**Summary**

The amount of available data to researchers in biomedical studies has grown so rapidly in recent years that grasping the current state of the art without some expertise and understanding of data analytics and bioinformatics is becoming more challenging. The central source for researcher and students to understand the fundamentals of bioinformatics in user friendly way is still extinct. An Introduction to Bioinformatics with Python: A Guidebook for Researchers leads the reader through the fundamentals of computational data analysis used in modern biology. Readers will learn how to organize appropriate analyses of biological information and how to implement these analyses using the Python programming, even if they have no prior knowledge with mathematics or computing. This is accomplished by utilizing Python to solve research problems using numerous biological datasets in a qualitative case study.  Broadly employed statistical and biological methods will explained, including distance measures, similarity search, evaluation tree matrices.  The core biological concepts including, HMM for biological problems, Drug discovery, drug target interaction, inferencing of phylogenetic tree will be explained for basic understanding without prior knowledge of bioinformatics.  These methodologies are then employed in the research studies as needed to demonstrate how they might be used to solve research issues.

**Objective**

The aim of this book is to provide a hands-on course in computational biology for students or researchers who have never worked with computer programing. It will also explain the theoretical foundation for computation biological techniques utilized throughout the work, starting with the fundamentals. Walkthroughs of data analysis tasks using python and example datasets are presented. All python instructions are shown and explained so that the reader may complete these tasks on their own. It will also give the practical lessons oriented toward cancer research challenges that can be used to a variety of computational biology and medical research problems.

**Literature Survey**

|  |  |
| --- | --- |
| Book Name | Description |
| Essential Bioinformatics | Essential Bioinformatics is a book related to essential topics for bioinformatics. This book consists of six chapter related to sequence alignment, HMM model, similarity search for genomes and protein prediction. It will not cover the many areas of bioinformatics like phylogeny, biological data extraction and integration. |
| Bioinformatics For Dummies | This book provides the basic knowledge of bioinformatics principles. The theoretical concepts are explained in very simple English. It also discusses the different tools for biological data analysis rather then the code for user hands on practice. |
| Bioinformatics: Sequence and Genome Analysis | This book deals with the computational methods for genome sequencing and proteomic data. This book is divided into 13 chapters that discussed the sequence alignment methods and statistical analysis methods. This book also elaborates the use of different tools for biological data analysis. |
| Introduction To Bioinformatics | Introduction to bioinformatics is an introductory book that explains the concepts of bioinformatics. It is divided into 10 chapters that explains the almost all basic concepts of bioinformatics related to genes, genome sequences, sequence alignment and phylogenetic. It also touches the Machine learning and Artificial Intelligence. Reader must require the prior knowledge of computing and bioinformatics to understand this book. |
| Bioinformatics: Genes, Proteins and Computers | This book discussed the biological concepts with computational techniques to solve complex problems of bioinformatics computationally. The topics covered in this book are sequence compression, sequence alignment, sequence search and protein prediction. It also covers the above topics with protein sequences. This is an intermediate level book that required some prior knowledge pf bioinformatics. |
| Introduction to Bioinformatics with Python: | The massive amount of material on bioinformatic topics have been published to date that describes the concepts of bioinformatics from beginner to advance level. But the explanation of biological theories with code examples in simple English for researchers and student is still missing. Introduction to bioinformatics with python will be a beginner level book that will lead the readers from beginner to intermediate level. In this book the basic concepts of bioinformatics like genome sequence alignment, similarity matching, genome scoring matrices, evaluation tree will be discussed with practical examples. This book will also leave some practical examples uncomplete to urge the reader to get hands on practice on these concepts. |

**Table of content**

1. Introduction
2. Introduction to Python
3. Fundamentals of Genes and Genomes
4. Data, Databases, Data Format, Database Search, Data Retrieval Systems, and Genome Browsers
5. Sequence Alignment and Similarity Searching in Genomic Databases: BLAST and FASTA
6. Phylogenetic Analysis
7. Additional Bioinformatic Analyses Involving Nucleic-Acid Sequences
8. Additional Bioinformatic Analyses Involving Protein Sequences
9. Computational Methods for Pathways and System biology
10. Introduction to Drug Discovery
11. Emerging Role of Biomarkers in Drug Development

**Introduction**

Introduction chapter will contain the basic terminologies related to bioinformatics and explained fundamental concepts of computational biology. It will also discuss the theoretical concept and give the summary of the content that will explain in later chapters. This chapter will be consisted of 5 pages.

**Introduction to Python**

This chapter will explain the python utilities for beginners. The researchers and students who have no previous exposure to computer programming will able to get basic understandings of code. This chapter will help the reader to configure the python environment on his machine to practically doing the remaining stuff of the book. The python environment configurations, basic command for computational biology will be discussed in this chapter.

**Fundamentals of Genes and Genomes**

In this chapter, we discussed the basic concepts of genome sequences and introduce the readers with basic terms related to genomes and genes. Further, the related topic will be discussed including the, gene, exon, intron, gene mutations, 3 prime and 5 prime UTR. The readers will introduce the practical examples to analyze the genome sequence. The length of this chapter will be approximately 10 pages. The genome sequence part will also be discussed with graphical representation for better understanding.

**Data, Databases, Data Format, Database Search, Data Retrieval Systems, and Genome Browsers**

This chapter will discuss the biological data formats like the genome sequencing data, mutational data, pathway data, NGS data, cancer gene data etc. for computational biology. Further the different sources for the extraction of biological data will be discussed in this Chapter. Later the heterogeneity and complexity of biological data will be explained to get depth knowledge of biological data. The extraction of data and integration of biological data into central source will be discussed with practical examples. This chapter will be base on 30-35 pages.

**Sequence Alignment and Similarity Searching in Genomic Databases: BLAST and FASTA**

This chapter will lead the reader from basic to advance level in sequence alignment and similarity matching. Firstly, the basic idea of sequence alignment will be elaborated with the core python program. Different methods of sequence alignment will be discussed later on with the practical use in python. How the similarity score is calculated for two genome sequences will be explained. The use of scoring matrices for similarity search will be explained with different coding examples. The generic idea of BLAST and FASTA algorithm for similarity search will be discussed for basic understanding of the reader. The use of BLAST and FASTA algorithm with python code will be elaborated with numerous examples. This chapter will be based on the 30-40 pages length.

**Phylogenetic Analysis**

Phylogenetic analysis describes the evolutionary relationship between genome sequences of different species or same species. In this chapter the basic idea of evolutionary discussed on front lines. The extraction of evolutionary relationship between different genome sequence will discussed with practical example. The calculation of phylogenetic tree will be explained by core python programing and built-in libraries. Lastly the use of different tools for the inferencing of phylogenetic tree will be included in this chapter. This chapter will be long on to 30 pages approximately.

**Additional Bioinformatic Analyses Involving Nucleic-Acid Sequences**

In the additional bioinformatics analysis, the annotation of genome will be discussed. The format of annotations file, procedure to annotate genome sequence and the agenda behind the genome annotation will be a part of this chapter. The practical examples of genome annotations will be implemented for the better understanding of researchers and students. The transformation of complex annotation format into user friendly format will be discussed with practical examples. This chapter will be 25 to 30 pages long.

**Additional Bioinformatic Analyses Involving Protein Sequences**

In this chapter, the annotation process will be implemented with protein sequences. The code examples will elaborate the process and practical use of annotation process. This chapter will help the researchers and students to annotate protein sequence for the extraction of protein mutations for newly discovered disease. This chapter will also introduce the readers with the prediction of proteins properties by protein sequence.

**Introduction to Drug Discovery**

This chapter will start with the basic concepts about drugs like drug molecular structure, drugs smiles and drug interactions. Further, this chapter leads the readers to introduce the purpose of drug discovery. The short examples will be implemented for the drug discovery and prediction of drugs for disease symptoms. On the basis of inherited knowledge different tasks will be assigned to reader for practical experience of the researchers. This chapter will be consisted of 30 to 35 pages approximately.

**Emerging Role of Biomarkers in Drug Development**

Biomarkers allow the measurement of drug activity and safety using an end point that is integrated into the therapeutic action of the drug. In this chapter role biomarkers will be elaborated and the examples will be discussed with python code. This chapter will be based on 20 to 25 pages.

**Sample of Proposed Book**

**Instructor’s introduction**

I am Muhammad Usman Ghani Khan. I did BSc Computer Engineering from UET Lahore in 2004. My MS in Computer Science was also from UET Lahore in 2007. My MS Thesis topic was “Design and Implementation of Information Retrieval System for Chest X Ray Images”.

Then I did PhD Computer Science from University of Sheffield, UK in 2012. My Research Topic was Natural Language Description of Video Sequences. Currently, I am working as an Professor CS Dept. UET Lahore.

I have established Bioinformatics Lab at AL-Khwarizmi Institute of Computer Science, in, UET, where I am working as the director of the lab. We are working on different projects such as Brain Informatics, Modeling and Simulation of human body organs such as heart, kidney, lungs, medical imaging, Colorization and Classification of timorous cells, Databank generation of DNA, RNA and PSs.

**Definitions**

A recent Google search for "bioinformatics definition" yielded almost 43,000 entries! As the areas have developed in recent years, there has been more confusion over these two names. For some, the phrases bioinformatics and computational biology are nearly synonymous, but for others, there is a significant difference.

Bioinformatics and Computational biology are multidisciplinary topics that bring together academics from several domains such as statistics, computer science, physics, biochemistry, genetics, and molecular biology. There are 1000s of definitions available for this subject. Here I have just picked some relevant definitions which are directly related with computer science.

The application of information technology to the analysis and management of biological material is known as bioinformatics. Computer technology is more related towards automation of software’s related to biological domains. What stuff can biologists do with computers? Well, there are plenty, calculation of data, finding patterns and information from data, performing measurements and so on.

Design, development, and application of software tools for generating, storing, annotating, accessing, and analyzing data and information in the field of molecular biology. Computational tools and methodologies for boosting the use of biological, medical, behavioral, or health data, including those that obtain, store, organize, archive, analyze, or visualize such data, are being researched, developed, and applied.

**Motivations**

Bioinformatics is a relatively new and diverse field that incorporates biological, information, and computer scientific contributions. The research efforts in the bioinformatics area overlap with the research efforts in computational biology in terms of making it capable of  processing biological  data, data management, and data modelling difficulties. In reality, the realm of bioinformatics is distinct from that of computational biology. Bioinformatics is the study of biological data with the goal of storing, retrieving, and manipulating biological data in a computer laboratory (Computer Lab). Computational biology, on the other hand, deals with sequence alignment, genome assembly, prediction and discovery of new genes/protein structures, as well as the creation of numerous tools and techniques for statistical analysis and biological data curation in the Wet Lab with some extent in dry Lab. The difference between wet labs, dry labs and computer labs is shown as following.

Biological Data Acquisition, Processing and Analysis

Rectangles signify processes, while arrows denote the processes' input and output. The biological data collecting, processing, and analysis functions are depicted in the diagram. Wet Laboratory (Wet Lab), Dry Laboratory (Dry Lab), and Computer Laboratory are the three working phases/stages. A wet lab is a lab where biologists conduct physical tests with various samples, chemicals, medications, and other things. Biologists in this lab acquire information by assisting with tests on living organism samples or by personally studying living organisms under specified precautions. Raw data refers to the information acquired during an experiment. Investigations in the wet lab are carried out in facilities that are physically enclosed and designed expressly for this purpose. Biologists move raw data (also known as uncured data) from their wet lab investigations to the dry lab. Because the uncured data is still in biological form, it is treated further for future study. Data curation is the process of cleaning uncured data, which is carried out in a dry lab by biologists and robots. Semi-experiments are used in dry labs to carry out experimental tasks. Because of the automatic instrumentation, there are few risks of error. Data is output from Dry Lab; it is cleansed and can be used by computer scientists in Computer Lab for analysis and other computational tasks.

This sort of experimental data is quite expensive and is acquired from raw data via an annotation procedure, also known as annotated data. The annotated material is utilized to create a variety of structure and function annotations, as well as sequence determination and other structural predictions.

**What we will study**

In the basic course of bioinformatics, you have already learnt many things such as Molecular biology, Databases Introduction, Computing concepts related to bioinformatics, Pairwise Sequence Alignment, Tools for similarity search and sequence alignment, Multiple Sequence Alignment and Protein Structure Modeling and Simulation.

In this course we will look at advanced areas of the bioinformatics fields such as molecular evolution, i.e., Phylogeny and Phylogenic Algorithms, Hidden Markov Models and their usability in bioinformatics. Drug Discovery: Technology and Strategies, classification and understanding of biological documents, medical imaging primitives, programming languages for bioinformatics such java, mat lab and python.

**Introduction to Biological Concepts**

Organism is the single source of biological data, from which our lives have been started. At abstract level, Organisms is classified into two broad categories 1- Plants and 2- is Animals. Animals are further sub-divided into Prokaryotic organisms as well as eukaryotic organisms. Most common sources of biological data are different types of plants, sea animals, flying birds, creeping animals etc. are some of them.

The bodies of living organisms based on DNA, RNA Protein structures which make the fundamental pillars of the bodies of all of the organisms.

Wet Labs are the physically housed laboratories, where biologists accomplish their physical researches by using different substances, drugs and other materials. In this lab, biologists collect data and information of their experiments through samples of living organisms or on living organisms directly or indirectly like blood, urine, Siemens and deoxyribonucleic acid (DNA) under certain circumstances or conditions. The data that is gathered by using physical experiments may consist of much anomalies and data replication among them, that is why it is referred to as the raw data.

Biologist brings raw data of their experiments to the dry labs. This uncured data is still in biological formats, and it needs curation for its further analysis and usage. It is initially processed in dry labs using automated sequencing instruments, robots, and the assistance of biologists. Semi-experiments are used in dry labs to carry out experimental tasks. Because of the automatic instrumentation, there are few risks of error. Data is output from Dry Lab; it is cleansed and can be used by computer bioinformaticians in Computation Lab for analysis and other computing tasks. This form of scientific results is quite expensive and is generated through an annotation procedure from raw data. The annotated data should be utilized for the computation of various types of structure and function annotations, sequence determining and other structure predictions, data modelling, simulation, and analysis.

The biological data available in computer lab may contain diversities of data formats. It may consist chains of sequences of nucleotides, DNA, RNA, Protein structures, genes, neurons, molecular structures, chemical structures, chemical reactions, graphs, hypothesis, scalars, audios, videos, texts, numeric, alpha-numeric and many others data types.

These contents of biological data are stored in diversities of data banks or local repositories like DDBJ, PDB, Knowledgebase, CCSB, Bio-UML, Open Tree of life, prototype selection generation etc.

Furthermore, these databanks or repositories are lying at diverse locations, their contents are lying in fragmented or isolation form due to not having specific data formats, standards. Local research groups have made their own standards for future usage.

**DNA and parts**

Life on this planet, i.e., Earth, is thought to have evolved from a common root for all creatures. A singleton (single) cell organism or unicellular organism is thought to have been the ancestor. Due to differences in the genetic traits of organisms a variety of species has been formed. In today's earth, there are more than 100 million living species (Earth). Cells are the fundamental unit or tenet of all living creatures, whether they comprise micro- or macromolecules. A cell shares and carries all of the characteristics of life that are required for a living entity to function. Prokaryotes, which include bacteria, archaea, and protists, are organisms that are made up of only one cell. The majority of entities, such as fungus, humans, and animals, are made up of multicellular organisms known as eukaryotes. All of these living things rely on cells for their survival. Our bodies are made up of more than 1013 cells, each of which is unique in terms of shape, function, and size. Intracellular components or micro molecules are terms used to describe the components of a cell. In all kinds of living organisms, some of the recognized activities carry out division of cells, mitosis/meiosis. These various living cells function as a computer interface, carrying genetic information in the form of chromosomes and genes among their materials. Eukaryotic cells and prokaryotic cells are the two types of cells that exist. The nucleus of eukaryotic cells is where deoxyribonucleic acid (DNA) is stored. The nucleus is separated from other antibodies in eukaryotic cells by a nuclear membrane. Be a result, the nucleus is referred to as the eukaryotic cell's main part. A prokaryotic cell is one that lacks a nucleus and instead contains deoxyribonucleic acid in its cytoplasm. Deoxyribonucleic acid (DNA), ribonucleic acid (RNA), and a variety of protein structures are self-contained in the nucleus, its portions, and all components of a cell.

Deoxyribonucleic Acid (DNA)

Deoxyribonucleic acid (DNA) functions as a computer tape or storage facility that can store data for a longer period of time. It (DNA) keeps track of, manages, and oversees all cell activity within molecular creatures' bodies. When we look at the structure of a single DNA molecule, we see that it has the shape of a double-stranded helical structure. These helical strands are thought to be travelling in the reverse direction from one another, resulting in an anti-parallel structure. Each strand of DNA is made up of a long and complex succession of polynucleotides. The nucleotides that are bonded closer together are anti-parallel and create a lengthy DNA molecule chain. If one strand of nucleotides reflects a direction of 5' to 3', the other will be 3' to 5' in the direction of same strand. These strands twist around one another to form a helical configuration that appears to be derived from any single strand of DNA molecule. Every nucleotide strand is classified as an organic compound, with three basic groups: deoxyribose (sugar), phosphate, and nitrogen bases. All of these nucleotides with various nitrogen bases have sugar and phosphate groups in common. Sugar is made up of different atoms and resembles a pentose-ring structure. Sugar links oxygen at position 1I and carbon atoms at places 2 to 5 in a single molecule in a clockwise orientation. A variety of distinct phosphate group molecules are bonded to carbon atoms at various locations. Simple bases are a term used to describe nitrogen bases. Every nucleotide has four different forms of nitrogen bases. Adenine (A), cytosine (C), guanine (G), and thymine (T) are the four bases, each of which is represented by a single letter: A, C, G, and T respectively. The nucleotide strands are joined together by hydrogen bonds. The bases adenine (A) and thymine (T) complement each other, making a double covalent bond. On the opposing strands, cytosine (C) and guanine (G) create a triple covalent bond. Purine is the base pairing of A+T, while pyrimidines is the base pairing of C+G. Pyrimidines are more persistent than purines because of their triple hydrogen bonding.

Coding regions and non-coding regions molecules make up the DNA molecule. The coding regions compose up 3% to 5% of the total DNA. In wet labs, these zones contain a significant set of genes and templates that form protein structures as an output and play a key role in several biological processes such as replication, transcription, and translation. The non-coding region makes up 95 to 97 percent of the DNA molecule. These non-coding areas are not involved in the replication, transcription, or translation processes in any way. These sections, on the other hand, convey various types of information that serve as pauses in the replication, transcription, and translation processes. Chromatin protein and chromosomes make up coding areas. There are 23 (twenty-three) chromosomal pairs in the human body. There are 22 pairs of autosomes among them. In males, the 23rd pair of chromosomes, known as sex chromosomes, comprises of the X and Y chromosomes. Both X chromosomes are present in females. Hundreds to thousands of genes are found on a single chromosome.

Replication and transcription are two processes that DNA accomplishes. Through the meiosis/mitosis process, parental DNA separates into two double helix daughter strands, maintaining the data in daughter molecules. Every daughter double strand molecule is further split into 2 child double strand molecules during replication, which is a cyclic and dynamic process. Just the sequences of nitrogenous nucleotides are duplicated during Replication of DNA. DNA is transformed into RNA polymerase, messenger ribonucleic acid (mRNA), transfer ribonucleic acid (tRNA), and ribosomal ribonucleic acid (rRNA) during the transcription process. Although RNA polymerase cannot start the transcriptional activation on its own, the transcription factor in the promoter coding region prompts RNA polymerase to do so. Each mRNA molecule is transformed into a protein structure during translation. Protein structures are encoded using a linear amino acid sequence. The sequence of amino acids is represented by the mixture of three nucleotides. The codon is a triplet of nucleotides.

Ribonucleic Acid (RNA)

Ribonucleic acid (RNA) is a type of organic element that is similar to DNA but differs in several internal aspects such as shape and size. RNA is a polymer of ribonucleotides with a single strand that wraps back on itself to produce a double helix shape. The majority of RNAs (tRNA, rRNA, and mRNA) are located in the cytoplasm, with only a few mRNAs detected in the nucleus. Protein synthesis has been aided by rRNA. RNA is made up of a lengthy chain of ribo nucleotides, each of which is made up of ribose sugar, phosphate, and nitrogen bases, just like DNA. In RNA, the phosphate ion is comparable to the phosphate ion in DNA. The ribose molecule of RNA differs from the deoxyribose molecule of DNA. In RNA, ribose has an extra oxygen atom at the 2' position. The RNA molecule, like the DNA molecule, is made up of four different types of nitrogen bases. Adenine (A), cytosine (C), guanine (G), and uracil (U) are the four types of bases. Thymine (T) replaces the nitrogen base uracil in RNA. In the RNA molecule, there is also a base pairing between complement nucleotides. Between A and U, a double covalent link is created, while between C and G, a triple covalent bond is produced. The RNA molecule is less stable than the DNA molecule, and it serves as a bridge between the two for the transfer of genetic information from DNA to protein structures. As a result, understanding its inherent features is critical for transformation.

There are three kinds of RNAs: mRNA, tRNA, and rRNA. During the process of transcription in organisms' bodies, they perform several roles. Initiation, elongation, and termination are three minor steps that surround the transcription process (from DNA to RNA). The messenger RNA (mRNA) transports messages from the nucleus to the locations where protein is made. It is made up of a single strand that is complementary to the single strand of DNA. Three sequentially joined mRNAs form a codon, which further defines the amino acid sequences. The tRNA molecule transports amino acids to the ribosome. It also binds a single amino acid to a single molecule of mRNA. It joins 3 amino acids in a precise sequence to generate the anti-codon. The rRNA portion of RNA that stays operational during the synthesis of protein structures is known as the acting component of RNA.

Protein Structures

Protein is derived from the Greek word protoas, which meaning "first and foremost." Protein molecules are large chemical complexes made up of many amino acids linked together. Proteins play a vital function in our daily lives. Proteins play a variety of roles in the human body. Some proteins work as enzymes, which act as catalysts in our bodies, triggering metabolic functions. Some proteins are rigid, forming tendons, muscles, nails, and bones, while others are fibrous, forming connective tissues in our body. In summary, proteins are required for digestion of food in the small intestine, oxygen transport in the blood, and the formation of epithelial cells in our skin. On the basis of poly peptide chain structure, there are four different types of protein structures. Primary, secondary, tertiary, and quaternary structures are among them. Because of the varied combinations of amino acid sequences, these protein structures have different forms. Linear, coil, helix, parallel sheet, folds, loops, and helices are some of the stuffs they make.

Protein structures are formed from amino acid sequences. Genes have coded for these amino acids. Protein structure's primary purpose is synthesis, which is accomplished by forming a peptide bond between two amino acids. This reaction is performed several times, resulting in a lengthy polypeptide chain. In a protein structure, there are typically twenty (20) different types of amino acids, and numerous combinations are conceivable in a codon. Between one to six codons code are used for each amino acid. The ribosome falls off when it hits one of the three condones for which there is no tRNA translation, and the produced protein is released.

**Structure of DNA**

Protein and Gene

Polypeptides with a three-dimensional architecture are known as proteins. There are four different layers that can be used to characterise them:

• Primary structure — the amino acid sequence that makes up the polypeptides.

• Secondary structure — the grouping of peptide chains segments into secondary structures like α - helices and β - sheets on a local level.

• Tertiary structure — the three-dimensional configurations of amino acids when they interact to one another due to polarity and the interactions that ensue among their side chains.

• Quaternary structure - if a protein is made up of many protein subunits bound collectively, the number and location of the subunits can be used to define the protein.

|  |  |
| --- | --- |
| **Visualization of Protein Structures.** | |
| http://www.ebi.ac.uk/microarray/biology_intro_files/rasmol.gif | http://www.ebi.ac.uk/microarray/biology_intro_files/triphos.jpe |
| Magenta: alpha helix  Gold: Beta Sheets | Blue: Monomer A  Orange: Monomer B |

Calculating a protein's secondary structure from its primary structure is a difficult undertaking. The topic of protein folding prediction will be covered near the end of the book.

Monomer - Any tiny molecule that may be joined with others of the same sort to form a polymers is referred to as a monomer. Nucleic acids, amino acids, and proteins are examples of substances that could be used in this class.

Dimer - A dimer is made up of two tiny molecules of the same sort that are bonded together.

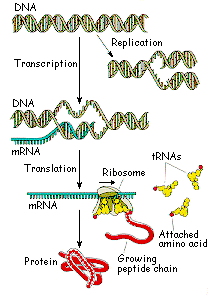
Trimer - A trimer is made up of three tiny molecules of the same sort that are bonded together.

Oligomer - A short polymer made up mostly of nucleic acids or amino acids that is referred to as an oligomer.

Polymer - Any big molecule made up of numerous identical or similar subunits united by covalent connections is known as a polymer.

When we put these together, we should get stream of genetic data. To put it another way, DNA regulates the production of RNA, which then leads the production of protein. The Central Dogma of Molecular Biology refers to the passage of genetic information from nucleic acids to proteins.

Central Dogma of Molecular Biology



What is a Gene?

  In a nutshell, a gene is the functional and physical unit of heredity that transmit data from generation to generation. A gene is a part of DNA sequence that is required for the creation of a functioning protein or RNA molecule.

Genome, Transcriptome, Proteome

When the phrase genome is employed, it usually refers to an organism's chromosomal DNA, or, in the case of sequencing, the heterochromatic sections of the chromosomal DNA. The set of chromosomes and the length of the gene differ greatly between organisms. The table below shows an example of genomic sizes. Don't be deceived by this table into thinking that an organism's complexity is determined by its genome size and the number of genes. Many plant genomes are in fact substantially larger than the human genome!

|  |  |  |  |
| --- | --- | --- | --- |
| **ORGANISM** | **CHROMOSOMES** | **GENOME SIZE** | **GENES** |
| [*Homo sapiens*](http://www.ncbi.nlm.nih.gov/genome/guide/human/) (Humans) | 23 | 3,200,000,000 | ~ 30,000 |
| [*Mus musculus*](http://www.ncbi.nlm.nih.gov/genome/guide/mouse/)  (Mouse) | 20 | 2,600,000,000 | ~30,000 |
| [*Drosophila melanogaster*](http://www.ncbi.nlm.nih.gov/mapview/map_search.cgi?taxid=7227)(Fruit Fly) | 4 | 180,000,000 | ~18,000 |
| *Saccharomyces cerevisiae* (Yeast) | 16 | 14,000,000 | ~6,000 |
| *Zea mays (Corn)* | 10 | 2,400,000,000 | ??? |

The transcriptome denotes to an organism's entire collection of mRNAs (including isoform). The portions of an organism's genome that are transcribed into messenger RNA are known as transcribed segments. The transcriptome can be expanded in some circumstances to encompass all transcribed components, including non-coding RNAs that serve structural and regulatory functions.

The proteome is a term that refers to an organism's entire collection of proteins. The proteome can be analyzed as a static (all proteins found at a certain time point) or dynamic (all proteins discovered at a specific point in time) entity.

**Data Visualization**

Many professions consider data visualization to be the modern counterpart of visual communication. It isn't held by any one field, but rather finds use in a variety of them.  It entails the production and analysis of data visualizations, which are defined as "knowledge that has been represented in some graphical form, including qualities or variables for the units of data."

Data visualization's main purpose is to transmit information to users in a positive and effective manner using the statistical visuals, graphs, information plots, tables, and charts that are chosen. Users can utilize effective visualization to help them analyze and reason about facts and evidence. It improves the accessibility, comprehension, and usability of complex data. Users may be given specific analyses, such as drawing comparisons or comprehending causality, and the graphic's design concept (showing parallels or displaying causation) is dictated by the task. Tables are commonly used to search up a single measure of a variable, whereas other forms of charts are used to highlight trends and relationships in data for one or more factors.

Data visualization is both a science and an art. As a result of an increasingly information-based economy, the pace at which data is being generated has increased. "Big Data" refers to data generated by internet activity and an increasing number of sensor nodes in the environment, such as satellites and traffic cameras. Data visualization faces a number of moral and analytical issues when it comes to processing, interpreting, and communicating this data. To solve this problem, the discipline of data science and professionals known as data scientists have evolved.

3D cubes, dispersion charts, slopes, surfaces, link networks, images and movies, and parallel coordinates are all common data visualization inputs. Pie charts, scatter diagram, box plots, association rules, dendrograms, temporal evolution, and other output results are possible. What is the purpose of visualization? Why do we require it in the first place? "A picture is worth a million words," says a popular adage. We now have access to enormous volumes of data, yet output devices have restricted display capability. For data visualization, the strength of automatic calculations and the skills of human processing should be copied. The process of extracting structures from images is phenomenal in human perception.

**Data Visualization Methods**

Data can be

* Univariate
* Bivariate
* Multivariate

Univariate Data: Measurement of single quantitative variable. It can be represented using

* Histogram
* Pie Chart

Bivariate Data: Constitutes of paired samples of two quantitative variables. Variables are related to each other in one or the other way. It is represented using following methods

* Scatter plots
* Line graphs

Multivariate Data: Multi-dimensional representation of multivariate data. It is represented using

* Icon based methods
* Pixel based methods
* Dynamic parallel coordinate system

Some Visualization Tools

* Cn3D - uses MMDB-Entrez's structure database: http://www.ncbi.nlm.nih.gov/Structure/CN3D/cn3d.shtml
* RasMol: http://www.umass.edu/microbio/rasmol/
* Protein Explorer: http://www.umass.edu/microbio/rasmol/rotating.htm
* World Index of Molecular Visualization: <http://molvis.sdsc.edu/visres/index.html>

**Introduction- Sequence Alignment**

Sequence Alignment is a method of arranging DNA, RNA, or protein sequences in order to find regions of homology that may be the result of functionality, morphological, or evolutionary links between them.

Comparing two or more sequences by looking for a set of individual letters or patterns in the same order throughout the sequences.

Why Alignment?

1. To discover structural, functional and evolutionary information.
2. If two sequences are similar, they might share the same ancestor (Homology).
3. If two sequences are similar, they may share the same structure, therefore similar function

Homologs - Homologs are sequences that are similar in two separate organisms and were formed from a shared ancestor genome.

Orthologs - Orthologs are sequences that are similar in two separate organisms that have developed as a result of a speciation event and have maintained their functionality throughout evolution.

Identical regions within a single organism that have evolved as a result of a homologous recombination event are referred to as paralogs.

Xenologs: Sequences which are identical but do not have the same evolutionary basis. They have developed as a result of horizontal transfer events such as symbiosis, viruses, and so on.

**Distance Measures**

In this module, detail about distance measured is mentioned that how can we compare two sequences or two strings? Well there are several methods; in this module we will cover the very basic ones.

Hamming or edit distance: calculating the modify distance of two sequences is one way for detecting homologies. What is the similarity between the phrases pear and tear, for example? We see that if we alter the p to a t while keeping the ear, pear becomes tear. As a result, the first letter has a mismatch, whereas the last three have matches. The following is a diagram illustrating how these two are related:

B E A R

| | |

N E A R

Calculating the Hamming distance, which is the difference in letters between the two words, is one technique to grade this alignment. The Hamming distance in this case would be 1. When phrase is aligned to one another, the Hamming distance is found by summing up the number of mismatches.

When working with biological sequences, it's common to have to align two sequences that are different lengths or have had parts inserted or removed over time. As a result, the concept of gaps must be introduced. Take a look at the word’s "alignment" and "ligament." The following is one possible alignment of these phrase:

A L I G N M E N T

| | | | | | |

- L I G A M E N T

A '-'character is used to indicate a gap in the alignment in this scenario. A match between main individuals, a mismatch between two characters (also known as a substitution or mutation), a gap in the first series (which can be thought of as the removal of a letter in the first string), or a gap in the second sequence can all result from an alignment (which can be thought of as the insertion of a character in the first sequence).

Take the nucleotide sequences ACGGACT and ATCGGATCT for example. Two valid alignments are as follows:

A – C – G G – A C T

| | | | |

A T C G G A T \_ C T

A T C G G A T C T

| | | | | |

A – C G G – A C T

Longest Common Subsequence

x=ABCBDAB and y=BDCABA, BCA is a common subsequence and BCBA and BDAB are two LCSs

N- grams Methods: Subdivide words into N- grams -set of overlapping substrings of length N

N=2:

(radio) (ra - ad - di -io)

N=3:

(radio) (rad - adi - dio)

Alignment scoring schemes

Which of the two alignments is the better one? A positive number for each match, a negative mark for each mismatch, and a negative score for each insertion/deletion is one way to judge this (collectively referred to as indels).

The following values might be assigned by one scoring scheme:

match: +3

mismatch: -1

indel: -2

Using this scoring scheme, the first alignment has 5 matches, 1 mismatch, and 4 indels. The score for this alignment is: 5 \* 2 – 1(1) – 4(2) = 10 – 1 – 8 = 1.

The second alignment has 6 matches, 1 mismatch, and 2 indels. The score for the second alignment is 6 \* 2 – 1(1) – 2 (2) = 12 – 1 – 4 = 7.

Therefore, using the above scoring scheme, the second alignment is a better alignment, since it produces a higher alignment score.

**Types of Sequence Alignment**

There are two types of Sequence Alignment methods.

* Global Alignment
* Local Alignment

Global Alignment: In global alignment, an effort is made to align the whole sequence. Find finest match of both sequences in their entirety.

Input: Assume the two sequences as potentially equal.

Goal: recognize conserved regions and differences.

Useful when sequences are similar or roughly same length

Applications: Comparing two genes with same function (in human vs. mouse). Comparison of two proteins with similar functionality.

Local Alignment: Find the best subsequence match from two sequences.

Input: The two sequences may/may not be related.

Goal: see whether a substring in one sequence aligns well with a substring in the other.

Applications: Dissimilar sequences which are suspected to have similar portion. Useful for comparison of DNA sequences that share a same motif.

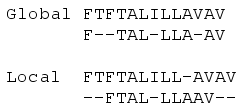
Suitable for comparing protein sequences that share a similar motif.

Why Local Alignment?

More meaningful: Finds conserved areas between two genome sequences.  Coincides two partially overlapping sequences and two sequences of varying length to be aligned. Two sequences are aligned, one of which is a subsequence of the other.

When the genomes in the request set are identical and of nearly equal length, global alignments, which seek to align every residue in every genome, are most useful. (This isn't to say that global alignments can't have gaps.) The Needleman–Wunsch algorithm, which is built on dynamic programming, is a worldwide alignment approach. Local alignments are more beneficial for heterogeneous genomes suspected of containing similar sequence motifs or patches of similarity within their wider context. The Smith–Waterman method is a dynamic programming-based general local alignment method.

Semi-global or "local" techniques are hybrid approaches that try to identify the best feasible alignment which includes the beginning and the end of one or both sequences. When the upstream section of one sequence overlaps with the downstream section of the other sequence, this can be extremely valuable. In this scenario, neither global neither local alignment is totally appropriate: a global alignment will try to push the alignment to extend beyond the overlap region, whereas a local alignment might not completely cover the overlapping area. [6] When one genome is small (for example, a gene sequence) the other is extremely lengthy, semi-global alignment is useful. In that situation, the short genome should be globally aligned but only a local alignment is anticipated for the long sequence.



**Dot Plots**

The use of a graphical alignment known as dot plots are among the simplest, yet crucial, approaches for detecting the alignment between two sequences. A matrix is used to construct dot plots of resemblance, with the rows of the matrix corresponding to the letters in the first genome and the columns corresponding to the letters in the second genome. The dot plot is made in the following way: Each row is looped. Select the character from the current row and match it to the letter from each column. Place a marker in the matrix if they are equal. Proceed until you've considered all of the entries in the matrix.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **C** | **C** | **T** | **G** | **A** | **G** | **C** | **T** | **C** | **A** | **C** | **C** | **T** | **G** | **A** | **G** | **T** | **T** | **A** |
| **A** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **T** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **G** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **A** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **G** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **T** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **A** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **T** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **G** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **A** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **G** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **T** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **T** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **A** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Global Alignment**

Fortunately, because sequence alignment has an optimal-substructure feature, that's a much simpler technique to evaluate all of the alternative alignments using optimization methods. Dynamic programming techniques are employed in a wide range of computer science applications. Dynamic programming algorithms tackle optimization problems by breaking them down into smaller, more manageable chunks. Each subproblem is thus only saved once, and the answer is saved in a table, eliminating the need to recalculate the solution.

The subtasks in sequence alignment can be thought of as the alignment of the two sequences' "prefixes" to a specific point. As a result, a dynamic programming matrix must be calculated. The ideal alignment score for any given point in the matrix is based on the optimal alignment score that has been calculated to that point.

Beginning at the endpoints of the two sequences, dynamic programming approaches strive to align all feasible pairs of letters (one from each genome sequence) using a grading system for matching, mismatch, and gaps. The best alignment between the two genome sequences is determined by the highest set of scores.

We'll start with dynamic programming in concepts of DNA, in which only exact matching are taken into account while calculating a match score. We'll go through how substitution matrices can be used to evaluate amino acid matches and mismatches in a later section.

Given a specific performance measurement, dynamic programming solutions are guaranteed to yield the best alignment. Dynamic programming can be costly and memory demanding when dealing with long sequences. Examine the amount of time and space required for microarray analysis.

Setting up the Dynamic Programming Matrix

We're now ready to begin constructing the dynamic programming matrix. One of the sequences must be aligned across the matrix's columns, while the other must be aligned across the rows. It's worth noting that an alignment can start with a gap in one of the genome sequences, so that must be addressed as well. Assume we're trying to match the genome sequence region GAATTCAGTTA to GGATCGA. The first sequence is 11 nucleotides long, while the second is 7 nucleotides long. Due to the possibility of starting an alignment with a gap, the matrix should be 8 x 12. Gaps will be represented by row 0 and column 0. Rows 1 to row 7 will be labeled with the corresponding nucleotide of the sequence GGATCGA, while columns 1 to 11 will be labeled with the corresponding nucleotide of the genome region GAATTCAGTTA. The initial matrix, S, is as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **-** | **G** | **A** | **A** | **T** | **T** | **C** | **A** | **G** | **T** | **T** | **A** |
| **-** |  |  |  |  |  |  |  |  |  |  |  |  |
| **G** |  |  |  |  |  |  |  |  |  |  |  |  |
| **G** |  |  |  |  |  |  |  |  |  |  |  |  |
| **A** |  |  |  |  |  |  |  |  |  |  |  |  |
| **T** |  |  |  |  |  |  |  |  |  |  |  |  |
| **C** |  |  |  |  |  |  |  |  |  |  |  |  |
| **G** |  |  |  |  |  |  |  |  |  |  |  |  |
| **A** |  |  |  |  |  |  |  |  |  |  |  |  |

Now we have to decide on a score system to use. A match ranking score, a mismatch ranking score, and a gap score are all required criteria. The match and mismatch scores will be added together to create a single match/mismatch score (ai bj). We'll see just how we can use this with a substitution matrix soon. A single linear gap penalty score, w, will also be used. The following are the parameters for our first instance:

Sequence #1: GAATTCAGTTA; M = 11

Sequence #2: GGATCGA; N = 7

* s(aibj) = +5 if ai = bj (match score)
* s(aibj) = -3 if aibj (mismatch score)
* w = -4 (gap penalty)

Three steps in dynamic programming

There are three main steps to calculate the ideal scoring alignment once you've specified the scoring functions and sequences to align. Whether regional or global sequence alignment is needed determines the strategies utilized to complete these three processes. The following are the three steps:

* Initialization
* Matrix Fill (scoring)
* Traceback (alignment)

Global Alignment: Needleman-Wunsch Algorithm

In global sequence alignment, the full length of two distinct genomes is aligned, up to and including the sequence endpoints. Needleman and Wunsch (1970) were among the first to develop a global sequence alignment method based on dynamic programming.

**Local Alignment**

Local Alignment: Smith-Waterman Algorithm

Temple Smith and Mike Waterman presented a Needleman-Wunsch approach modification in 1981 to get a local sequence alignment that resulted in the greatest local match between two genomes.

Why choose a local alignment algorithm?

* More meaningful – point out conserved regions between two sequences
* Aligns two sequences of different lengths to be matched
* Aligns two partially overlapping sequences
* Aligns two sequences where one is a subsequence of another

To turn the Needleman-Wunsch Algorithm in a local alignment technique, only two small changes are required. Negative scores are required for mismatches in the initial modification. The second change is that once the dynamic programming score matrix value turns negative, it is set to zero, thus halting any alignment that has occurred up to that point. As a result, the matrix score has changed to:

Si,j = MAXIMUM[

Si-1, j-1 + s(ai,bj) (match/mismatch in the diagonal),

Si,j-1 + w (gap in sequence #1),

Si-1,j + w (gap in sequence #2),

0]

Starting from the top scoring points in the scoring matrix, the local alignments are created by following a trace route from those positions up to a block that values zero.

Initialization is the first step. Each row Si,0 is set to 0 during the local alignment initialization step. Furthermore, each column S0, j is set to zero. The initialization stage yields the following results using the scoring scheme stated above:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **-** | **G** | **A** | **A** | **T** | **T** | **C** | **A** | **G** | **T** | **T** | **A** |
| **-** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **G** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **G** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **A** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **T** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **C** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **G** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **A** | **0** |  |  |  |  |  |  |  |  |  |  |  |

Matrix Fill is a step in the process of filling a matrix. By starting in the top left side of the matrix and getting the highest score Si,j for each point in the matrix, one potential solution of the matrix fill step yields the largest local alignment score. It is essential to know the score for the matrix locations to the left, above, and diagonal to I j in order to determine Si,j for any I j. It is important to know Si-1, j, Si, j-1, and Si-1, j-1 in terms of matrix locations.

For each position, Si,j is defined to be the maximum score at position i, j; i.e.

Si,j = MAXIMUM[

Si-1, j-1 + s(ai,bj) (match/mismatch in the diagonal),

Si,j-1 + w (gap in sequence #1),

Si-1,j + w (gap in sequence #2),

0]

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **-** | **G** | **A** | **A** | **T** | **T** | **C** | **A** | **G** | **T** | **T** | **A** |
| **-** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **G** | **0** | **5** |  |  |  |  |  |  |  |  |  |  |
| **G** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **A** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **T** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **C** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **G** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **A** | **0** |  |  |  |  |  |  |  |  |  |  |  |

Note that in the sample, Si-1,j-1 is red, Si,j-1 is green and Si-1,j is blue. Using this evidence, the score at position 1x1 in the matrix can be calculated. Since the first nucleotide in both genome sequences is a G, s(a1b1) = 5, and by the traditions stated earlier, w = -4. Thus, S1x1 = MAX[S0,0 + 5, S1,0 - 4, S0,1 – 4,0] = MAX[5, -4, -4, 0].

Now we continue to S1,2. Since a1 = G and b2 = A, there is a mismatch. Therefore, s(a1, b2) = -3 and by the conventions stated earlier, w = -4. Thus, S1,2 = MAX[S0,1 -3, S1,1 - 4, S0,2 – 4, 0] = MAX[0 - 3, 5 – 4, 0 – 4, 0] = MAX[-3, 1, -4, 0] = 1. An arrow is placed back into the cell that resulted in the maximum score, which is the cell S1,1.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **-** | **G** | **A** | **A** | **T** | **T** | **C** | **A** | **G** | **T** | **T** | **A** |
| **-** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **G** | **0** | **5** | 1 |  |  |  |  |  |  |  |  |  |
| **G** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **A** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **T** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **C** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **G** | **0** |  |  |  |  |  |  |  |  |  |  |  |
| **A** | **0** |  |  |  |  |  |  |  |  |  |  |  |