencing Consortium (IHGSC). Second, this project required that all human genome sequence information be freely and publicly available within 24 hours of its assembly. This founding principle ensured unrestricted access for scientists in academia and in industry, and it provided the means for rapid and novel discoveries by researchers of all types. At any given time, approximately 200 labs in the United States were funded by either the National Institutes of Health or the U.S. Department of Energy to support these efforts. In addition, more than 18 different countries from across the globe had contributed to the Human Genome Project by the time of its completion.

Just as the Human Genome Project revolved around two key principles, it also started with two early goals: (1) building genetic and physical maps of the human and mouse genomes, and (2) sequencing the smaller yeast and worm genomes as a test run for sequencing the larger, more complex human genome (IHGSC, 2001). When the yeast and worm efforts proved successful, the sequencing of the human genome proceeded with full force.

1.5.7.Phases of the Human Genome Project

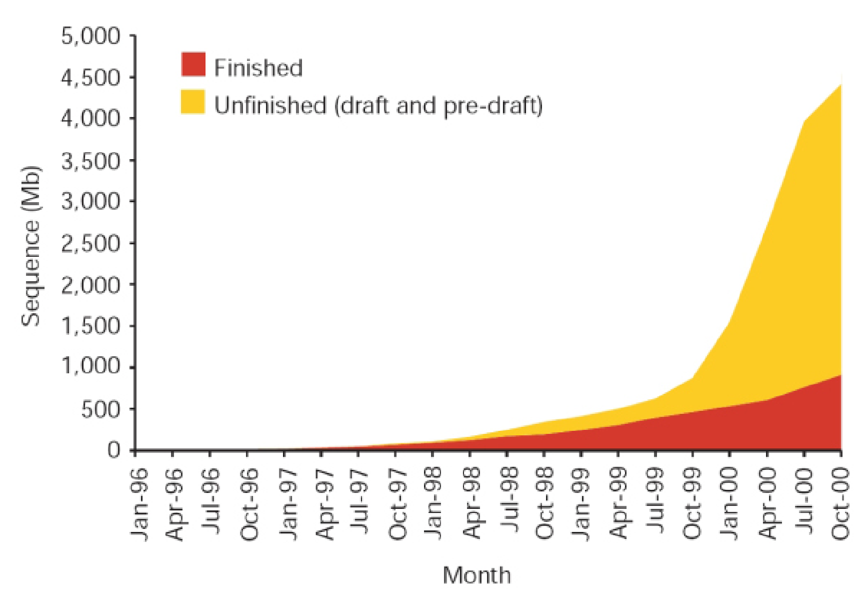

Figure 1 : Total amount of human sequence in the High-Throughput Genomic Sequences (HTGS) division of GenBank.

The total is the sum of finished sequence (red) and unfinished (draft plus predraft) sequence (yellow).

© 2001 Nature Publishing Group International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature* **409,** 867 (2001). All rights reserved.

Based on the insights gained from the yeast and worm studies, the Human Genome Project employed a two-phase approach to tackle the human genome sequence (IHGSC, 2001). The first phase, called the shotgun phase, divided human chromosomes into DNA segments of an appropriate size, which were then further subdivided into smaller, overlapping DNA fragments that were sequenced. The Human Genome Project relied upon the physical map of the human genome established earlier, which served as a platform for generating and analyzing the massive amounts of DNA sequence data that emerged from the shotgun phase. Next, the second phase of the project, called the finishing phase, involved filling in gaps and resolving DNA sequences in ambiguous areas not obtained during the shotgun phase. Figure 1 shows the exponential increase in DNA sequence information deposited in the High-Throughput Genomic Sequences (HTGS) division of GenBank by the end of the shotgun phase. Indeed, the shotgun phase yielded 90% of the human genome sequence in draft form.

The shotgun phase of the Human Genome Project itself consisted of three steps:

1. Obtaining a DNA clone to sequence
2. Sequencing the DNA clone
3. Assembling sequence data from multiple clones to determine overlap and establish a contiguous sequence

The approach used by the members of the IHGSC was called the hierarchical shotgun method, because the team members systematically generated overlapping clones mapped to individual human chromosomes, which were individually sequenced using a shotgun approach (Figure 2). The clones were derived from DNA libraries made by ligating DNA fragments generated by partial restriction enzyme digestion of genomic DNA from anonymous human donors into bacterial artificial chromosome vectors, which could be propagated in bacteria.

When possible, the DNA fragments within the library vectors were mapped to chromosomal regions by screening for sequence-tagged sites (STSs), which are DNA fragments, usually less than 500 base pairs in length, of known sequence and chromosomal location that can be amplified using polymerase chain reaction (PCR). Library clones were also digested with the restriction enzyme HindIII, and the sizes of the resulting DNA fragments were determined using agarose gel electrophoresis. Each library clone exhibited a DNA fragment "fingerprint," which could be compared to that of all other library clones in order to identify overlapping clones. Fluorescence *in situ* hybridization (FISH) was also used to map library clones to specific chromosomal regions. Collectively, the STS, DNA fingerprint, and FISH data allowed the IHGSC to generate contigs, which consisted of multiple overlapping bacterial artificial chromosome (BAC) library clones spanning each of the 24 different human chromosomes (i.e., 22 autosomes and the X and Y chromosomes).

Next, individual BAC clones selected for DNA sequence analysis were further fragmented, and the smaller genomic DNA fragments were subcloned into vectors to generate a BAC-derived shotgun library. The inserts were sequenced using primers matching the vector sequence flanking the genomic DNA insert, and overlapping shotgun clones were used to generate a DNA sequence spanning the entire BAC clone. A summary of this step is shown in Figure 3. The members of the IHGSC agreed that each center would obtain an average of fourfold sequence coverage, with no clone having less than threefold coverage. The term "shotgun" comes from the fact that the original BAC clone was randomly fragmented and sequenced, and the raw DNA sequence data was then subjected to computational analyses to generate an ordered set of DNA sequences that spanned the BAC clone.

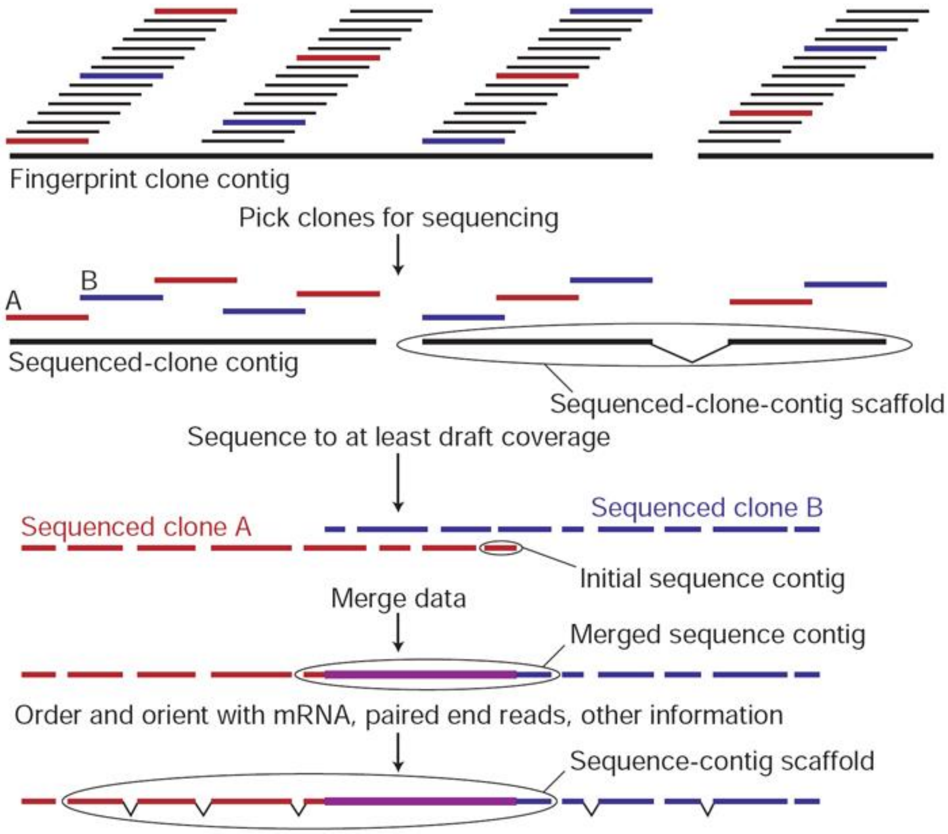

Fig.4. sequencing process

1.7.Various file formats in biological sequence

In the field of bioinformatics there exists many different file formats that store DNA and protein sequence information. There is no one sequence format that is ideal: many are used in different contexts, and can often be converted from one to another for easier access or sharing. Below is a list of file formats and a link to their respective file format specs and descriptions for anyone wishing to get to know the file formats a little better. While there are many different formats out there used by commercial software, this list focuses mainly on open, non-propietary file formats.

 Genbank - quite possibly the standard in sequence file formats, the Genbank format is widely used by public databases such as NCBI. The Genbank file format is quite flexible and allows annotations, comments, and references to be included within the file. The file is plain text and thus can be read with a text editor. Genbank files often have the file extension '.gb' or '.genbank'.

1.7.1 Genbank Sample Record

- EMBL - similar in form to the Genbank file, the EMBL format is used by public databases such as European Molecular Biology Laboratory. The Genbank file format is quite flexible and allows annotations, comments, and references to be included within the file. The file is plain text and thus can be read with a text editor. Genbank files often have the file extension '.gb' or '.genbank'.

1.7.2. EMBL Spec

- ABI - ABI is a binary file format containing sanger sequencing sequence and trace data. The format is used by sequencing facilities and require special readers capable of reading the file format to view the trace data and extract the sequence. The file format is difficult to parse given its binary nature and the complexity of the spec.

1.7.3.ABI Spec (PDF)

- PDB - the PDB file format is used to store both sequence information, but more importantly stores 3-dimensional structure information. This information can be used to visualize the crystal structure of a given molecule (typically a protein). PDB files are simply text files, thus can be viewed with a text editor, and often have the file extension '.pdb'.

PDB File Spec

- MDL - While not technically containing sequence data, the MDL file format is worth including in this list. The MDL mol file contains information regarding small molecules, the spec being quite similar to that of the PDB file format. The MDL mol file contains information regarding 2d (and possibly 3d) molecule structure, such as atom type and atom connectivity.

1.7.4. MDL Mol File

- BAM/SAM - The BAM/SAM format contains next-generation sequencing data. The BAM is a binary file format while the SAM file format contains the same information but is text based. These files can be analyzed and viewed by several free software tools, such as the command line open source tool SAMTools and the user interface tool IGV. Both the BAM/SAM format contain not only the sequence data for next-generation sequencing reads, but also have the capability of storing alignment data of those reads to a reference sequence.

1.7.6.SAMtools spec

- SFF - The SFF file format specifies a binary file which contains next-generation sequence information. The name stands for standard flowgram format, and contains the actual flow information used on several next-generation DNA sequencers, including Ion-Torrent and Roche's '454'

1.8. Data mining

1.8.1. Introduction

Data mining is looking for hidden, valid, and potentially useful patterns in huge data sets. Data Mining is all about discovering unsuspected/ previously unknown relationships amongst the data.

It is a multi-disciplinary skill that uses machine learning, statistics, AI and database technology.

The insights derived via Data Mining can be used for marketing, fraud detection, and scientific discover

1.8.1. Types of Data

Data mining can be performed on following types of data

Relational databases

Data warehouses

Advanced DB and information repositories

Object-oriented and object-relational databases

Transactional and Spatial databases

Heterogeneous and legacy databases

Multimedia and streaming database

Text databases

Text mining and Web mining

Data Mining Implementation Process

First, you need to understand business and client objectives. You need to define what your client wants (which many times even they do not know themselves)

Take stock of the current data mining scenario. Factor in resources, assumption, constraints, and other significant factors into your assessment.

Using business objectives and current scenario, define your data mining goals.

A good data mining plan is very detailed and should be developed to accomplish both business and data mining goals.

Data understanding:

In this phase, sanity check on data is performed to check whether its appropriate for the data mining goals.

First, data is collected from multiple data sources available in the organization.

These data sources may include multiple databases, flat filer or data cubes. There are issues like object matching and schema integration which can arise during Data Integration process. It is a quite complex and tricky process as data from various sources unlikely to match easily. For example, table A contains an entity named cust_no whereas another table B contains an entity named cust-id.

Therefore, it is quite difficult to ensure that both of these given objects refer to the same value or not. Here, Metadata should be used to reduce errors in the data integration process.

Next, the step is to search for properties of acquired data. A good way to explore the data is to answer the data mining questions (decided in business phase) using the query, reporting, and visualization tools.

Based on the results of query, the data quality should be ascertained. Missing data if any should be acquired.

1.8.2. Data preparation:

The data preparation process consumes about 90% of the time of the project.

The data from different sources should be selected, cleaned, transformed, formatted, anonymized, and constructed (if required).

Data cleaning is a process to "clean" the data by smoothing noisy data and filling in missing values.

For example, for a customer demographics profile, age data is missing. The data is incomplete and should be filled. In some cases, there could be data outliers. For instance, age has a value 300. Data could be inconsistent. For instance, name of the customer is different in different tables.

Data transformation operations change the data to make it useful in data mining. Following transformation can be applied

1.8.3. Data transformation:

Data transformation operations would contribute toward the success of the mining process.

Smoothing: It helps to remove noise from the data.

Aggregation: Summary or aggregation operations are applied to the data. I.e., the weekly sales data is aggegated to calculate the monthly and yearly total.

Generalization: In this step, Low-level data is replaced by higher-level concepts with the help of concept hierarchies. For example, the city is replaced by the county.

Normalization: Normalization performed when the attribute data are scaled up o scaled down. Example: Data should fall in the range -2.0 to 2.0 post-normalization.

Attribute construction: these attributes are constructed and included the given set of attributes helpful for data mining.

The result of this process is a final data set that can be used in modeling.

1.8.4. Modelling

Based on the business objectives, suitable modeling techniques should be selected for the prepared dataset.

Create a scenario to test check the quality and validity of the model.

Run the model on the prepared dataset.

Results should be assessed by all stakeholders to make sure that model can meet data mining objectives.

1.8.5. Evaluation:

Results generated by the data mining model should be evaluated against the business objectives.

Gaining business understanding is an iterative process. In fact, while understanding, new business requirements may be raised because of data mining.

A go or no-go decision is taken to move the model in the deployment phase.

1.8.6. Deployment:

The knowledge or information discovered during data mining process should be made easy to understand for non-technical stakeholders.

A detailed deployment plan, for shipping, maintenance, and monitoring of data mining discoveries is created.

A final project report is created with lessons learned and key experiences during the project. This helps to improve the organization's business policy

References:

1.Vasileios Lapatas, Michalis Stefanidakis, Rafael C. Jimenez, Allegra Via, Maria Victoria Schneider, Data integration in biological research: an overview, J Biol Res (Thessalon) 2015 Dec; 22(1): 9. Published online 2015 Sep 2. doi: 10.1186/s40709-015-0032-5, PMCID: PMC4557916

2. https://www.tutorialspoint.com/basics_of_computer_science/basics_of_computer_science_internet.htm

3. Gusfields G, "Algorithms on strings, trees and sequences- Computer Science and Computational Biology", Cambridge University Press, 1997

4.https://www.hpcwire.com/2006/09/22/computational_biology_challenges_and_opportunities-1/

5.https://www.tutorialspoint.com/dbms/dbms_overview.htm

6.https://www.guru99.com/data-mining-tutorial.html