1. Bioinformatics

Bioinformatics is an interdisciplinary field that develops methods and software tools for understanding biological data, in particular when the data sets are large and complex utilizing biology, computer science, information engineering, mathematics and statistics to analyze and interpret the biological data. Bioinformatics has been used for *in silico* analyses of biological queries using mathematical and statistical techniques.

Importance/objectives of bioinformatics

Common uses of bioinformatics include the identification of candidate genes and single nucleotide polymorphisms (SNPs). Often, such identification is made with the aim of better understanding the genetic basis of disease, unique adaptations, desirable properties (esp. in agricultural species), or differences between populations. Drug designing for newly occurring diseases. Phylogenetic analysis to understand the evolutionary pattern and development of particular protein/proteins, gene/genes or organism. Bioinformatics also tries to understand the organizational principles within nucleic acid and protein sequences.

2. Biological databases

Biological databases are libraries of life sciences information, collected from scientific experiments, published literature, high-throughput experiment technology, and computational analysis.[citation needed] 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 (both cellular and chromosomal), clinical effects of mutations as well as similarities of biological sequences and structures.

Biological databases can be broadly classified into sequence, structure and functional databases. Nucleic acid and protein sequences are stored in sequence databases and structure databases store solved structures of RNA and proteins. Functional databases provide information on the physiological role of gene products, for example enzyme activities, mutant phenotypes, or biological pathways.

2.1 Sequence databases/ primary databases

Sequence databases or primary databases contain nucleic acid and protein sequences.

2.1.1 Protein sequence databases

Protein Information Resource (PIR)

PIR located at Georgetown University Medical Center (GUMC), is an integrated public bioinformatics resource to support genomic and proteomic research, and scientific studies. It contains protein sequences databases. It is the first comprehensive collection of macromolecular sequences in the Atlas of Protein Sequence and Structure, published from 1964 to 1974 under Margaret Dayhoff.

UniProt

UniProt is a freely accessible database of protein sequence and functional information, many entries being derived from genome sequencing projects. It contains a large amount of information about the biological function of proteins derived from the research literature. It is maintained by the UniProt consortium, which consists of several European bioinformatics organizations and a foundation from Washington, DC, United States.

UniProt Knowledgebase (UniProtKB) is a protein database partially curated by experts, consisting of two sections: UniProtKB/Swiss-Prot (containing reviewed, manually annotated entries) and UniProtKB/TrEMBL (containing unreviewed, automatically annotated entries).

Swiss-Prot

Swiss-Prot is a manually annotated, non-redundant protein sequence database and is one of the core of UniProtKB. It combines information extracted from scientific literature and biocurator-evaluated computational analysis. The aim of UniProtKB/Swiss-Prot is to provide all known relevant information about a particular protein. Annotation is regularly reviewed to keep up with current scientific findings. The manual annotation of an entry involves detailed analysis of the protein sequence and of the scientific literature. Sequences from the same gene and the same species are merged into the same database entry. Differences between sequences are identified, and their cause documented (for example alternative splicing, natural variation, incorrect initiation sites, incorrect exon boundaries, frameshifts, unidentified conflicts).

TrEMBL

TrEMBL contains high-quality computationally analyzed records, which are enriched with automatic annotation and it is also a core of UniProtKB. It was introduced in response to increased dataflow resulting from genome projects, as the time- and labour-consuming manual annotation process of UniProtKB/Swiss-Prot could not be broadened to include all available protein sequences. The translations of annotated coding sequences in the EMBL-Bank/GenBank/DDBJ nucleotide sequence database are automatically processed and entered in UniProtKB/TrEMBL. UniProtKB/TrEMBL also contains sequences from PDB, and from gene prediction, including Ensembl, RefSeq and CCDS.

2.1.2 Nucleotide sequence databases

International Nucleotide Sequence Database (INSD) consists of the following databases. (a) DDBJ (Japan), (b) GenBank (USA) and (c) EMBL (Europe) are repositories for nucleotide sequence data from all organisms. All three accept nucleotide sequence submissions, and then exchange new and updated data on a daily basis to achieve optimal synchronization between them. These three databases are primary databases, as they house original sequence data. They collaborate with Sequence Read Archive (SRA), which archives raw reads from high-throughput sequencing instruments.

DNA Data Bank of Japan (DDBJ)

DNA Data Bank of Japan (DDBJ) is a biological database that collects DNA sequences. DDBJ began data bank activities in 1986 at NIG and remains the only nucleotide sequence data bank in

Asia. Although DDBJ mainly receives its data from Japanese researchers, it can accept data from contributors from any other country.

European Molecular Biology Laboratory (EMBL)

It is a molecular biology research institution. EMBL was created in 1974 and is an intergovernmental organization funded by public research money from its member states. Research at EMBL is conducted by approximately 85 independent groups covering the spectrum of molecular biology.

GenBank

GenBank sequence database is an open access, annotated collection of all publicly available nucleotide sequences and their protein translations. It is produced and maintained by the National Center for Biotechnology Information (NCBI; a part of the National Institutes of Health in the United States) as part of the International Nucleotide Sequence Database Collaboration (INSDC). GenBank and its collaborators receive sequences produced in laboratories throughout the world from more than 100,000 distinct organisms. The database started in 1982 by Walter Goad and Los Alamos National Laboratory. GenBank has become an important database for research in biological fields and has grown in recent years at an exponential rate by doubling roughly every 18 months.

2.2 Protein structure databases

Protein structure database is a database that is modeled around the various experimentally determined protein structures. Data included in protein structure databases often includes three-dimensional coordinates as well as experimental information, such as unit cell dimensions and angles for x-ray crystallography determined structures.

PDB

Protein Data Bank (PDB) was established in 1971 as the central archive of all experimentally determined protein structure data. Today the PDB is maintained by an international consortia collectively known as the Worldwide Protein Data Bank (wwPDB). The mission of the wwPDB is to maintain a single archive of macromolecular structural data that is freely and publicly available to the global community.

SCOP

Structural Classification of Proteins (SCOP) database is a largely manual classification of protein structural domains based on similarities of their structures and amino acid sequences. A motivation for this classification is to determine the evolutionary relationship between proteins. Proteins with the same shapes but having little sequence or functional similarity are placed in different superfamilies, and are assumed to have only a very distant common ancestor. Proteins having the same shape and some similarity of sequence and/or function are placed in "families", and are assumed to have a closer common ancestor. Similar to CATH and Pfam databases, SCOP provides a classification of individual structural domains of proteins, rather than a classification of the entire proteins which may include a significant number of different domains.

SCOP was created in 1994 in the Centre for Protein Engineering and the Laboratory of Molecular Biology.

CATH

CATH Protein Structure Classification database is a free, publicly available online resource that provides information on the evolutionary relationships of protein domains. It was created in the mid-1990s by Professor Christine Orengo and colleagues and continues to be developed by the Orengo group at University College London. CATH shares many broad features with the SCOP resource, however there are also many areas in which the detailed classification differs greatly.

Experimentally-determined protein three-dimensional structures are obtained from the Protein Data Bank and split into their consecutive polypeptide chains, where applicable. Protein domains are identified within these chains using a mixture of automatic methods and manual curation.

The domains are then classified within the CATH structural hierarchy: at the Class (C) level, domains are assigned according to their secondary structure content, i.e. all alpha, all beta, a mixture of alpha and beta, or little secondary structure; at the Architecture (A) level, information on the secondary structure arrangement in three-dimensional space is used for assignment; at the Topology/fold (T) level, information on how the secondary structure elements are connected and arranged is used; assignments are made to the Homologous superfamily (H) level if there is good evidence that the domains are related by evolution i.e. they are homologous.

DSSP

DSSP/database of secondary structure assignments of proteins program was designed by Wolfgang Kabsch and Chris Sander to standardize secondary structure assignment. DSSP is a database of secondary structure assignments (and much more) for all protein entries in the Protein Data Bank (PDB). DSSP is also the name of the program that calculates DSSP entries from PDB entries. It means there are actually two ways of looking at DSSP. First of all there are the precalculated DSSP files for each PDB entry. And then there's the application called DSSP that can create these files. The DSSP program works by calculating the most likely secondary structure assignment given the 3D structure of a protein.

This means you do need to have a full and valid 3D structure for a protein to be able to calculate the secondary structure. There's no magic in DSSP, so e.g. it cannot guess the secondary structure for a mutated protein for which you don't have the 3D structure. And, again, DSSP does not predict secondary structures, it just extracts this information from the 3D coordinates. The DSSP program defines secondary structure, geometrical features and solvent exposure of proteins, given atomic coordinates in Protein Data Bank format (PDB) or macromolecular Crystallographic Information File format. (mmCIF). In 1995 the format of the DSSP output files had to be changed. These changes are listed in this page, and are separately available. In the beginning of this century Elmar Krieger made a series of corrections and adaptations to PDB file format modifications. In 2011 Maarten Hekkelman completely rewrote DSSP. The original DSSP is from now on referred to as DSSPold. In 2017 the DSSP format was extended, to hold the 4-character long chain IDs in the mmCIF file format.

CCDC

Cambridge Crystallographic Data Centre (CCDC) are publicly available for download at the point of publication or at consent from the depositor. They are also scientifically enriched and included in the database used by software offered by the centre. Targeted subsets of the CSD are also freely available to support teaching and other activities. Cambridge Structural Database (CSD) is both a repository and a validated and curated resource for the three-dimensional structural data of molecules generally containing at least carbon and hydrogen, comprising a wide range of organic, metal-organic and organometallic molecules. The specific entries are complementary to the other crystallographic databases such as the Protein Data Bank (PDB), Inorganic Crystal Structure Database and International Centre for Diffraction Data. The data, typically obtained by X-ray crystallography and less frequently by electron diffraction or neutron diffraction, and submitted by crystallographers and chemists from around the world, are freely accessible (as deposited by authors) on the Internet via the CSD's parent organization's website (CCDC, Repository). The CSD is overseen by the not-for-profit incorporated company called the Cambridge Crystallographic Data Centre, CCDC. The CCDC grew out of the activities of the crystallography group led by Olga Kennard in the Department of Organic, Inorganic and Theoretical Chemistry of the University of Cambridge. From 1965, the group began to collect published bibliographic, chemical and crystal structure data for all small molecules studied by X-ray or neutron diffraction.

2.3 Functional databases

KEGG pathway Database

KEGG (Kyoto Encyclopedia of Genes and Genomes) is a collection of databases dealing with genomes, biological pathways, diseases, drugs, and chemical substances. KEGG is utilized for bioinformatics research and education, including data analysis in genomics, metagenomics, metabolomics and other omics studies, modeling and simulation in systems biology, and translational research in drug development. KEGG database project was initiated in 1995 by Minoru Kanehisa at Kyoto University. KEGG is a "computer representation" of the biological system. Databases of KEGG are categorized into systems, genomic, chemical, and health information.

- Systems information
  - PATHWAY — pathway maps for cellular and organismal functions
  - MODULE — modules or functional units of genes
  - BRITE — hierarchical classifications of biological entities
- Genomic information
  - GENOME — complete genomes
  - GENES — genes and proteins in the complete genomes
  - ORTHOLOGY — ortholog groups of genes in the complete genomes
- Chemical information
  - COMPOUND, GLYCAN — chemical compounds and glycans
  - REACTION, RPAIR, RCLASS — chemical reactions
  - ENZYME — enzyme nomenclature

Health information
DISEASE — human diseases
DRUG — approved drugs
ENVIRON — crude drugs and health-related substances

Swiss-2Dpage

SWISS-2DPAGE (Two-dimensional Polyacrylamide Gel Electrophoresis Database) database which gathers data on proteins identified on various two-dimensional polyacrylamide gel electrophoresis (2-D PAGE) and SDS-PAGE maps. Established in 1993 and maintained collaboratively by the Central Clinical Chemistry Laboratory of the Geneva University Hospital and the Swiss Institute of Bioinformatics (SIB). Each SWISS-2DPAGE entry contains data on one protein, including mapping procedures, physiological and pathological data and bibliographical references, as well as several 2-D PAGE images showing the protein location. Links are also provided to other databases such as SWISS-PROT, EMBL, PROSITE and OMIM. The database has been set up on a server which may be accessed from any computer connected to the internet and it also makes it possible to display the theoretical location of proteins, the positions of which are not yet known on the 2-D PAGE.

COGS

The database of Clusters of Orthologous Groups of proteins (COGs) is an attempt on a phylogenetic classification of the proteins encoded in 21 complete genomes of bacteria, archaea and eukaryotes. The COGs were constructed by applying the criterion of consistency of genome-specific best hits to the results of an exhaustive comparison of all protein sequences from these genomes. The database comprises 2091 COGs that include 56–83% of the gene products from each of the complete bacterial and archaeal genomes and ~35% of those from the yeast *Saccharomyces cerevisiae* genome. The COG database is accompanied by the COGNITOR program that is used to fit new proteins into the COGs and can be applied to functional and phylogenetic annotation of newly sequenced genomes. Each COG consists of a group of proteins found to be orthologous across at least three lineages and likely corresponds to an ancient conserved domain. For more information check out the NCBI COG website. Since the COG database is significantly smaller than the NCBI non-redundant (NR) database, it provides a fast alternative for rapidly describing the functional characteristics of one microbe or a community of microbes.

PROSITE

PROSITE is a protein database. It consists of entries describing the protein families, domains and functional sites as well as amino acid patterns and profiles in them. These are manually curated by a team of the Swiss Institute of Bioinformatics and tightly integrated into Swiss-Prot protein annotation. PROSITE was created in 1988 by Amos Bairoch, who directed the group for more than 20 years. PROSITE's uses include identifying possible functions of newly discovered proteins and analysis of known proteins for previously undetermined activity. Properties from well-studied genes can be propagated to biologically related organisms, and for different or poorly known genes biochemical functions can be predicted from similarities. PROSITE offers

tools for protein sequence analysis and motif detection (see sequence motif, PROSITE patterns). It is part of the ExPASy proteomics analysis servers.
The database ProRule builds on the domain descriptions of PROSITE. It provides additional information about functionally or structurally critical amino acids. The rules contain information about biologically meaningful residues, like active sites, substrate- or co-factor-binding sites, posttranslational modification sites or disulfide bonds, to help function determination. These can automatically generate annotation based on PROSITE motifs.

2.4 Secondary/ sequence cluster database

ProDom

ProDom is a protein domain family database constructed automatically by clustering homologous segments. The ProDom building procedure MKDOM2 is based on recursive PSI-BLAST searches. The source protein sequences are non-fragmentary sequences derived from UniProtKB (SWISS-PROT and TrEMBL databases). ProDom was first established in 1993 and maintained by the Laboratoire de Génétique Cellulaire and the Laboratoire de Interactions Plantes-Microorganismes (INRA/CNRS) in Toulouse. It is now maintained by the PRABI (bioinformatics center of Rhone-Alpes). The ProDom database consists of domain family entries. Each entry provides a multiple sequence alignment of homologous domains and a family consensus sequence. ProDom families are currently cross-referenced to the following databases: GO (Gene Ontology), INTERPRO, PROSITE, PFAMA (Pfam-A protein domain), PDB.

SYSTERS
SYSTERS project aims to provide a meaningful partitioning of the whole protein sequence space by a fully automatic procedure. A refined two-step algorithm assigns each protein to a family and a superfamily. The sequence data underlying SYSTERS release 4 now comprise several protein sequence databases derived from completely sequenced genomes (ENSEMBL, TAIR, SGD and GeneDB), in addition to the comprehensive Swiss-Prot/TrEMBL databases. The SYSTERS web server (http://systers.molgen.mpg.de) provides access to 158 153 SYSTERS protein families. To augment the automatically derived results, information from external databases like Pfam and Gene Ontology are added to the web server. Furthermore, users can retrieve pre-processed analyses of families like multiple alignments and phylogenetic trees. New query options comprise a batch retrieval tool for functional inference about families based on automatic keyword extraction from sequence annotations. A new access point, PhyloMatrix, allows the retrieval of phylogenetic profiles of SYSTERS families across organisms with completely sequenced genomes.

ProtoMap
ProtoMap offers an exhaustive classification of all proteins in the SWISS-PROT database, into groups of related proteins. The classification is based on analysis of all pairwise similarities among protein sequences. The analysis makes essential use of transitivity to identify homologies among proteins. Within each group of the classification, every two members are either directly or

transitively related. However, transitivity is applied restrictively in order to prevent unrelated proteins from clustering together. The classification is done at different levels of confidence, and yields a hierarchical organization of all proteins. The resulting classification splits the protein space into well-defined groups of proteins, which are closely correlated with natural biological families and superfamilies. Many clusters contain protein sequences that are not classified by other databases. The hierarchical organization suggested by our analysis may help in detecting finer subfamilies in families of known proteins. In addition it brings forth interesting relationships between protein families, upon which local maps for the neighborhood of protein families can be sketched.

3. MS Excel

Microsoft Excel is a spreadsheet developed by Microsoft for Windows, macOS, Android and iOS. It features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications. It has been a very widely applied spreadsheet for these platforms, especially since version 5 in 1993, and it has replaced Lotus 1-2-3 as the industry standard for spreadsheets. Excel forms part of the Microsoft Office suite of software.

Features

1. Microsoft Excel has the basic features of all spreadsheets, using a grid of cells arranged in numbered rows and letter-named columns to organize data manipulations like arithmetic operations.
2. It has a battery of supplied functions to answer statistical, engineering and financial needs.
3. It can display data as line graphs, histograms and charts, and with a very limited three-dimensional graphical display.
4. It allows sectioning of data to view its dependencies on various factors for different perspectives (using pivot tables and the scenario manager).
5. It has a programming aspect, Visual Basic for Applications (VBA), allowing the user to employ a wide variety of numerical methods, for example, for solving differential equations of mathematical physics, and then reporting the results back to the spreadsheet.
6. It also has a variety of interactive features allowing user interfaces that can completely hide the spreadsheet from the user, so the spreadsheet presents itself as a so-called application, or decision support system (DSS), via a custom-designed user interface, for example, a stock analyzer, or in general, as a design tool that asks the user questions and provides answers and reports.
7. An Excel application can automatically poll external databases and measuring instruments using an update schedule, analyze the results, make a Word report or PowerPoint slide show, and e-mail these presentations on a regular basis to a list of participants.

Application in Agriculture

1. Spreadsheet could be utilized to fill field data directly to minimize the effort to computerize the data for further analysis.

2. Statistical analyses as correlation, regression, anova could be performed easily to analyze properties of data.
3. Graphs, histograms and plots could be prepared easily to visualize and interpret data in meaningful way.

4. MS word

Microsoft Word is a word processor developed by Microsoft. It was first released on October 25, 1983. Commercial versions of Word are licensed as a standalone product or as a component of Microsoft Office. Microsoft Word's native file formats are denoted either by a .doc or .docx filename extension.
Among its features, Word includes a built-in spell checker, a thesaurus, a dictionary, and utilities for manipulating and editing text. The following are some aspects of its feature set.
1. Templates
Several later versions of Word include the ability for users to create their own formatting templates, allowing them to define a file in which the title, heading, paragraph, and other element designs differ from the standard Word templates. Users can find how to do this under the Help section located near the top right corner (Word 2013 on Windows 8).
2. Image formats

Word can import and display images in common bitmap formats such as JPG and GIF. It can also be used to create and display simple line-art. Microsoft Word added support for the common SVG vector image format in 2017 for Office 365 ProPlus subscribers and this functionality was also included in the Office 2019 release.
3. WordArt
WordArt enables drawing text in a Microsoft Word document such as a title, watermark, or other text, with graphical effects such as skewing, shadowing, rotating, stretching in a variety of shapes and colors and even including three-dimensional effects. Users can apply formatting effects such as shadow, bevel, glow, and reflection to their document text as easily as applying bold or underline. Users can also spell-check text that uses visual effects, and add text effects to paragraph styles.
4. Macros
A Macro is a rule of pattern that specifies how a certain input sequence (often a sequence of characters) should be mapped to an output sequence according to a defined process. Frequently used or repetitive sequences of keystrokes and mouse movements can be automated. Like other Microsoft Office documents, Word files can include advanced macros and even embedded programs. The language was originally WordBasic, but changed to Visual Basic for Applications as of Word 97.
5. Layout issues
Since Word 2010, the program now has advanced typesetting features which can be enabled: OpenType ligatures, kerning, and hyphenation. Other layout deficiencies of Word include the inability to set crop marks or thin spaces. Various third-party workaround utilities have been developed.

6. Bullets and numbering
Microsoft Word supports bullet lists and numbered lists. It also features a numbering system that helps add correct numbers to pages, chapters, headers, footnotes, and entries of tables of content; these numbers automatically change to correct ones as new items are added or existing items are deleted. Bullets and numbering can be applied directly to paragraphs and convert them to lists.
7. Tables
Users can also create tables in Word. Depending on the version, Word can perform simple calculations along with support for formulas and equations as well.

5. PowerPoint

Microsoft PowerPoint is a presentation program, created by Robert Gaskins and Dennis Austin, released in 1987. PowerPoint became a component of the Microsoft Office suite, first offered in 1989 for Macintosh and in 1990 for Windows. The PowerPoint file extensions are .ppt and .pptx file format. PowerPoint is a computer program that allows to create and show slides to support a presentation. It can combine text, graphics and multi-media content to create professional presentations. As a presentation tool PowerPoint can be used to:

- organise and structure your presentation;
- create a professional and consistent format;
- provide an illustrative backdrop for the content of your presentation;
- animate your slides to give them greater visual impact.

PowerPoint has become enormously popular and you are likely to have seen it used by your lecturers and fellow students or in a presentation outside of the University. Learning to present with PowerPoint will increase your employability as it is the world's most popular presentational software. Used well, PowerPoint can improve the clarity of your presentations and help you to illustrate your message and engage your audience. The strategies contained in this study guide will help you to use PowerPoint effectively in any type of presentation.

6. Multiplicity and redundancy of data

Data and reports often present the same data in multiple forms when reporting different intervention groups, time points, and outcome measures. Although this multiplicity has always been a challenge and adds biasness to the data. Data redundancy is the existence of data that is additional to the actual data and permits correction of errors in stored or transmitted data. The additional data can simply be a complete copy of the actual data, or only select pieces of data that allow detection of errors and reconstruction of lost or damaged data up to a certain level. While different in nature, data redundancy also occurs in database systems that have values repeated unnecessarily in one or more records or fields, within a table, or where the field is replicated/repeated in two or more tables. Often this is found in Unnormalized database designs and results in the complication of database management, introducing the risk of corrupting the data, and increasing the required amount of storage.

7. Data integration

Data integration involves combining data residing in different sources and providing users with a unified view of them. This process becomes significant in a variety of situations, which include both commercial (such as when two similar companies need to merge their databases) and scientific (combining research results from different bioinformatics repositories, for example) domains. Data integration appears with increasing frequency as the volume (that is, big data) and the need to share existing data explodes. It has become the focus of extensive theoretical work, and numerous open problems remain unsolved. Data integration encourages collaboration between internal as well as external users.

8. Data analysis

Data analysis is a process of inspecting, cleansing, transforming and modeling data with the goal of discovering useful information, informing conclusion and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. Data analysis is a process for obtaining raw data and converting it into information useful for decision-making by users. Data is collected and analyzed to answer questions, test hypotheses or disprove theories.

9. SPSS

SPSS Statistics is a software package used for interactive, or batched, statistical analysis. The current versions (2015) are named IBM SPSS Statistics. The software name originally stood for **Statistical Package for the Social Sciences** (SPSS), reflecting the original market, then later changed to **Statistical Product and Service Solutions**. SPSS is a widely used program for statistical analysis in social science. It is also used by agriculture researchers, market researchers, health researchers, survey companies, government, education researchers, marketing organizations, data miners, and others. In addition to statistical analysis, data management (case selection, file reshaping, creating derived data) and data documentation (a metadata dictionary is stored in the data file) are features of the base software.
Statistics included in the base software:
1. Descriptive statistics: Cross tabulation, Frequencies, Descriptives, Explore, Descriptive Ratio Statistics
2. Bivariate statistics: Means, t-test, ANOVA, Correlation (bivariate, partial, distances), Nonparametric tests, Bayesian
3. Prediction for numerical outcomes: Linear regression
4. Prediction for identifying groups: Factor analysis, cluster analysis (two-step, K-means, hierarchical), Discriminant
5. Geo spatial analysis, simulation
6. R extension (GUI), Python

10. SAS

SAS ("Statistical Analysis System") is a statistical software suite developed by SAS Institute for data management, advanced analytics, multivariate analysis, business intelligence, criminal investigation, and predictive analytics. SAS was developed at North Carolina State University from 1966 until 1976, when SAS Institute was incorporated. SAS is a software suite that can mine, alter, manage and retrieve data from a variety of sources and perform statistical analysis on it. SAS provides a graphical point-and-click user interface for non-technical users and more through the SAS language.
SAS programs have DATA steps, which retrieve and manipulate data, and PROC steps, which analyze the data. Each step consists of a series of statements.
The DATA step has executable statements that result in the software taking an action, and declarative statements that provide instructions to read a data set or alter the data's appearance. The DATA step has two phases: compilation and execution. In the compilation phase, declarative statements are processed and syntax errors are identified. Afterwards, the execution phase processes each executable statement sequentially. Data sets are organized into tables with rows called "observations" and columns called "variables". Additionally, each piece of data has a descriptor and a value.
The PROC step consists of PROC statements that call upon named procedures. Procedures perform analysis and reporting on data sets to produce statistics, analyses, and graphics. There are more than 300 named procedures and each one contains a substantial body of programming and statistical work. PROC statements can also display results, sort data or perform other operations

12. Sequence alignment

In bioinformatics, a sequence alignment is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. Gaps are inserted between the residues so that identical or similar characters are aligned in successive columns.
If two sequences in an alignment share a common ancestor, mismatches can be interpreted as point mutations and gaps as indels (that is, insertion or deletion mutations) introduced in one or both lineages in the time since they diverged from one another. In sequence alignments of proteins, the degree of similarity between amino acids occupying a particular position in the sequence can be interpreted as a rough measure of how conserved a particular region or sequence motif is among lineages. The absence of substitutions, or the presence of only very conservative substitutions (that is, the substitution of amino acids whose side chains have similar biochemical properties) in a particular region of the sequence, suggest that this region has structural or functional importance. Although DNA and RNA nucleotide bases are more similar to each other than are amino acids, the conservation of base pairs can indicate a similar functional or structural role.

12.1 Global alignment
Global alignments, which attempt to align every residue in every sequence, are most useful when the sequences in the query set are similar and of roughly equal size. (This does not mean global

alignments cannot start and/or end in gaps.) A general global alignment technique is the Needleman–Wunsch algorithm, which is based on dynamic programming. The Needleman–Wunsch algorithm is an algorithm used in bioinformatics to align protein or nucleotide sequences. It was one of the first applications of dynamic programming to compare biological sequences. The algorithm was developed by Saul B. Needleman and Christian D. Wunsch and published in 1970. The algorithm essentially divides a large problem (e.g. the full sequence) into a series of smaller problems, and it uses the solutions to the smaller problems to find an optimal solution to the larger problem. It is also sometimes referred to as the optimal matching algorithm and the global alignment technique. The Needleman–Wunsch algorithm is still widely used for optimal global alignment, particularly when the quality of the global alignment is of the utmost importance. The algorithm assigns a score to every possible alignment, and the purpose of the algorithm is to find all possible alignments having the highest score. However, the algorithm is expensive with respect to time and space, proportional to the product of the length of two sequences and hence is not suitable for long sequences.

Match = +1
Mismatch = -1
Gap = -2, -10

Sequences	Best alignments		
GCATGCU	GCATG-CU	GCA-TGCU	GCAT-GCU
GATTACA	G-ATTACA	G-ATTACA	G-ATTACA

12.2 Local alignment

Local alignments are more useful for dissimilar sequences that are suspected to contain regions of similarity or similar sequence motifs within their larger sequence context. The Smith–Waterman algorithm is a general local alignment method based on the same dynamic programming scheme but with additional choices to start and end at any place.

The Smith–Waterman algorithm performs local sequence alignment; that is, for determining similar regions between two strings of nucleic acid sequences or protein sequences. Instead of looking at the entire sequence, the Smith–Waterman algorithm compares segments of all possible lengths and optimizes the similarity measure. The algorithm was first proposed by Temple F. Smith and Michael S. Waterman in 1981. Like the Needleman–Wunsch algorithm, of which it is a variation, Smith–Waterman is a dynamic programming algorithm. As such, it has the desirable property that it is guaranteed to find the optimal local alignment with respect to the scoring system being used (which includes the substitution matrix and the gap-scoring scheme). The main difference to the Needleman–Wunsch algorithm is that negative scoring matrix cells are set to zero, which renders the (thus positively scoring) local alignments visible. Traceback procedure starts at the highest scoring matrix cell and proceeds until a cell with score zero is encountered, yielding the highest scoring local alignment. Because of its quadratic complexity in time and space, it often cannot be practically applied to large-scale problems and is replaced in

favor of less general but computationally more efficient alternatives such as (Gotoh, 1982), (Altschul and Erickson, 1986), and (Myers and Miller, 1988).

Match = +1

Sequences	Best alignments
GCATGCU	GC TGC
GCTTGCA	GC TGC

Importance of local alignment:

In recent years, genome projects conducted on a variety of organisms generated massive amounts of sequence data for genes and proteins, which requires computational analysis. Sequence alignment shows the relations between genes or between proteins, leading to a better understanding of their homology and functionality. Sequence alignment can also reveal conserved domains and motifs.

One motivation for local alignment is the difficulty of obtaining correct alignments in regions of low similarity between distantly related biological sequences, because mutations have added too much 'noise' over evolutionary time to allow for a meaningful comparison of those regions.

Another motivation for using local alignments is that there is a reliable statistical model (developed by Karlin and Altschul) for optimal local alignments. The alignment of unrelated sequences tends to produce optimal local alignment scores which follow an extreme value distribution.

12.3 Gap penalty

Gap penalty designates scores for insertion or deletion. A simple gap penalty strategy is to use fixed score for each gap. In biology, however, the score needs to be counted differently for practical reasons. On one hand, partial similarity between two sequences is a common phenomenon; on the other hand, a single gene mutation event can result in insertion of a single long gap. Therefore, connected gaps forming a long gap usually is more favored than multiple scattered, short gaps. In order to take this difference into consideration, the concepts of gap opening and gap extension have been added to the scoring system. The gap opening score is usually higher than the gap extension score. For instance, the default parameter in EMBOSS Water are: gap opening = 10, gap extension = 0.5.

12.4 Dot Plots

The dot-matrix approach, which implicitly produces a family of alignments for individual sequence regions, is qualitative and conceptually simple, though time-consuming to analyze on a large scale. In the absence of noise, it can be easy to visually identify certain sequence features—such as insertions, deletions, repeats, or inverted repeats—from a dot-matrix plot. To construct a dot-matrix plot, the two sequences are written along the top row and leftmost column of a two-dimensional matrix and a dot is placed at any point where the characters in the appropriate

columns match—this is a typical recurrence plot. Some implementations vary the size or intensity of the dot depending on the degree of similarity of the two characters, to accommodate conservative substitutions. The dot plots of very closely related sequences will appear as a single line along the matrix's main diagonal.
Problems with dot plots as an information display technique include: noise, lack of clarity, non-intuitiveness, difficulty extracting match summary statistics and match positions on the two sequences. There is also much wasted space where the match data is inherently duplicated across the diagonal and most of the actual area of the plot is taken up by either empty space or noise, and, finally, dot-plots are limited to two sequences.
Dot plots can also be used to assess repetitiveness in a single sequence. A sequence can be plotted against itself and regions that share significant similarities will appear as lines off the main diagonal. This effect can occur when a protein consists of multiple similar structural domains.

13. BLAST

In bioinformatics, BLAST (basic local alignment search tool) is an algorithm and program for comparing primary biological sequence information, such as the amino-acid sequences of proteins or the nucleotides of DNA and/or RNA sequences. A BLAST search enables a researcher to compare a subject protein or nucleotide sequence (called a query) with a library or database of sequences, and identify library sequences that resemble the query sequence above a certain threshold.

Different types of BLASTs are available according to the query sequences and the target databases. For example, following the discovery of a previously unknown gene in the mouse, a scientist will typically perform a BLAST search of the human genome to see if humans carry a similar gene; BLAST will identify sequences in the human genome that resemble the mouse gene based on similarity of sequence.
It is one of the most widely used bioinformatics programs for sequence searching. It addresses a fundamental problem in bioinformatics research. The heuristic algorithm it uses is much faster than other approaches, such as calculating an optimal alignment. This emphasis on speed is vital to making the algorithm practical on the huge genome databases currently available.
BLAST came from the 1990 stochastic model of Samuel Karlin and Stephen Altschul. They "proposed a method for estimating similarities between the known DNA sequence of one organism with that of another".

Steps of BLAST:
Input
Input sequences (in FASTA or Genbank format) and weight matrix.
Output
BLAST output can be delivered in a variety of formats. These formats include HTML, plain text, and XML formatting. For NCBI's web-page, the default format for output is HTML. When performing a BLAST on NCBI, the results are given in a graphical format showing the hits

found, a table showing sequence identifiers for the hits with scoring related data, as well as alignments for the sequence of interest and the hits received with corresponding BLAST scores for these. The easiest to read and most informative of these is probably the table.

Types of BLAST:
BLAST is actually a family of programs (all included in the blastall executable). These include:

Nucleotide-nucleotide BLAST (blastn)
This program, given a DNA query, returns the most similar DNA sequences from the DNA database that the user specifies.
Protein-protein BLAST (blastp)
This program, given a protein query, returns the most similar protein sequences from the protein database that the user specifies.
Position-Specific Iterative BLAST (PSI-BLAST) (blastpgp)
This program is used to find distant relatives of a protein. First, a list of all closely related proteins is created. These proteins are combined into a general "profile" sequence, which summarises significant features present in these sequences. A query against the protein database is then run using this profile, and a larger group of proteins is found. This larger group is used to construct another profile, and the process is repeated.
By including related proteins in the search, PSI-BLAST is much more sensitive in picking up distant evolutionary relationships than a standard protein-protein BLAST.
Nucleotide 6-frame translation-protein (blastx)
This program compares the six-frame conceptual translation products of a nucleotide query sequence (both strands) against a protein sequence database.
Nucleotide 6-frame translation-nucleotide 6-frame translation (tblastx)
This program is the slowest of the BLAST family. It translates the query nucleotide sequence in all six possible frames and compares it against the six-frame translations of a nucleotide sequence database. The purpose of tblastx is to find very distant relationships between nucleotide sequences.
Protein-nucleotide 6-frame translation (tblastn)
This program compares a protein query against the all six reading frames of a nucleotide sequence database.
Large numbers of query sequences (megablast)
When comparing large numbers of input sequences via the command-line BLAST, "megablast" is much faster than running BLAST multiple times. It concatenates many input sequences together to form a large sequence before searching the BLAST database, then post-analyzes the search results to glean individual alignments and statistical values.

Of these programs, BLASTn and BLASTp are the most commonly used because they use direct comparisons, and do not require translations. However, since protein sequences are better conserved evolutionarily than nucleotide sequences, tBLASTn, tBLASTx, and BLASTx, produce more reliable and accurate results when dealing with coding DNA. They also enable one

to be able to directly see the function of the protein sequence, since by translating the sequence of interest before searching often gives you annotated protein hits.

Significance of BLAST results:

BLAST can be used for several purposes. These include identifying species, locating domains, establishing phylogeny, DNA mapping, and comparison.
1. Identifying species: With the use of BLAST, you can possibly correctly identify a species or find homologous species. This can be useful, for example, when you are working with a DNA sequence from an unknown species.
2. Locating domains: When working with a protein sequence you can input it into BLAST, to locate known domains within the sequence of interest.
3. Establishing phylogeny: Using the results received through BLAST you can create a phylogenetic tree using the BLAST web-page. Phylogenies based on BLAST alone are less reliable than other purpose-built computational phylogenetic methods, so should only be relied upon for "first pass" phylogenetic analyses.
4. DNA mapping: When working with a known species, and looking to sequence a gene at an unknown location, BLAST can compare the chromosomal position of the sequence of interest, to relevant sequences in the database(s). NCBI has a "Magic-BLAST" tool built around BLAST for this purpose.
5. Comparison: When working with genes, BLAST can locate common genes in two related species, and can be used to map annotations from one organism to another.

FASTA format
In bioinformatics and biochemistry, the FASTA format is a text-based format for representing either nucleotide sequences or amino acid (protein) sequences, in which nucleotides or amino acids are represented using single-letter codes. The format also allows for sequence names and comments to precede the sequences. Now became a near universal standard in the field of bioinformatics. The simplicity of FASTA format makes it easy to manipulate and parse sequences. FASTA format has filename extensions as .fasta, .fna, .ffn, .faa, .frn. Developed by David J. Lipman and William R. Pearson in 1985.

14. FASTA

FASTA is a DNA and protein sequence alignment software package first described by David J. Lipman and William R. Pearson in 1985. The current FASTA package contains programs for protein:protein, DNA:DNA, protein:translated DNA (with frameshifts), and ordered or unordered peptide searches. Recent versions of the FASTA package include special translated search algorithms that correctly handle frameshift errors (which six-frame-translated searches do not handle very well) when comparing nucleotide to protein sequence data. In addition to rapid heuristic search methods, the FASTA package provides SSEARCH, an implementation of the optimal Smith–Waterman algorithm. A major focus of the package is the calculation of accurate similarity statistics, so that biologists can judge whether an alignment is likely to have occurred

by chance, or whether it can be used to infer homology. The FASTA package is available from the University of Virginia and the European Bioinformatics Institute. The FASTA file format used as input for this software is now largely used by other sequence database search tools (such as BLAST) and sequence alignment programs (Clustal, T-Coffee, etc.).

Process:

FASTA takes a given nucleotide or amino acid sequence and searches a corresponding sequence database by using local sequence alignment to find matches of similar database sequences.

The FASTA program follows a largely heuristic method which contributes to the high speed of its execution. It initially observes the pattern of word hits, word-to-word matches of a given length, and marks potential matches before performing a more time-consuming optimized search using a Smith–Waterman type of algorithm.

Difference between BLAST and FASTA:

Both BLAST (Basic Local Alignment Search Tool) and FASTA (Fast All) are used to find matches of similar database sequences. Both the software have been shown to perform equally well except for a few differences.

The major difference between BLAST and FASTA are:

1. BLAST uses local alignment, while FASTA initially uses local alignment then extends to global alignment.
2. BLAST searches similarities in local alignment by comparing individual residues in two sequences, while FASTA searches similarities in local alignment by comparing sequence patterns or words.
3. BLAST better for similarity searching in closely matched or locally optimal sequences while FASTA is better for similarity searching in less similar sequences.
4. BLAST works best for protein sequences while FASTA is for nucleotide sequences.
5. Seeding Process. Both BLAST and FASTA uses different seeding process. BLAST uses a substitution matrix to find matching words, whereas FASTA identifies identical matching words using the hashing procedure.
6. Search Window size. FASTA has a smaller window size and gives more sensitive results than BLAST with better coverage rate for homologs.
7. BLAST uses low-complexity masking which means it may have higher specificity than FASTA because potential false positives are reduced.
8. BLAST sometimes gives multiple best-scoring alignments from the same sequence, FASTA returns only one final alignment.
9. In BLAST gaps between query and target sequences are not allowed while allowed in FASTA.
10. BLAST is faster than FASTA

15. Sequence assembly

In bioinformatics, sequence assembly refers to aligning and merging fragments from a longer DNA sequence in order to reconstruct the original sequence. This is needed as DNA sequencing technology cannot read whole genomes in one go, but rather reads small pieces of between 20 and 30,000 bases, depending on the technology used. Typically the short fragments, called reads, result from shotgun sequencing genomic DNA, or gene transcript (ESTs).

{The problem of sequence assembly can be compared to taking many copies of a book, passing each of them through a shredder with a different cutter, and piecing the text of the book back together just by looking at the shredded pieces. Besides the obvious difficulty of this task, there are some extra practical issues: the original may have many repeated paragraphs, and some shreds may be modified during shredding to have typos. Excerpts from another book may also be added in, and some shreds may be completely unrecognizable.}

Genome assemblers

The first sequence assemblers began to appear in the late 1980s and early 1990s as variants of simpler sequence alignment programs to piece together vast quantities of fragments generated by automated sequencing instruments called DNA sequencers. As the sequenced organisms grew in size and complexity (from small viruses over plasmids to bacteria and finally eukaryotes), the assembly programs used in these genome projects needed increasingly sophisticated strategies to handle: (i) terabytes of sequencing data which need processing on computing clusters; (ii) identical and nearly identical sequences (known as repeats) which can, in the worst case, increase the time and space complexity of algorithms quadratically; (iii) errors in the fragments from the sequencing instruments, which can confound assembly.

EST assemblers

Expressed sequence tag or EST assembly was an early strategy, dating from the mid-1990s to the mid-2000s, to assemble individual genes rather than whole genomes. The problem differs from genome assembly in several ways. The input sequences for EST assembly are fragments of the transcribed mRNA of a cell and represent only a subset of the whole genome. A number of algorithmical problems differ between genome and EST assembly. For instance, genomes often have large amounts of repetitive sequences, concentrated in the intergenic regions. Transcribed genes contain many fewer repeats, making assembly somewhat easier. On the other hand, some genes are expressed (transcribed) in very high numbers (e.g., housekeeping genes), which means that unlike whole-genome shotgun sequencing, the reads are not uniformly sampled across the genome.

EST assembly is made much more complicated by features like (cis-) alternative splicing, trans-splicing, single-nucleotide polymorphism, and post-transcriptional modification. Beginning in 2008 when RNA-Seq was invented, EST sequencing was replaced by this far more efficient technology, described under de novo transcriptome assembly.

De-novo vs. mapping assembly

In sequence assembly, two different types can be distinguished:
de-novo: assembling short reads to create full-length (sometimes novel) sequences, without using a template (see de novo sequence assemblers, de novo transcriptome assembly)
mapping: assembling reads against an existing backbone sequence, building a sequence that is similar but not necessarily identical to the backbone sequence
In terms of complexity and time requirements, de-novo assemblies are orders of magnitude slower and more memory intensive than mapping assemblies. This is mostly due to the fact that the assembly algorithm needs to compare every read with every other read (an operation that has a naive time complexity of O(n2).
Handling repeats in de-novo assembly requires the construction of a graph representing neighboring repeats. Such information can be derived from reading a long fragment covering the repeats in full or only its two ends. On the other hand, in a mapping assembly, parts with multiple or no matches are usually left for another assembling technique to look into.