



NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE
(NAAC Accredited)
(Approved by AICTE, Affiliated to APJ Abdul Kalam Technological University, Kerala)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE MATERIALS



CS 465 BIO INFORMATICS

VISION OF THE INSTITUTION

To mould true citizens who are millennium leaders and catalysts of change through excellence in education.

MISSION OF THE INSTITUTION

NCERC is committed to transform itself into a center of excellence in Learning and Research in Engineering and Frontier Technology and to impart quality education to mould technically competent citizens with moral integrity, social commitment and ethical values.

We intend to facilitate our students to assimilate the latest technological know-how and to imbibe discipline, culture and spiritually, and to mould them in to technological giants, dedicated research scientists and intellectual leaders of the country who can spread the beams of light and happiness among the poor and the underprivileged.

ABOUT DEPARTMENT

- ◆ Established in: 2002
- ◆ Course offered : B.Tech in Computer Science and Engineering
M.Tech in Computer Science and Engineering
M.Tech in Cyber Security
- ◆ Approved by AICTE New Delhi and Accredited by NAAC
- ◆ Affiliated to A P J Abdul Kalam Technological University.

DEPARTMENT VISION

Producing Highly Competent, Innovative and Ethical Computer Science and Engineering Professionals to facilitate continuous technological advancement.

DEPARTMENT MISSION

1. To Impart Quality Education by creative Teaching Learning Process
2. To Promote cutting-edge Research and Development Process to solve real world problems with emerging technologies.
3. To Inculcate Entrepreneurship Skills among Students.
4. To cultivate Moral and Ethical Values in their Profession.
- 5.

PROGRAMME EDUCATIONAL OBJECTIVES

- PEO1:** Graduates will be able to Work and Contribute in the domains of Computer Science and Engineering through lifelong learning.
- PEO2:** Graduates will be able to Analyse, design and development of novel Software Packages, Web Services, System Tools and Components as per needs and specifications.
- PEO3:** Graduates will be able to demonstrate their ability to adapt to a rapidly changing environment by learning and applying new technologies.
- PEO4:** Graduates will be able to adopt ethical attitudes, exhibit effective communication skills, Teamwork and leadership qualities.

PROGRAM OUTCOMES (POS)

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems :** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage :** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work :** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO)

PSO1: Ability to Formulate and Simulate Innovative Ideas to provide software solutions for Real-time Problems and to investigate for its future scope.

PSO2: Ability to learn and apply various methodologies for facilitating development of high quality System Software Tools and Efficient Web Design Models with a focus on performance

optimization.

PSO3: Ability to inculcate the Knowledge for developing Codes and integrating hardware/software products in the domains of Big Data Analytics, Web Applications and Mobile Apps to create innovative career path and for the socially relevant issues.

COURSE OUTCOMES

CO1	Demonstrate the knowledge of fundamental concepts in graph theory, including properties and characterization of graphs.
CO2	Demonstrate the fundamental theorems on Eulerian and Hamiltonian graphs.
CO3	Demonstrate the properties and characterization of trees. Illustrate the working of Prim's and Kruskal's algorithms for finding minimum cost spanning tree.
CO4	Explain planar graphs, their properties and an application for planar graphs.
CO5	Illustrate how one can represent a graph in a computer.
CO6	Develop the efficient algorithms for graph related problems in different domains of engineering and science.

MAPPING OF COURSE OUTCOMES WITH PROGRAM OUTCOMES

CO'S	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
C407.1	3	3										
C407.2		3	3	3								
C407.3		3	2	3								
C407.4	3		2									
C407.5	2	3	3	3								
C407.6	2	3	3	3	3							
C407	2.5	3	2.6	3	3							

Note: H-Highly correlated=3, M-Medium correlated=2, L-Less correlated=1

CO'S	PSO1	PSO2	PSO3
C407.1			
C407.2	2		
C407.3	3	2	
C407.4	2	3	
C407.5	3	3	
C407.6	2	3	3
C407	2.4	2.75	3

SYLLABUS



Course code	Course Name	L-T-P Credits	Year of Introduction
CS465	BIOINFORMATICS	3-0-0-3	2016

Course Objectives:

- To introduce concepts and data representations in bioinformatics
- To introduce fundamentals of Sequence alignment and Gene Recognition
- To discuss predictive methods using DNA and Protein Sequences

Syllabus:

Introduction to bioinformatics and molecular biology: Databases tools and their uses, Data searches and Pairwise Alignments, Multiple Sequence Alignments, Molecular Phylogenetic, Genomics and Gene Recognition, Protein and RNA structure Prediction

Expected Outcome:

The Students will be able to :

- interpret the concepts of bioinformatics
- identify different types of biological sequence
- analyse multiple sequences and find conserved regions
- predict RNA and Protein secondary structures
- analyse genomic sequences and identify encoded gene regions

References:

1. S C Rastogi, N Mendiratta and P Rastogi, " Bioinformatics: Methods and Applications", ISBN : 978-81-203-4785-4, published by PHI Learning Private Limited, New Delhi, 2015.
2. D E Krane and M L Raymer, Fundamental Concepts of Bioinformatics, ISBN 978-81-7758-757-9, Pearson Education, 2006.
3. Andreas D.Baxevanis, B F Francis Ouellette, "Bioinformatics - A Practical Guide to the Analysis of Genes and Proteins", Third Edition, 2005-2006, ISBN: 978-81-265-2192-0, published by John Wiley & Sons INC. , U.K.
4. Neil C Jones and Pavel A Pevzner, An Introduction to Bioinformatics Algorithms, MIT press, 2004.

Course Plan

Module	Contents	Hours	End Sem. Exam Marks
I	Bioinformatics and Computational Biology, Nature & Scope of Bioinformatics. The central dogma of molecular biology and bio-sequences associated with it, RNA classification –coding and non coding RNA- mRNA, tRNA, miRNA and sRNA, RNAi. DNA and RNA structure – Nucleic Acid structure and function, Genetic Code, Genes and Evolution	6	15%
II	Importance of databases - Biological databases-primary sequence databases, Composite sequence databases- Secondary databases- nucleic acid sequence databases - Protein sequence data bases - structure databases, Types of databases, Data retrieval tools - Entrez	8	15%

[For more study materials>www.ktustudents.in](http://www.ktustudents.in)

FIRST INTERNAL EXAM			
III	Sequence alignment – local/global, pairwise sequence alignment, scoring methods. Needleman and Wunsch algorithm, global and local alignments. Multiple sequence alignment. Scoring matrices: basic concept of a scoring matrix, Matrices for nucleic acid and proteins sequences, PAM and BLOSUM series, principles based on which these matrices are derived. Differences between distance & similarity matrix.	8	20%
IV	Introduction, Advantages, Phylogenetic Trees, Tree topologies, Methods for phylogenetic analysis- Distance Matrix methods, Character based methods. HMM (Hidden Markov Model): Introduction to HMM, Forward algorithm, Viterbi algorithm, applications in Bioinformatics	6	15%
SECOND INTERNAL EXAM			
V	General introduction to Gene expression in prokaryotes and eukaryotes- Prokaryotic Genomes – Gene structure, GC content, Gene Density, Eukaryotic Genomes- Gene structure, GC content, Gene Density, Gene Expression, Transposition, Gene prediction approaches.	8	20%
VI	Protein and RNA structure Prediction: Predicting RNA secondary structure - Nussinov Algorithm, Energy minimisation methods - Zuker Algorithm. Amino Acids, Polypeptide Composition, Protein Structures, Algorithm for protein folding, Structure prediction	6	15%
END SEMESTER EXAM			

Question Paper Pattern (End semester exam)

1. There will be **FOUR** parts in the question paper – A, B, C, D
2. **Part A**
 - a. **Total marks : 40**
 - b. **TEN** questions, each have **4 marks**, covering **all the SIX modules (THREE** questions from **modules I & II; THREE** questions from **modules III & IV; FOUR** questions from **modules V & VI)**.
All the TEN questions have to be answered.
3. **Part B**
 - a. **Total marks : 18**
 - b. **THREE** questions, each having **9 marks**. One question is from **module I**; one question is from **module II**; one question **uniformly** covers **modules I & II**.
 - c. **Any TWO** questions have to be answered.
 - d. Each question can have **maximum THREE** subparts.
4. **Part C**
 - a. **Total marks : 18**

[For more study materials > www.ktustudents.in](http://www.ktustudents.in)

- b. **THREE** questions, each having **9 marks**. One question is from **module III**; one question is from **module IV**; one question *uniformly* covers **modules III & IV**.
 - c. **Any TWO** questions have to be answered.
 - d. Each question can have *maximum THREE* subparts.
- 5. Part D**
- a. **Total marks : 24**
 - b. **THREE** questions, each having **12 marks**. One question is from **module V**; one question is from **module VI**; one question *uniformly* covers **modules V & VI**.
 - c. **Any TWO** questions have to be answered.
 - d. Each question can have *maximum THREE* subparts.
6. There will be **AT LEAST 60%** analytical/numerical questions in all possible combinations of question choices.



[For more study materials>www.ktustudents.in](http://www.ktustudents.in)

QUESTION BANK

MODULE I				
Q:NO:	QUESTIONS	CO	KL	PAGE NO:
1	Explain Bioinformatics and Point out scopes of bioinformatics	CO1	K2	1
2	Distinguish between RNA and DNA structure with appropriate diagram	CO1	K3	9
3	Explain Central Dogma of Molecular Biology	CO1	K1	13
4	Differentiate between Replication , Transcription and Translation	CO1	K3	16
5	Point out the difference between coding and non coding RNA	CO1	K5	19
6	Define non coding RNA and Classify different types of non coding RNA	CO1	K2	20
7	Explain different characteristics of the genetic code	CO1	K2	22
8	Write a short note on Genetic code	CO1	K2	23
9	Describe nucleic acid and also illustrate its structure	CO1	K2	25
10	Explain functions of nucleic acid structures	CO1	K1	26
MODULE II				
1	Define BLAST and point out its variants	CO2	K2	29
2	Point out types of biological databases	CO2	K1	32

3	Describe protein sequence database and its classification	CO2	K2	36
4	Write a short note on Entrez	CO2	K2	40
5	Classify different types of data retrieval tools	CO2	K4	41
6	Explain nucleic acid sequence database	CO2	K1	42
7	Point out the classifications of protein sequence database	CO2	K4	47
8	Differentiate between primary sequence databases, Composite sequence databases and Secondary databases	CO2	K5	51
9	Write a short note on structure database	CO2	K2	53
MODULE III				
1	Explain scoring alignment and Point out its example	CO3	K1	57
2	Distinguish between global and local alignments	CO3	K4	59
3	Describe the Needleman and Wunsch algorithm using appropriate example	CO3	K3	63
4	Classify methods used in multiple sequence alignments	CO3	K4	64
5	Explain BLOSSUM series in detail	CO3	K2	66
6	Classify methods used in pairwise sequence alignments.	CO3	K4	69
7	Classify the difference between Dot-matrix method, Dynamic programming and Word methods	CO3	K4	71
8	Point out the difference between Progressive method, Iterative method and Motif method	CO3	K1	74
9	Describe PAM series	CO3	K2	75
MODULE IV				

1	Write a short note on Hidden markov model	CO4	K2	77
2	Point out advantages of phylogenetic trees	CO4	K1	83
3	Point out methods for phlogenetic trees construction	CO4	K2	85
4	Explain Viterbi algorithm in detail	CO4	K3	89
5	Point out applications of Hidden markov model in biometrics	CO4	K1	93
6	Define Distance Matrix methods	CO4	K2	94
7	Explain Character based methods	CO4	K3	96
8	Describe about Forward algorithm	CO4	K3	98
9	Write a short note on Tree topologies	CO4	K2	99
10	Point out Advantages of phylogenetic trees	CO4	K2	100
MODULE V				
1	Classify the differences between the gene expression of eukaryotes and prokaryotes	CO5	K4	103
2	Define GC content	CO5	K2	107
3	Explain Gene Density	CO5	K3	111
4	Differentiate between the gene structures of eukaryotes and prokaryotes	CO5	K2	112
5	Describe about gc content and gene density of prokaryotes	CO5	K3	113
6	Explain different gene prediction methods	CO5	K2	117
7	Describe about gc content and gene density of Eukaryotes	CO5	K2	119

8	Write a short note about transposition	CO5	K3	118
9	Give a brief description about gene expression of eukaryotic genomes	CO5	K3	121
MODULE 6				
1	Explain about NUSSINOV algorithm	CO6	K4	123
2	Define energy minimization and give an idea about energy minimization methods	CO6	K2	125
3	Describe about ZUCKER algorithm	CO6	K3	128
4	Give a brief description about amino acids	CO6	K4	131
5	Define structure prediction	CO6	K2	133
6	Differentiate the methods used in structure prediction	CO6	K2	134
7	Write a short note on polypeptide chains	CO6	K1	135

APPENDIX 1

CONTENT BEYOND THE SYLLABUS

S:NO;	TOPIC	PAGE NO:
1	Bioconductor	136
2	GenoCAD	136
3	Apache Taverna	136

MODULE NOTES

A *nucleic acid sequence* is a succession of letters that indicate the order of nucleotides forming alleles within a DNA (using GACT) or RNA (GACU) molecule. By convention, sequences are usually presented from the 5' end to the 3' end. For DNA, the sense strand is used. Because nucleic acids are normally linear (unbranched) polymers, specifying the sequence is equivalent to defining the covalent structure of the entire molecule. For this reason, the nucleic acid sequence is also termed the primary structure.

GenBank is your best bet for most sequence searches; it is updated daily, has detailed online help, and lets you do keyword searches of an organism's or enzyme's name to get sequence information. This service can be very slow during peak hours, however.

EMBL (the European Molecular Biology Laboratory) is a flat-file database that isn't quite as easy to use as GenBank, and is usually slow for people in North America since it's based in Europe, but can be useful if you're looking for a limited amount of data and when you are not trying to identify a gene by sequence analysis.

DDBJ (the DNA Databank of Japan) is hard for beginners to use, but it is best for people who would prefer a Japanese-language interface.

GenBank

Summary: For most sequence searches, GenBank is your best bet. It offers a daily exchange of information with other major sequence databases, has a variety of user interfaces, fairly detailed online help (with e-mail addresses for more information if what is already available is not sufficient), and a speedy interface. Because of its popularity, however, GenBank can also be very slow during peak research hours. Very detailed searches or searches with massive amounts of output might be completed more quickly after hours.

Established by the National Center for Biotechnology Information (NCBI), GenBank is a collection of all known DNA sequences from scientists around the world. As of July 1, 1996, approximately 286,000,000 bases and 352,400 sequences are stored in GenBank, and many more are added each day.

Searching GenBank is fairly straightforward and can be done with a variety of search tools. If you are using a forms-capable WWW browser (such as Netscape 1.0 or higher) and if you have never used GenBank before, you will probably want to start your search with a general query. Other means of searching GenBank include:

- BLAST (Basic Local Alignment Search Tool) Searches
- dbEST (Database of Expressed Sequence Tags)
- dbSTS (Database of Sequence Tagged Sites)

Submitting sequences to GenBank is also very easy and is required by most journals before articles pertaining to the sequence are published (this provides easy access to the information for the journal's readers). You can submit sequences via the WWW with BankIt.

EMBL

Summary: EMBL is good to use when you need a limited amount of data and when you are not trying to identify a gene by sequence analysis. However, because EMBL and all of its mirror sites are located in Europe, your connection will be slow more often than not. All of the information submitted to EMBL is mirrored daily in both GenBank and DDBJ, so searching elsewhere might provide the same amount of information in less time.

EMBL is the database for the European Molecular Biology Laboratory. It is a flat-file database that is searched by a multitude of various search engines. EMBL sequences are stored in a form corresponding to the biological state of the information in vivo. Thus, cDNA sequences are stored in the database as RNA sequences, even though they usually appear in the literature as DNA.

[DBGET](#)

DBGET is a science links database that summarizes the major databases for nucleic acids, proteins, ligands, medicine, etc. It could prove useful for those trying to cross-reference information.

[dbEST](#)

dbEST is a subdivision of GenBank specific for queries on expressed sequence tags ("single pass cDNA sequences").

[DDBJ](#)

Summary: Because DDBJ mirrors its information daily with GenBank and EMBL, beginning sequence searchers might want to try a database with a friendlier searching interface. However, DDBJ also offers all of its pages in Japanese as well, so if you are more comfortable reading the Japanese versions of the pages, it can be very useful.

DDBJ, the DNA Data Bank of Japan, was established in 1986 to be one of the major international DNA Databases (with GenBank and EMBL). It is certified to collect information from researchers and assign accession numbers to submitted entries.

[Searching DDBJ](#) is somewhat awkward, as the only way to access most of the data is by its accession number via anonymous FTP.

Introduction to Biological Databases

1. Introduction

As biology has increasingly turned into a **data-rich science**, the need for storing and communicating large datasets has grown tremendously. The obvious examples are the nucleotide sequences, the protein sequences, and the 3D structural data produced by X-ray crystallography and macromolecular NMR. A new field of science dealing with issues, challenges and new possibilities created by these databases has emerged: **bioinformatics**.

Bioinformatics is the application of Information technology to store, organize and analyze the vast amount of biological data which is available in the form of sequences and structures of proteins (the building blocks of organisms) and nucleic acids (the information carrier). The biological information of nucleic acids is available as sequences while the data of proteins is available as sequences and structures. Sequences are represented in single dimension where as the structure contains the three dimensional data of sequences.

Sequences and structures are only among the several different types of data required in the practice of the modern molecular biology. Other important data types includes metabolic pathways and molecular interactions, mutations and polymorphism in molecular sequences and structures as well as organelle structures and tissue types, genetic maps, physicochemical data, gene expression profiles, two dimensional DNA chip images of mRNA expression, two dimensional gel electrophoresis images of protein expression, data A biological database is a collection of data that is organized so that its contents can easily be accessed, managed, and updated. There are two main functions of biological databases:

- **Make biological data available to scientists.**
 - As much as possible of a particular type of information should be available in one single place (book, site, and database). Published data may be difficult to find or access and collecting it from the literature is very time-consuming. And not all data is actually published explicitly in an article (genome sequences!).
- **To make biological data available in computer-readable form.**
 - Since analysis of biological data almost always involves computers, having the data in computer-readable form (rather than printed on paper) is a necessary first step.

Data Domains

- Types of data generated by molecular biology research:
 - Nucleotide sequences (DNA and mRNA)
 - Protein sequences
 - 3-D protein structures
 - Complete genomes and maps

- Also now have:
 - Gene expression
 - Genetic variation (polymorphisms)

2. Biological Databases

When Sanger first discovered the method to sequence proteins, there was a lot of excitement in the field of Molecular Biology. Initial interest in Bioinformatics was propelled by the necessity to create databases of biological sequences.

Biological databases can be broadly classified into sequence and structure databases. Sequence databases are applicable to both nucleic acid sequences and protein sequences, whereas structure database is applicable to only Proteins. The first database was created within a short period after the Insulin protein sequence was made available in 1956. Incidentally, Insulin is the first protein to be sequenced. The sequence of Insulin consisted of just 51 residues (analogous to alphabets in a sentence) which characterize the sequence. Around mid nineteen sixties, the first nucleic acid sequence of Yeast tRNA with 77 bases (individual units of nucleic acids) was found out. During this period, three dimensional structures of proteins were studied and the well known Protein Data Bank was developed as the first protein structure database with only 10 entries in 1972. This has now grown in to a large database with over 10,000 entries. While the initial databases of protein sequences were maintained at the individual laboratories, the development of a consolidated formal database known as SWISS-PROT protein sequence database was initiated in 1986 which now has about 70,000 protein sequences from more than 5000 model organisms, a small fraction of all known organisms. These huge varieties of divergent data resources are now available for study and research by both academic institutions and industries. These are made available as public domain information in the larger interest of research community through Internet (www.ncbi.nlm.nih.gov) and CDROMs (on request from www.rcsb.org). These databases are constantly updated with additional entries.

Databases in general can be classified in to **primary**, **secondary** and **composite** databases. A **primary** database contains information of the sequence or structure alone. Examples of these include Swiss-Prot & PIR for protein sequences, GenBank & DDBJ for Genome sequences and the Protein Databank for protein structures.

A **secondary** database contains derived information from the primary database. A secondary sequence database contains information like the conserved sequence, signature sequence and active site residues of the protein families arrived by multiple sequence alignment of a set of related proteins. A secondary structure database contains entries of the PDB in an organized way. These contain entries that are classified according to their structure like all alpha proteins, all beta proteins, etc. These also contain information on conserved secondary structure motifs of a particular protein. Some of the secondary database created and hosted by various researchers at their individual laboratories includes SCOP, developed at Cambridge University; CATH developed at University College of London, PROSITE of Swiss Institute of Bioinformatics, eMOTIF at Stanford.

Composite database amalgamates a variety of different primary database sources, which obviates the need to search multiple resources. Different composite database use different primary database and different criteria in their search algorithm. Various options for search

have also been incorporated in the composite database. The National Center for Biotechnology Information (NCBI) which hosts these nucleotide and protein databases in their large high available redundant array of computer servers, provides free access to the various persons involved in research. This also has link to OMIM (Online Mendelian Inheritance in Man) which contains information about the proteins involved in genetic diseases.

2.1 Primary Nucleotide Sequence Repository – GenBank, EMBL, DDBJ

These are three chief databases that store and make available raw nucleic acid sequences. GenBank is physically located in the USA and is accessible through NCBI portal over internet. EMBL (European Molecular Biology Laboratory) is in UK and DDJB (DNA databank of Japan) is in Japan. They have uniform data formats (but not identical) and exchange data on daily basis. Here we will describe one of the database formats, GenBank, in detail. The access to GenBank, as to all databases at NCBI is through the Entrez search program. This front end search interface allows a great variety of search options.

Bioinformatics Example: Growthfactor, implicated in parkinson syndrome

Entry in Genbank

LOCUS	AF053749	1943 bp	DNA	PRI	09-JUL-1999
DEFINITION	Homo sapiens glial cell line-derived neurotrophic factor (GDNF) gene, 5' flanking sequence and exon 1.				
ACCESSION	AF053749				
NID	g5430697				
VERSION	AF053749.1 GI:5430697				
KEYWORDS	.				
SOURCE	human.				
ORGANISM	Homo sapiens				
	Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.				
REFERENCE	1 (bases 1 to 1943)				
AUTHORS	Baecker,P.A., Lee,W.H., Verity,A.N., Eglen,R.M. and Johnson,R.M.				
TITLE	Characterization of a promoter for the human glial cell line-derived neurotrophic factor gene				
JOURNAL	Brain Res. Mol. Brain Res. 69 (2), 209-222 (1999)				
MEDLINE	99296655				
REFERENCE	2 (bases 1 to 1943)				
AUTHORS	Baecker,P.A., Lee,W.H., Verity,A.N., Eglen,R.M. and Johnson,R.M.				
TITLE	Direct Submission				
JOURNAL	Submitted (16-MAR-1998) Molecular and Cellular Biochemistry, Roche Bioscience, 3401 Hillview Avenue, Palo Alto, CA 94304, USA				
				

Bioinformatics

Example: Growthfactor, implicated in parkinson syndrome

Entry in Genbank

```

LOCUS       AF053749      1943 bp      DNA           PRI           09-JUL-1999
DEFINITION  Homo sapiens glial cell line-derived neurotrophic factor (GDNF)
            gene, 5' flanking sequence and exon 1.
ACCESSION   AF053749
NID         g5430697
VERSION     AF053749.1   GI:5430697
KEYWORDS    .
SOURCE      human.
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Mammalia;
            Eutheria; Primates; Catarrhini; Hominidae; Homo.
REFERENCE   1 (bases 1 to 1943)
  AUTHORS   Baecker,P.A., Lee,W.H., Verity,A.N., Eglen,R.M. and Johnson,R.M.
  TITLE     Characterization of a promoter for the human glial cell
            line-derived neurotrophic factor gene
  JOURNAL   Brain Res. Mol. Brain Res. 69 (2), 209-222 (1999)
  MEDLINE   99296655
REFERENCE   2 (bases 1 to 1943)
  AUTHORS   Baecker,P.A., Lee,W.H., Verity,A.N., Eglen,R.M. and Johnson,R.M.
  TITLE     Direct Submission
  JOURNAL   Submitted (16-MAR-1998) Molecular and Cellular Biochemistry, Roche
            Bioscience, 3401 Hillview Avenue, Palo Alto, CA 94304, USA
  ....

```

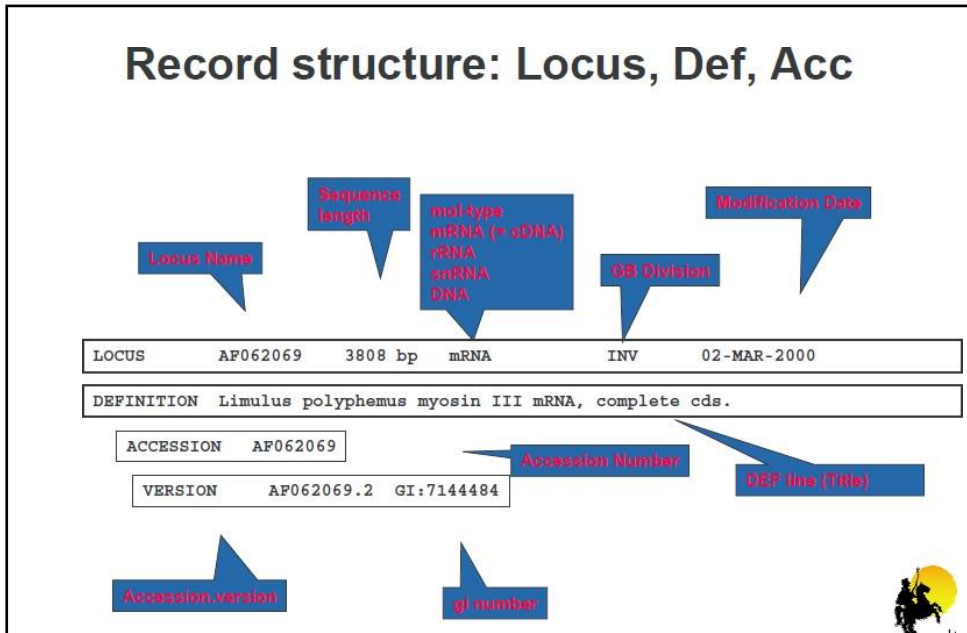
Bioinformatics

Example: Growthfactor, implicated in parkinson syndrome

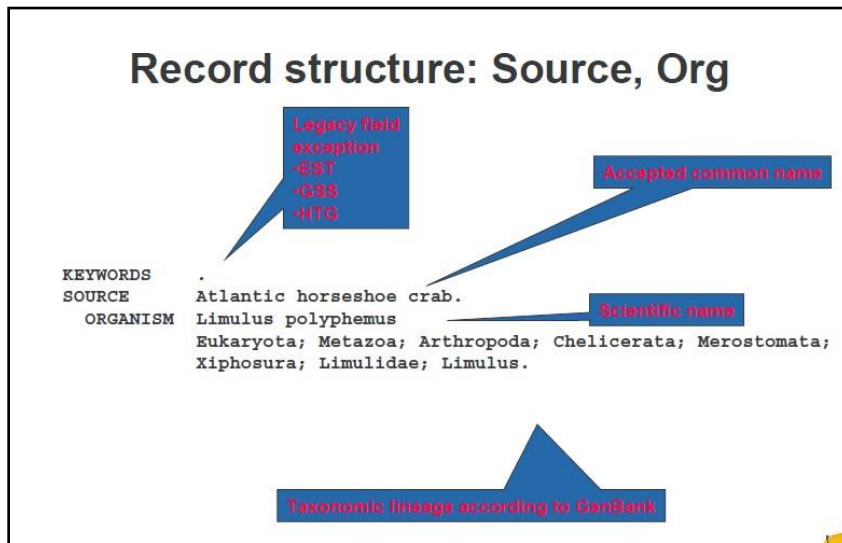
```

FEATURES             Location/Qualifiers
     source            1..1943
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
                     /chromosome="5"
                     /map="5p12-p13.1"
     gene              1..>1943
                     /gene="GDNF"
     misc_feature      1..1643
                     /gene="GDNF"
                     /note="5' flanking region"
     mRNA              1644..>1817
                     /gene="GDNF"
                     /product="glial cell line-derived neurotrophic factor"
     5'UTR             1644..>1817
                     /gene="GDNF"
     exon              1644..1817
                     /gene="GDNF"
                     /number=1
BASE COUNT           356 a    662 c    576 g    349 t
ORIGIN
GAATTCAGGT CCAATGGCTT CCGGAAAACA GTTTCTGCT TAGCAAAGAC ATGCCCTATT      60
TAGTACATTA TTTTAGAGGT ACAGCCAATT CCATGCCCCA TGTGAATGAA ATGTATTTAT      120
GGTTATAGCC ATGCACAGGG TGTGTAAGGA CTGCCCCTCC TCCTGTCCCTC TACAAAAGAA      180
GGCTCAGGCA GCTTCTGGTG GTGAACTAAC CAACAAAAGG AATGCCCAGA AGTCTCACC      240
TCTCCCATCC ACAGAGCTCT GGAATGGGGG CCGGGCCCCT GATCGCTGGA AACTCAGCAT      300
CCAAGTGGGC GCTTGCTGAA GTTCCCATC TGCATTTTCG AAAATCTGGA TAAAAGCAGG      360
TTTAGTCAA  CCTCCCCTAA CCGGTTCCCTG ATAAAGTGAT CTTACGCCCTC TGGAAATTGGG      420
  ....

```




The word accession number defines a field containing unique identification numbers. The sequence and the other information may be retrieved from the database simple by searching for a given accession number. Taking the field names in order, we have first all the word 'LOCUS'. This is a GenBank title that names the sequence entry. Apart for accession number, it also specifies the number of bases in the entry, a nucleic acid type, a codeword PRI that indicates the sequence is from primate, and the date on which the entry was made. PRI is one of the 17 keyword search that are used to classify the data. The next line of the file contains the definition of the entry, giving the name of the sequence. The unique accession number came next, followed by a version number in case the entries have gone through more than one version.



The next item is a list of specially defined keywords that used to index the entries. Next come a set of SOURCE records which describe the organism from which sequence was extracted. The complete scientific classification is given. This is followed by publication details.

Record structure: Citation

REFERENCE	1 (bases 1 to 3808)	Article
AUTHORS	Battelle,B.-A., Andrews,A.W., Calman,B.G., Sellers,J.R., Greenberg,R.M. and Smith,W.C.	
TITLE	A myosin III from Limulus eyes is a clock-regulated phosphoprotein	
JOURNAL	J. Neurosci. (1998) In press	
REFERENCE	2 (bases 1 to 3808)	Submitter Blast
AUTHORS	Battelle,B.-A., Andrews,A.W., Calman,B.G., Sellers,J.R., Greenberg,R.M. and Smith,W.C.	
TITLE	Direct Submission	
JOURNAL	Submitted (29-APR-1998) Whitney Laboratory, University of Florida, 9505 Ocean Shore Blvd., St. Augustine, FL 32086, USA	
REFERENCE	3 (bases 1 to 3808)	Update history
AUTHORS	Battelle,B.-A., Andrews,A.W., Calman,B.G., Sellers,J.R., Greenberg,R.M. and Smith,W.C.	
TITLE	Direct Submission	
JOURNAL	Submitted (02-MAR-2000) Whitney Laboratory, University of Florida, 9505 Ocean Shore Blvd., St. Augustine, FL 32086, USA	
REMARK	Sequence update by submitter	
COMMENT	On Mar 2, 2000 this sequence version replaced gi:3132700.	

[Previous version](#) 

In the beginning, sequences were extracted from the published literature and painstakingly entered in the database. Each entry was therefore associated with a publication. The features table includes coding region, exons, introns, promoters, alternate splice patterns, mutation, variations and a translation into protein sequence, if it code for one. Each feature may be accompanied by a cross-reference to another database. After the feature table, a single line gives the base count statistics for the sequence. Finally comes the sequence itself. The sequence is typed in lower cases, and for ease of reading, each line is divided into six columns of ten bases each. A single number on the left numbers the bases.

Record structure: Features

FEATURES	Location/Qualifiers	
source	1..3808	Biosource
	/organism="Limulus polyphemus"	
	/db_xref="taxon:6850"	
	/tissue_type="lateral eye"	
CDS	258..3302	
	/note="N-terminal protein kinase domain; C-terminal myosin heavy chain head" or PKA"	Reading Frame
	/codon_start=1	
	/product="myosin III"	
	/protein_id="AAC16332.2"	Gene/Protein Identifiers
	/db_xref="GI:7144485"	
	/translation="MEYKCISEHLPPETLPDQDRFEVQELVGTGTATVYSIDK NKKVALKIIIGHIAENLLDIETERYRIYKAVNGIQPFPEFRGAPFKRGERESDNEVWL"	
	"	

[Coding Sequence](#)

Record structure: sequence

Indicates beginning of sequence data

```

BASE COUNT      1201 a    689 c    782 g    1136 t
ORIGIN
  1 tcgacatctg tggtcgcttt ttttagtaat aaaaaattgt attatgacgt cctatctgtt
    <sequence omitted>
 3721 accaatgtta taatatgaaa tgaaataaag cagtcatggt agcagtggtt gtttgaata
 3781 aagatacagt aactagggaa aaaaaaaa
//
  
```

End of record

Databases

Genbank divisions

PRI: primate sequences
ROD: rodent sequences
MAM: other mammalian sequences
VRT: other vertebrate sequences
INV: invertebrate sequences
PLN: plant, fungal and algal sequences
BCT: bacterial sequences
VRL: viral sequences
PHG: bacteriophage sequences
SYN: synthetic sequences
UNA: unannotated sequences
EST: expressed sequence tags
PAT: patent sequences
STS: sequence tag sites
GSS: genome survey sequences
HTC: high throughput cDNA sequences
HTG: high throughput genomic sequences

The above description does not cover all the fields used in GenBank, but only the more important ones.

2.2 Primary Protein Sequence Repositories

PIR-PSD or protein information resource – protein sequence database, at the NBRF (National Biomedical Research Foundation, USA), and SWISS-PROT at the SBI (Swiss Biotechnology Institute), Switzerland are protein sequence databases.

The PIR-PSD is a collaborative endeavour between the PIR, the MIPS (Munich Information Centre for Protein Sequences, Germany) and the JIPIID (Japan International Protein Information Database, Japan). The PIR-PSD is now a comprehensive, non-redundant, expertly annotated, object relational DBMS. It is available at <http://pir.georgetown.edu/pirwww>. A unique characteristic of the PIR-PSD is its classification of protein sequences based on the super family concept. Sequence in PIR-PSD is also classified based on homology domain and sequence motifs. Homology domains may correspond to evolutionary building blocks, while sequence motifs represent functional sites or conserved regions. The classification approach allows a more complete understanding of sequence function structure relationship.

The other well known and extensively used protein database is SWISS-PROT(<http://www.expasy.ch/sprot>). Like the PIR-PSD, this curated proteins sequence database also provides a high level of annotation. The data in each entry can be considered separately as core data and annotation. The core data consists of the sequences entered in common single letter amino acid code, and the related references and bibliography. The taxonomy of the organism from which the sequence was obtained also forms part of this core information. The annotation contains information on the function or functions of the protein, post-translational modification such as phosphorylation, acetylation, etc., functional and structural domains and sites, such as calcium binding regions, ATP-binding sites, zinc fingers, etc., known secondary structural features as for examples alpha helix, beta sheet, etc., the quaternary structure of the protein, similarities to other protein if any, and diseases that may arise due to different authors publishing different sequences for the same protein, or due to mutations in different strains of an described as part of the annotation.

Lines of code in SWISS-PROT database:

Code	Expansion	Remarks
ID	Identification	Occurs at the beginning of the entry. Contains a unique name for the entry, plus information on the status of the entry. If it has been checked and conforms to SWISS-PROT standards, it is called STANDARD.
AC	Accession numbers	This is a stable way of identifying the entry. The name may change but not the AC. If the line has more than one number, it means that the entry was constituted by merging other entries.
DT	Date	There are three dates corresponding to the creation date of the entry and modification dates of the sequence and the annotation respectively
DE	Description	Lines that start with the identifier contain general description about the sequence.
GN	Gene name	The name of the gene (or genes) that codes for the protein
OS, OG,OC	Organism name, Organelle, Organism classification	The name and taxonomy of the organism, and information regarding the organelle containing the gene e.g. mitochondria or chloroplast, etc.
RN, RP,RX,RA RT,RL	Reference number, Position, comments, cross-reference, authors, title and location.	Bibliographic reference to the sequence. This includes information (following the code RP) on the extent of work carried out by the authors.
CC	Comments	These are free text comments that provide any relevant information pertaining to the entry.
DR	Database cross- reference	This line gives cross-references to other databases where information regarding this entry is also found. As for example to structural information for the protein in the PDB.

KW	Keywords	This line gives a list of keywords that can be used in indexes. Search programs very often simply go through such indices to identify required information
FT	Features Table	These lines describe regions or sites of interest in the sequence, e.g. post-translational modifications, binding sites, enzyme active sites and local secondary structures
SQ	Sequence Header	This line indicates the beginning of the sequence data and gives a brief summary of its contents.

Bioinformatics Example: Growthfactor, implicated in parkinson syndrome

Entry in Swiss-Prot

```

ID  GDNF_HUMAN  STANDARD;  PRT;  211 AA.
AC  P39905;
DT  01-FEB-1995 (Rel. 31, Created)
DT  01-FEB-1995 (Rel. 31, Last sequence update)
DT  01-NOV-1997 (Rel. 35, Last annotation update)
DE  GLIAL CELL LINE-DERIVED NEUROTROPHIC FACTOR PRECURSOR.
GN  GDNF.
OS  Homo sapiens (Human).
OC  Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Mammalia;
OC  Eutheria; Primates; Catarrhini; Hominidae; Homo.
RN  [1]
RP  SEQUENCE FROM N.A.
RX  MEDLINE; 93262463.
RA  LIN L.-F.H., DOHERTY D.H., LILE J.D., BEKTESH S., COLLINS F.;
RT  "GDNF: a glial cell line-derived neurotrophic factor for midbrain
RT  dopaminergic neurons.";
RL  Science 260:1130-1132(1993).
RN  [2]
RP  PARTIAL SEQUENCE, AND DISULFIDE BONDS.
RX  MEDLINE; 97141760.
RA  HANJU M., HUI J., YOUNG Y., LE J., KATTA V., LEE R., SHIMAMOTO G.,
RA  ROHDE M.F.;
```

Both PIR-PSD and SWISS-PROT have software that enables the user to easily search through the database to obtain only the required information. SWISS-PROT has the SRS or the sequence retrieval system that searches also through the other relevant databases on the site, such as TrEMBL.

TrEMBL (for Translated EMBL) is a computer-annotated protein sequence database that is released as a supplement to SWISS-PROT. It contains the translation of all coding sequences present in the EMBL Nucleotide database, which have not been fully annotated. Thus it may contain the sequence of proteins that are never expressed and never actually identified in the organisms.

2.3 Derived or Secondary databases of nucleotide sequences

Many of the secondary databases are simply sub-collection of sequences culled from one or the other of the primary databases such as GenBank or EMBL. There is also usually a great deal of value addition in terms of annotation, software, presentation of the information and the cross-references. There are other secondary databases that do not present sequences at all, but only information gathered from sequences databases.

An example of the former type of database is the FlyBase or The Berkeley Drosophila Genome Project (<http://www.fruitfly.org>). A consortium sequenced the entire genome of the fruit fly *D. Melanogaster* to a high degree of completeness and quality.

Another database that focuses on a single organism is ACeDB. More than a database, this is a database management system that was originally developed for the *C. Elegans* (a nematode worm) genome project. It is a repository of not only the sequence, but also the genetic map as well as phenotypic information about the *C. Elegans* nematode worm.

The comprehensive Microbial Resource maintained by TIGR (The Institute for Genomic Research) at <http://www.tigr.org> allows access to a database called Omniome. This contains all the focus on one organism. Omniome has not only the sequence and annotation of each of the completed genomes, but also has associated information about the organisms (such as taxon and gram stain pattern), the structure and composition of their DNA molecules, and many other attributes of the protein sequences predicted from the DNA sequences. The presence of all microbial genomes in a single database facilitated meaningful multi-genome searches and analysis, for instance, alignment of entire genomes, and comparison of the physical proper of proteins and genes from different genomes etc.

A database of the genomes of mitochondria and other such organelles is available at the Organelle Genome Database at the University of Montreal, Canada, and is called GOBASE (<http://megasun.bch.umontreal.ca/gobase>).

2.4 Derived or Secondary databases of amino acid sequences - Subcollections

Another family of a database focussed on a particular family protein is GPCRGB (<http://rose.man.pozen.pl/aars/>). These are transmembrane protein used by cells to communicate with the outside world. They are involved in vision, smell, hearing, taste and feeling. GPCRGB is in fact more than a collection of sequences of the protein family. It includes additional data on multiple sequences alignments. Ligands and ligands binding data, 3D models, mutation data, literature reference, disease patterns, cell lines, protocols, vectors etc. It is fully integrated information system with data, and browsing and query tools.

MHC Pep (<http://wehih.wehi.edu.au/mhcpep/>) is a database comprising over 13000 peptide sequences known to bind the Major Histocompatibility Complex of the immune system. Each entry in the database contains not only the peptide sequence, which may be 8 to 10 amino acid long, but in addition has information on the specific MHC molecules to which it binds, the experimental method used to assay the peptide, the degree of activity and the binding affinity observed , the source protein that, when broken down gave rise to this peptide along with other, the positions along the peptide where it anchors on the MHC molecules and references and cross links to other information.

The CluSTR (Cluster of SWISS-PROT and TrEMBL proteins at <http://ebi.ac.uk.clustr>) database offers an automatic classification of the entries in the SWISS-PROT and TrEMBL databases into groups of related proteins. The clustering is based on the analysis of all pair wise comparisons between protein sequences.

Similar to CluSTRr is the COGS or Cluster of Orthologous Groups of database that is accessible at <http://ncbi.nlm.nih.gov/COG>. An orthologous group of proteins is one in which the members are related to each other by evolutionary descent. Such orthology may not be just from one protein to another, and then to another and so on down the line. It may involve one-to-many ad many-to-many evolutionary relationships, and hence the term 'groups'. COGS is thus a database of phylogenetic relationships. The approximately 2500

groups have been divided into 17 broad categories. The utility of COGS, as of CluSTr, is that it helps assign function to new protein sequences without going through tedious biochemical discovery processes.

2.5 Derived or Secondary databases of amino acid sequences – Patterns and Signature

A set of databases collects together patterns found in protein sequences rather than the complete sequences. The patterns are identified with particular functional and/or structural domains in the protein, such as for example, ATP binding site or the recognition site of a particular substrate. The patterns are usually obtained by first aligning a multitude of sequences through multiple alignment techniques. This is followed by further processing by different methods, depending on the particular database.

PROSITE is one such pattern database, which is accessible at <http://www.expasy.ch/prosite>. The protein motif and pattern are encoded as "regular expressions". The information corresponding to each entry in PROSITE is of the two forms – the patterns and the related descriptive text. The regular expression is placed in a format reminiscent of the SWISS-PROT entries, with a two letter identifier at beginning of the each line specifying the type of information the line contains. The expression itself is placed on line identified by "PA". The entry also contains references and links to all the proteins sequences that contains that pattern. The related descriptive text is placed in a documentation file with the accession number making the connection to the expression data.

In the PRINTS database (<http://www.bioinfo.man.ac.uk/dbbrowser/PRINTS>), the protein sequence patterns are stored as 'fingerprints'. A finger print is a set of motifs or patterns rather than a single one. The information contained in the PRINT entry may be divided into three sections. In addition to entry name, accession number and number of motifs, the first section contains cross links to other databases that have more information about the characterized family. The second section provides a table showing how many of the motifs that make up the finger print occurs in the how many of the sequences in that family. The last section of the entry contains the actual finger prints that are stored as multiply aligned set of sequences, the alignment being made without gaps. There is therefore one set of aligned sequences for each motif.

The ProDom protein domain database (<http://www.toulouse.inrs.fr/prodom.html>) is a compilation of homologous domains that have been automatically identified sequence comparison and clustering methods using the program PSI-BLAST. No identification of patterns is made.. The focus is here to look for complete and self-contained structural domains and the search methods includes signals for such features. A graphical user interface allows easy interactive analysis of structural and therefore functional homology relationships among protein sequences.

A database called Pfam contains the profiles used using Hidden markov models (<http://www.sanger.ac.uk/Software/Pfam>). HMMs build the model of the pattern as a series of match, substitute, insert or delete states, with scores assigned for alignment to go from one state to another. Each family or pattern defined in the Pfam consists of the four elements. The first is the annotation, which has the information on the source to make the entry, the method used and some numbers that serve as figures of merit. The second is the seed alignment that is used to bootstrap the rest of the sequences into the multiple

alignments and then the family. The third is the HMM profile. The fourth element is complete alignment of all the sequences identified in that family.

2.6 Structure Databases

Structure databases like sequence databases come in two varieties, primary and secondary. Strictly speaking there is only one database that stores primary structural data of biological molecules, namely the PDB. In the context of this database, term macromolecule stretches to cover three orders of magnitude of molecular weight from 1000 Daltons to 1000 kilo Daltons. Small biological and organic molecules have their structures stored in another primary structure database the CSD, which is also widely used in biological studies. This contains the three dimensional structure of drugs, inhibitors and fragments or monomers of the macromolecule.

2.6.1 The primary structure database - PDB and CSD

PDB stands for Protein Databank. In spite of the name, PDB archive the three-dimensional structures of not only proteins but also all biologically important molecules, such as nucleic acid fragments, RNA molecules, large peptides such as antibiotic gramicidin and complexes of protein and nucleic acids. The database holds data derived from mainly three sources. Structure determined by X-ray crystallography form the large majority of the entries. This is followed by structures arrived at by NMR experiments. There are also structures obtained by molecular modelling. The data in the PDB is organized as flat files, one to a structure, which usually means that each file contain one molecule, or one molecular complex.

The Cambridge Structural Database (CSD) was originally a project of the University of Cambridge, which is set up to collect together the published three-dimensional structure of small organic molecules. This excludes proteins and medium sized nucleic acid fragments, but small peptides such as neuropeptides, and monomer and dimers of nucleic acid finds a place in the CSD. Currently CSD holds crystal structures information for about 2.5 lakhs organic and metal organic compounds. All these crystal structures have been obtained using X-ray or neutron diffraction technique. For each entry in the CSD there are three distinct types of information stored. These are categorized as bibliographic information, chemical connectivity information and the three-dimensional coordinates. The annotation data field incorporates all of the bibliographic material for the particular entry and summarized the structural and experimental information for the crystal structure.

2.6.1.1 Derived or Secondary databases of bimolecular structures

NDB stands for Nucleic acid data bases. It is a relational database of three-dimensional structures containing nucleic acid. This encompasses DNA and RNA fragments, including those with unusual chemistry such as NDB, and collections of patterns and motifs such as SCOP, PALI etc. The structures are the same as those found in the PDB and therefore the NDB qualifies to be called a specialized sub collection. However a substantial amount, and, unlike the PDB, the NDB is much more than just a collection of files. The structure of DNA has been classified into A, B and Z polymorphic forms, based on the information specified by authors. Other classes include RNA structures, unusual structures and protein-nucleic acid complexes. These classes of structures are arranged in the form of an ATLAS of Nucleic Acid Containing Structures, which can be browse and searched to obtain the structure or structures required. Each entry in the atlas has information on the

sequence, crystallisation condition, references and details of the parameters and the figures of the merit used in structure solution. The entry has links not only to the coordinated but also to automatically generated graphical views of the molecule. NDB also has also have archives of structural geometries calculated for all the structures or for a subset of them. And finally, the database stores average geometrical parameters for nucleic acids, obtained by statistical analysis of the structures. These parameters are widely used in computer simulations of nucleic acids and their interactions. The NDB may be accessed at <http://ndbserve.rutgers.edu/NDB/>.

The SCOP database (Structural Classification of Proteins: <http://scop.mrc-lmb.cam.ac.uk/scop/>) is a manual classification of protein structures in a hierarchical scheme with many levels. The principal classes are the family, the super family and the fold. SCOP is a searchable and browsable database. In other words, one may either enter SCOP at the top of the hierarchy or examine different folds and families as one pleases, or one may supply a keyword or a phrase to be search the database and retrieve corresponding entries. Once a structure, or a set of structures, has been selected, they may be obtained or viewed wither as graphical images. Each entry also has other annotation regarding function, etc., and links to other databases, including other structural classification such as CATH.

CATH stands for Class, Architecture, Topology and Homologous super family. The name reflects the classification hierarchy used in the database. The structures chosen for classification are a subset of PDB, consisting of those that have been determined to a high degree of accuracy.

Conclusion

The present challenge is to handle a huge volume of data, such as the ones generated by the human genome project, to improve database design, develop software for database access and manipulation, and device data-entry procedures to compensate for the varied computer procedures and systems used in different laboratories. There is no doubt that Bioinformatics tools for efficient research will have significant impact in biological sciences and betterment of human lives.

DNA, genes and chromosomes

Learning objectives

By the end of this learning material you would have learnt about the components of a DNA and the process of DNA replication, gene types and sequencing and the structural properties of a chromosome.

DNA

DNA (or deoxyribonucleic acid) is the molecule that carries the genetic information in all cellular forms of life and some viruses. It belongs to a class of molecules called the nucleic acids, which are polynucleotides - that is, long chains of nucleotides.

Each nucleotide consists of three components:

- a nitrogenous base: cytosine (C), guanine (G), adenine (A) or thymine (T) a five-carbon sugar molecule (deoxyribose in the case of DNA)
- a phosphate molecule

The backbone of the polynucleotide is a chain of sugar and phosphate molecules. Each of the sugar groups in this sugar-phosphate backbone is linked to one of the four nitrogenous bases.

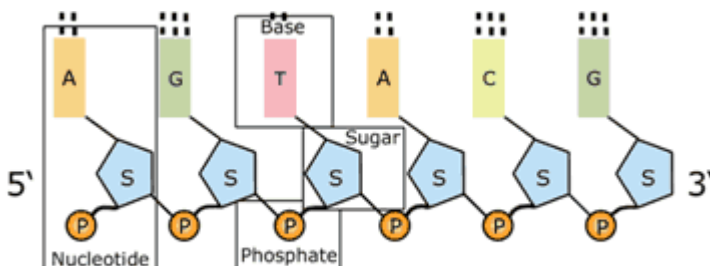


Image adapted from: National Human Genome Research Institute.
White, Sorenson, Data Mining Techniques and Tools for Knowledge Discovery in Agricultural Datasets"

DNA's ability to store - and transmit - information lies in the fact that it consists of two polynucleotide strands that twist around each other to form a double-stranded helix. The bases link across the two strands in a specific manner using hydrogen bonds: cytosine (C) pairs with guanine (G), and adenine (A) pairs with thymine (T).

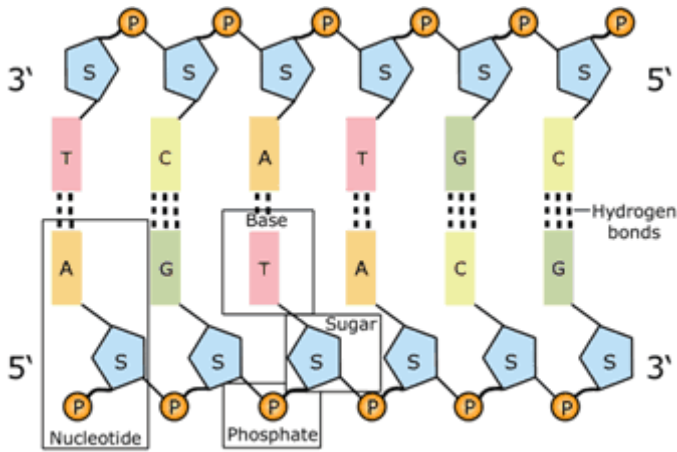


Image adapted from: National Human Genome Research Institute.

The double helix of the complete DNA molecule resembles a spiral staircase, with two sugarphosphate backbones and the paired bases in the centre of the helix. This structure explains two of the most important properties of the molecule. First, it can be copied or 'replicated', as each strand can act as a template for the generation of the complementary strand. Second, it can store information in the linear sequence of the nucleotides along each strand.

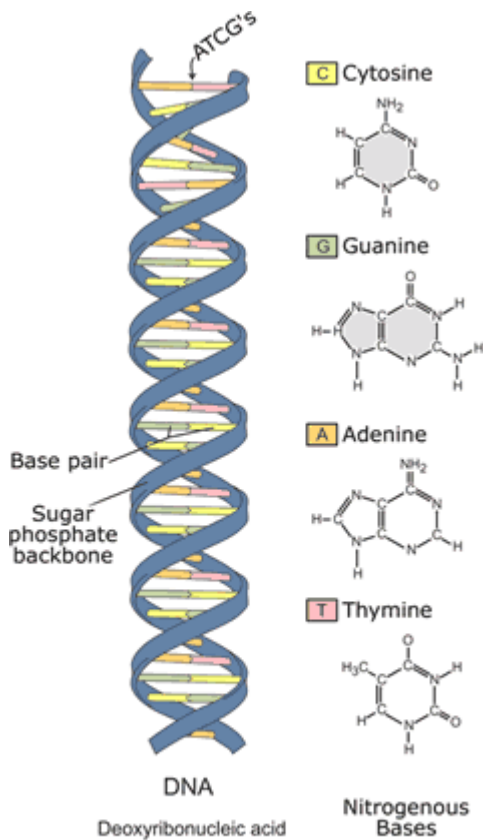
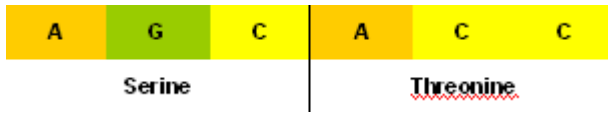


Image adapted from: National Human Genome Research Institute.

It is the order of the bases along a single strand that constitutes the genetic code. The four-letter 'alphabet' of A, T, G and C forms 'words' of three letters called codons. Individual codons code for specific amino acids. A gene is a sequence of nucleotides along a DNA

strand - with 'start' and 'stop' codons and other regulatory elements - that specifies a sequence of amino acids that are linked together to form a protein.

So, for example, the codon AGC codes for the amino acid serine, and the codon ACC codes for the amino acid threonine.



There are two points to note about the genetic code:

- It is **universal**. All life on Earth uses the same code (with a few minor exceptions). It is **degenerate**. Each amino acid can be coded for by more than one codon. For example, AGC and ACC both code for the amino acid serine.

A codon table sets out how the triplet codons code for specific amino acids.

		Second base of codon				
		U	C	A	G	
First base of codon	U	UUU Phenylalanine UUC phe	UCU Serine UCC ser UCA ser UCG ser	UAU Tyrosine UAC tyr UAA STOP codon UAG STOP codon	UGU Cysteine UGC cys UGA STOP codon UGG Tryptophan trp	U C A G
	C	CUU Leucine CUC leu CUA leu CUG leu	CCU Proline CCC pro CCA pro CCG pro	CAU Histidine CAC his CAA Glutamine CAG gin	CGU Arginine CGC arg CGA arg CGG arg	U C A G
	A	AUU Isoleucine AUC ile AUA ile AUG Methionine met (start codon)	ACU Threonine ACC thr ACA thr ACG thr	AAU Asparagine AAC asn AAA Lysine AAG lys	AGU Serine AGC ser AGA Arginine AGG arg	U C A G
	G	GUU Valine GUC val GUA val GUG val	GCU Alanine GCC ala GCA ala GCG ala	GAU Aspartic acid GAC asp GAA Glutamic acid GAG glu	GGU Glycine GGC gly GGA gly GGG gly	U C A G

© Clinical Tools, Inc.

DNA replication

The enzyme helicase breaks the hydrogen bonds holding the two strands together, and both strands can then act as templates for the production of the opposite strand. The process is catalysed by the enzyme DNA polymerase, and includes a proofreading mechanism.

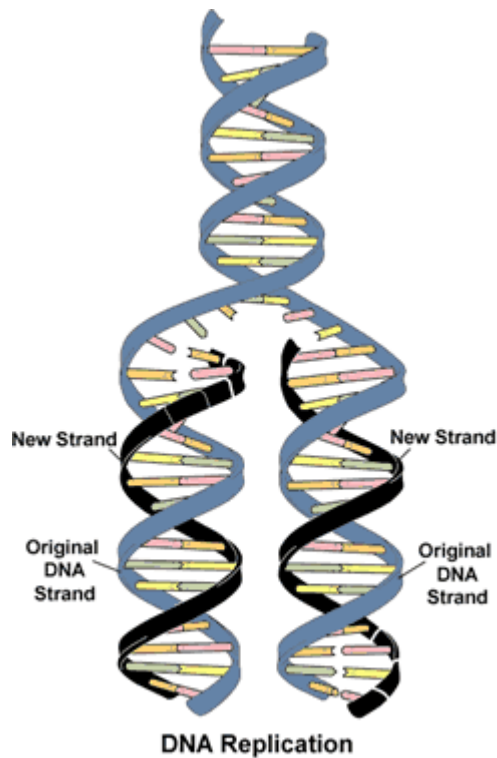


Image adapted from: National Human Genome Research Institute.

Genes

The **gene** is the basic physical and functional unit of heredity. It consists of a specific sequence of nucleotides at a given position on a given chromosome that codes for a specific protein (or, in some cases, an RNA molecule).

Genes consist of three types of nucleotide sequence:

- coding regions, called **exons**, which specify a sequence of amino acids
- non-coding regions, called **introns**, which do not specify amino acids
- regulatory sequences, which play a role in determining when and where the protein is made (and how much is made)

A human being has 20,000 to 25,000 genes located on 46 chromosomes (23 pairs). These genes are known, collectively, as the human genome.

Chromosomes

Eukaryotic chromosomes

The label **eukaryote** is taken from the Greek for 'true nucleus', and eukaryotes (all organisms except viruses, Eubacteria and Archaea) are defined by the possession of a nucleus and other membrane-bound cell organelles.

The nucleus of each cell in our bodies contains approximately 1.8 metres of DNA in total, although each strand is less than one millionth of a centimetre thick. This DNA is tightly packed into structures called **chromosomes**, which consist of long chains of DNA and

associated proteins. In eukaryotes, DNA molecules are tightly wound around proteins - called **histone proteins** - which provide structural support and play a role in controlling the activities of the genes. A strand 150 to 200 nucleotides long is wrapped twice around a core of eight histone proteins to form a structure called a **nucleosome**. The histone octamer at the centre of the nucleosome is formed from two units each of histones H2A, H2B, H3, and H4. The chains of histones are coiled in turn to form a **solenoid**, which is stabilised by the histone H1. Further coiling of the solenoids forms the structure of the chromosome proper.

Each chromosome has a **p arm** and a **q arm**. The p arm (from the French word 'petit', meaning small) is the short arm, and the q arm (the next letter in the alphabet) is the long arm. In their replicated form, each chromosome consists of two **chromatids**.

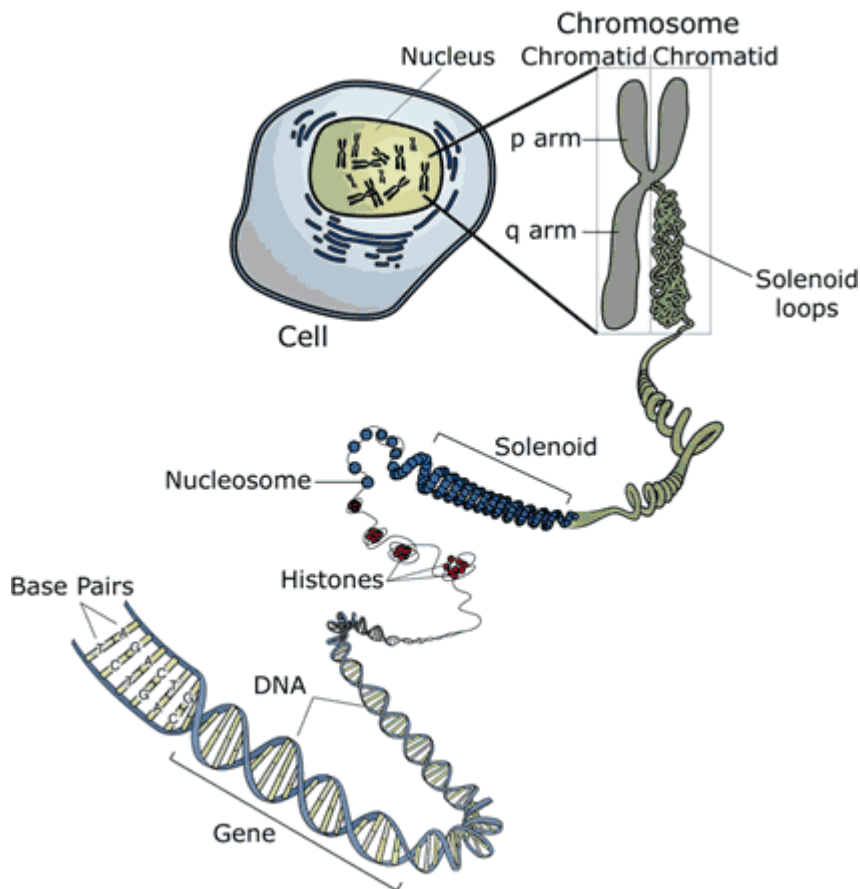
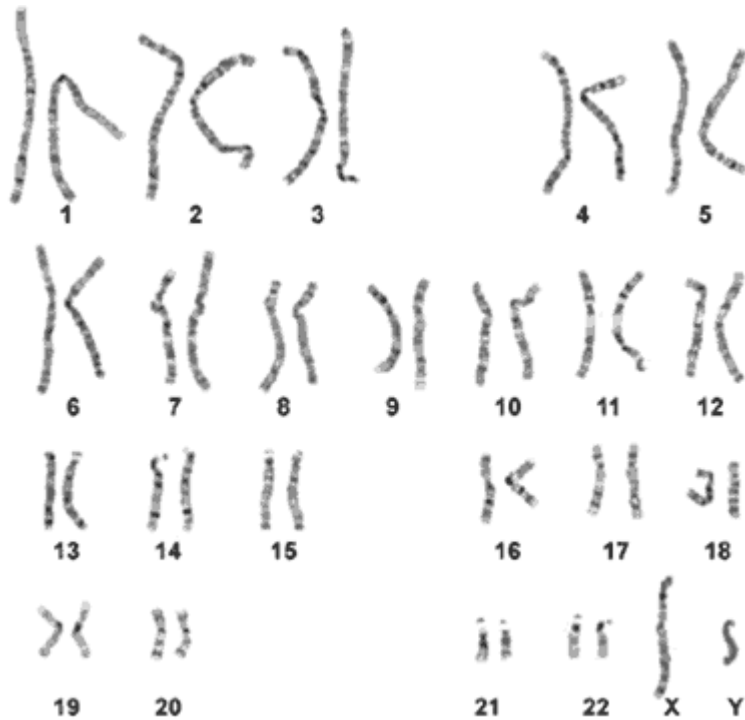


Image adapted from: National Human Genome Research Institute.

The chromosomes - and the DNA they contain - are copied as part of the cell cycle, and passed to daughter cells through the processes of mitosis and meiosis. Human beings have 46 chromosomes, consisting of 22 pairs of **autosomes** and a pair of **sex chromosomes**: two X sex chromosomes for females (XX) and an X and Y sex chromosome for males (XY). One member of each pair of chromosomes comes from the mother (through the egg cell); one member of each pair comes from the father (through the sperm cell).

A photograph of the chromosomes in a cell is known as a **karyotype**. The autosomes are numbered 1-22 in decreasing size order.



© Clinical Tools, Inc.

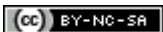
Prokaryotic chromosomes

The **prokaryotes** (Greek for 'before nucleus' - including Eubacteria and Archaea) lack a discrete nucleus, and the chromosomes of prokaryotic cells are not enclosed by a separate membrane.

Most bacteria contain a single, circular chromosome. (There are exceptions: some bacteria - for example, the genus *Streptomyces* - possess linear chromosomes, and *Vibrio cholerae*, the causative agent of cholera, has two circular chromosomes.) The chromosome - together with ribosomes and proteins associated with gene expression - is located in a region of the cell cytoplasm known as the **nucleoid**.

The genomes of prokaryotes are compact compared with those of eukaryotes, as they lack introns, and the genes tend to be expressed in groups known as **operons**. The circular chromosome of the bacterium *Escherichia coli* consists of a DNA molecule approximately 4.6 million nucleotides long.

In addition to the main chromosome, bacteria are also characterised by the presence of extra-chromosomal genetic elements called **plasmids**. These relatively small circular DNA molecules usually contain genes that are not essential to growth or reproduction.



This work is licensed under a [Creative Commons Licence](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Module 2

Biological Databases :

Bioinformatics databases or biological databases are storehouses of biological information. They can be defined as libraries containing data collected from scientific experiments, published literature and computational analysis. It provides users an interface to facilitate easy and efficient recording, storing, analyzing and retrieval of biological data through application of computer software. Biological data comes in several different formats like text, sequence data, structure, links, etc. and these needs to be taken into account while creating the databases.

There are various criteria on the basis of which the databases can be classified. On the basis of structure, databases can be classified as a text file, flat file, object oriented and relational databases. On the basis of information, they can be classified as general and specialized databases. Most commonly, they are classified on the basis of the type of data stored in primary, secondary and composite databases

E-Resources of bioinformatics are important because they are:

- Accessible from any web server
- Commonly used databases are frequently reconciled with each other, so that searching anyone is virtually the same as searching all others
- Support multiple input type for query sequence
- With less effort enable to search homologous data. There are generally two types of Databases:
 - Generalized (DNA, protein, carbohydrate, 3D-structures ...)
 - Specialized (EST, RNA, genomes, protein families, pathways ...)

Biological databases broadly classified in to sequence and structure databases. The biological information of nucleic acids is available as sequences while the data of proteins is available as sequences and structures. Sequences are represented in single dimension where as the structure contains the three dimensional data of sequences.

There are two main functions of biological databases:

- [Make biological data available to scientists.](#)

- As much as possible of a particular type of information should be available in one single place (book, site, and database). Published data may be difficult to find or access and collecting it from the literature is very time-consuming. And not all data is actually published explicitly in an article (genome sequences!).
- To make biological data available in computer-readable form.
 - Since analysis of biological data almost always involves computers, having the data in computer-readable form (rather than printed on paper) is a necessary first step.

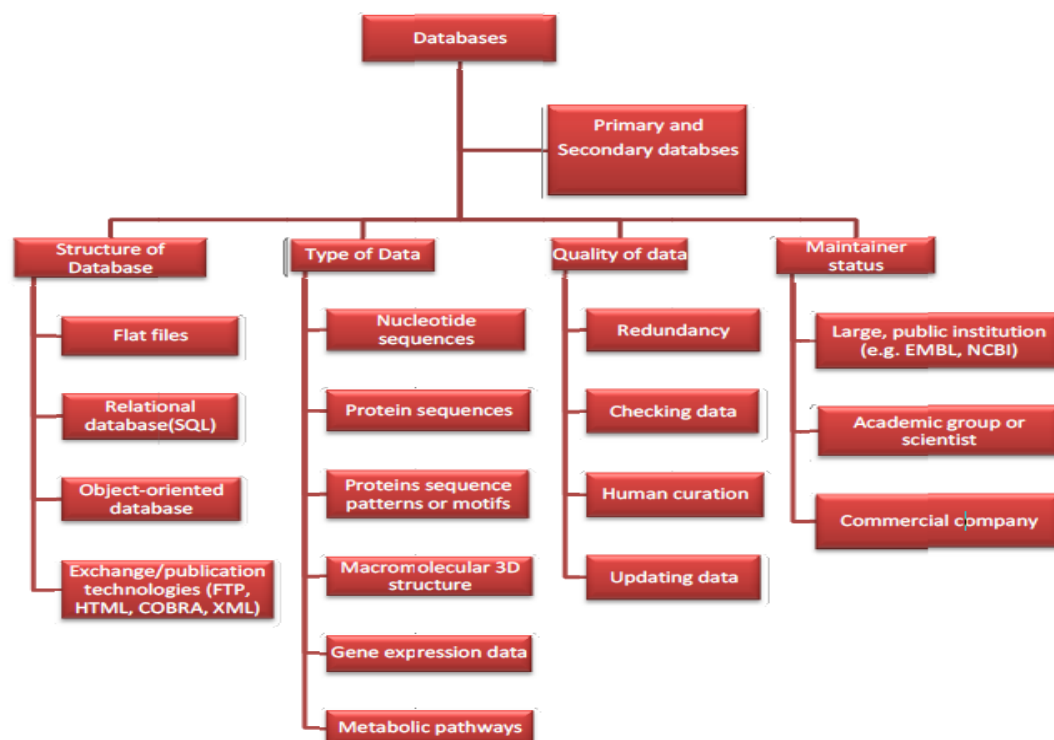


Figure 1: Schematic representation of Database.

Databases in general can be classified in to **primary**, **secondary** and **composite** databases.

Primary Databases: Primary databases contain data that is derived experimentally. They usually store information related to the sequences or structures of biological components. They can be further divided into protein or nucleotide databases which can be further divided as sequence or structure databases. The most commonly used primary databases are: DNA Data Bank of Japan (DDBJ), European Molecular

Biology Laboratory (EMBL) Nucleotide Sequence Database, GenBank, and Protein Data Bank (PDB).

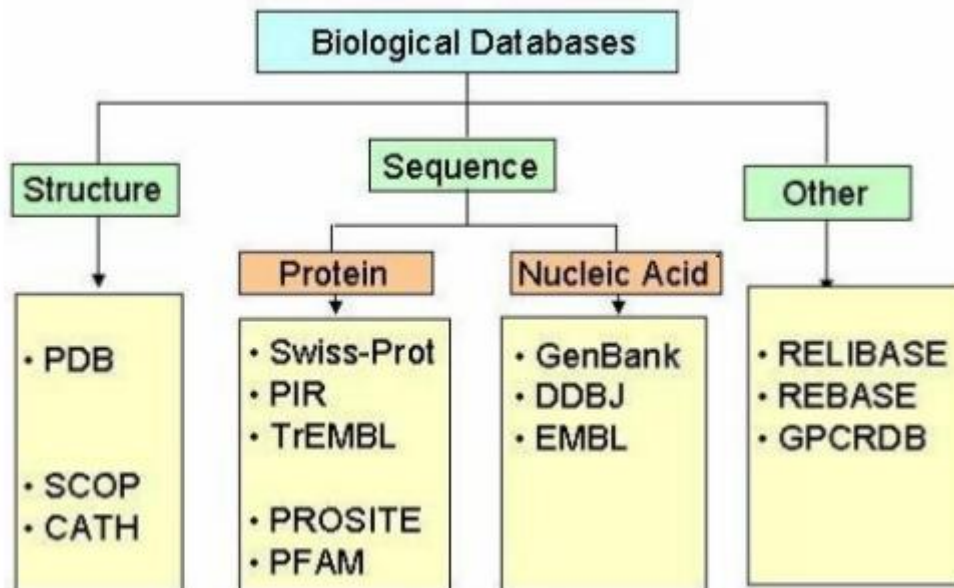
Secondary Database: contains derived information from the primary database. A secondary sequence database contains information like the conserved sequence, signature sequence and active site residues of the protein families arrived by multiple sequence alignment of a set of related proteins. A secondary structure database contains entries of the PDB in an organized way. These contain entries that are classified according to their structure like all alpha proteins, all beta proteins, etc. These also contain information on conserved secondary structure motifs of a particular protein. Some of the secondary database created and hosted by various researchers at their individual laboratories includes SCOP, developed at Cambridge University; CATH developed at University College of London, PROSITE of Swiss Institute of Bioinformatics, eMOTIF at Stanford.

Composite Databases: Composite databases are collections of several (usually more than two) primary database resources. This helps in the lessening the tedious task of searching through multiple databases referring to the same data. The approach used, for instance, the search algorithm employed, differs considerably in every composite database. For example Drug Bank offers details on drug and their targets, BioGraph incorporates assorted knowledge of biomedical science and Bio Model is a storehouse of computational models of the biological developments, etc. There are many composite databases which provide users with various tools and software for analysis of data. NCBI being a composite database has stored a lot of sequence of nucleotide and protein within its server and thereby suffers from high redundancy in the data deposited.

Nucleic acid sequence database :

NDB has developed generalized software for processing, archiving, querying and distributing structural data for nucleic acid-containing structures. The core of the NDB has been its relational database of nucleic acid-containing crystal structures. Recognizing the importance of a standard data representation in building a database, the NDB became an active participant in the mmCIF project and was the test-bed for this format. With a foundation of well curated data, the NDB created a searchable relational database of primary and derivative data with very rich query and reporting capabilities. This robust database was unique in that it allowed researchers to perform comparative analyses of nucleic acid-containing structures selected from the NDB according to the many attributes stored in the database.

Structures available in the NDB include RNA and DNA oligonucleotides with two or more bases either alone or complexed with ligands, natural nucleic acids such as tRNA and protein±nucleic acid complexes. The archive stores both primary and derived information about the structures reference frame



GenBank

Summary: For most sequence searches, GenBank is your best bet. It offers a daily exchange of information with other major sequence databases, has a variety of user interfaces, fairly detailed online help (with e-mail addresses for more information if what is already available is not sufficient), and a speedy interface. Because of its popularity, however, GenBank can also be very slow during peak research hours. Very detailed searches or searches with massive amounts of output might be completed more quickly after hours.

Established by the National Center for Biotechnology Information (NCBI), GenBank is a collection of all known DNA sequences from scientists around the world. As of July 1, 1996, approximately 286,000,000 bases and 352,400 sequences are stored in GenBank, and many more are added each day. Other means of searching GenBank include:

- BLAST (Basic Local Alignment Search Tool) Searches
- dbEST (Database of Expressed Sequence Tags)
- dbSTS (Database of Sequence Tagged Sites)

Submitting sequences to GenBank is also very easy and is required by most journals before articles pertaining to the sequence are published (this provides easy access to the information for the journal's readers). You can submit sequences via the WWW with BankIt.

EMBL

Summary: EMBL is good to use when you need a limited amount of data and when you are not trying to identify a gene by sequence analysis. However, because EMBL and all of its mirror sites are located in Europe, your connection will be slow more often than not. All of the information submitted to EMBL is mirrored daily in both GenBank and DDBJ, so searching elsewhere might provide the same amount of information in less time.

EMBL is the database for the European Molecular Biology Laboratory. It is a flat-file database that is searched by a multitude of various search engines. EMBL sequences are stored in a form corresponding to the biological state of the information in vivo. Thus, cDNA sequences are stored in the database as RNA sequences, even though they usually appear in the literature as DNA.

DBGET

DBGET is a science links database that summarizes the major databases for nucleic acids, proteins, ligands, medicine, etc. It could prove useful for those trying to cross-reference information.

dbEST

dbEST is a subdivision of GenBank specific for queries on expressed sequence tags ("single pass cDNA sequences").

DDBJ

Summary: Because DDBJ mirrors its information daily with GenBank and EMBL, beginning sequence searchers might want to try a database with a friendlier searching interface. However, DDBJ also offers all of its pages in Japanese as well, so if you are more comfortable reading the Japanese versions of the pages, it can be very useful.

DDBJ, the DNA Data Bank of Japan, was established in 1986 to be one of the major international DNA Databases (with GenBank and EMBL). It is certified to collect information from researchers and assign accession numbers to submitted entries.

Searching DDBJ is somewhat awkward, as the only way to access most of the data is by its accession number via anonymous FTP.

Protein sequence databases :

A variety of protein sequence databases exist, ranging from simple sequence repositories, which store data with little or no manual intervention in the creation of the records, to expertly curated universal databases that cover all species and in which the original sequence data are enhanced by the manual addition of further information in each sequence record.

- identify large numbers of proteins
- to map their interactions
- to determine their location within the cell
- analyse their biological activities

SWISS-PROT

SWISS-PROT is a curated protein sequence database which strives to provide a high level of annotation (such as the description of the function of a protein, its domains structure, post-translational modifications, variants, etc.), a minimal level of redundancy and high level of integration with other databases. The SWISS-PROT protein sequence database consists of sequence entries. Sequence entries are composed of different line types, each with their own format.

The SWISS-PROT database distinguishes itself from other protein sequence databases by three distinct criteria: (i) annotations, (ii) minimal redundancy and (iii) integration with other databases.

Annotation includes the description of properties such as the function of a protein, post-translational modifications, domains and sites, secondary and quaternary structure, similarities to other proteins, diseases associated with deficiencies in a protein, developmental stages in which the protein is expressed, in which tissues the protein is found, pathways in which the protein is involved, and sequence conflicts and variants.

The database is non-redundant, which means that all reports for a given protein are merged into a single entry, and is highly integrated with other Databases

PIR:

Protein Information Resource (PIR) has been providing the scientific community with databases and tools for the organization and analysis of protein sequence data. The PIR is the only sequence database to provide context cross-references between its own database entries. These cross-references assist searchers in exploring relationships such as subunit associations in molecular complexes, enzyme–substrate interactions,

activation and regulation cascades, as well as in browsing entries with shared features and annotations.

PIR-PSD :

The oldest universal curated protein sequence database is the Protein Information Resource Protein Sequence Database. It compiles comprehensive, non-redundant protein sequence data, organized by superfamily and family, and annotated with functional, structural, bibliographic and genetic data. In addition to the sequence data, the database contains the name and classification of the protein, the name of the organism in which it naturally occurs, references to the primary literature, function and general characteristics of the protein, and regions of biological interest within the sequence. The database is extensively cross-referenced with DDBJ/EMBL/GenBank nucleic acid and protein identifiers, PubMed and MEDLINE IDs, and unique identifiers from many other source databases.

PIR-NREF :

The PIR-NREF is a Non-redundant REFerence database that provides a timely and comprehensive collection of protein sequence data, keeping pace with the genome sequencing projects and containing source attribution and minimal redundancy. The database contains all sequences in PIR-PSD, Swiss-Prot, TrEMBL, RefSeq, GenPept, and PDB, totaling almost 1,000,000 entries currently. Identical sequences from the same source organism (species) reported in different databases are presented as a single NREF entry with protein IDs, accession numbers, and protein names from each underlying database, as well as amino acid sequence, taxonomy, and composite bibliographic data. NREF can be used for sequence searching and protein identification against the entire sequence collection or a subset of one or more genomes. The collective protein names, including synonyms, and the bibliographic information can be used to develop a protein name ontology. The different protein names assigned by different databases may help detect annotation errors, especially those resulting from large-scale genomic annotation. The web site supports both text and sequence searches. Direct report retrieval is based on sequence unique identifiers of the source databases. The text search matches protein and species names using combinations of text strings

TrEMBL:

TrEMBL (Translation from EMBL) database was introduced to make new sequences available as quickly as possible . TrEMBL consists of computer-annotated entries derived from the translation of all coding sequences in the DDBJ/EMBL/ GenBank nucleotide sequence database that are not yet included in Swiss-Prot. To ensure completeness, it also contains several protein sequences extracted from the literature or submitted directly by the user community. TrEMBL Release 25.6 of November 2003 contained 1 079 094 entries from more than 62 000 different species.

The production of TrEMBL starts with the translation of coding sequences in the DDBJ/EMBL/GenBank nucleotide sequence database. At this stage, all annotation in a TrEMBL entry derives from the corresponding nucleotide entry. The next steps involve redundancy removal through merging of multiple records and the automated enhancement of the information content in TrEMBL . The process is based on a system of standardized transfer of annotation from well-characterized proteins in Swiss-Prot to unannotated TrEMBL entries belonging to defined groups . To assign entries to these groups, InterPro , an integrated resource of protein families, domains and functional sites, is used.

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'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

Specialized databases includes

- Databases of individual protien
- Protein sequence motifs and active sites

- Metalloprotein sites
- Protein structure databases
- Protein folding databases
- Protein properties databases
- Secreted protein databases, etc..

Structure Databases

Structure databases like sequence databases comes in two varieties, primary and secondary. Strictly speaking there is only one database that stores primary structural data of biological molecules, namely the PDB. In the context of this database, term macromolecule stretches to cover three orders of magnitude of molecular weight from 1000 Daltons to 1000 kilo Daltons. Small biological and organic molecules have their structures stored in another primary structure database the CSD, which is also widely used in biological studies. This contains the three dimensional structure of drugs, inhibitors and fragments or monomers of the macromolecule.

The primary structure database - PDB and CSD

PDB stands for Protein Databank. In spite of the name, PDB archive the three-dimensional structures of not only proteins but also all biologically important molecules, such as nucleic acid fragments, RNA molecules, large peptides such as antibiotic gramicidin and complexes of protein and nucleic acids. The database holds data derived from mainly three sources. Structure determined by X-ray crystallography form the large majority of the entries. This is followed by structures arrived at by NMR experiments. There are also structures obtained by molecular modelling. The data in the PDB is organized as flat files, one to a structure, which usually means that each file contain one molecule, or one molecular complex.

The Cambridge Structural Database (CSD) was originally a project of the University of Cambridge, which is set up to collect together the published three-dimensional structure of small organic molecules. This excludes proteins and medium sized nucleic acid fragments, but small peptides such as neuropeptides,

and monomer and dimers of nucleic acid finds a place in the CSD. Currently CSD holds crystal structures information for about 2.5 lakhs organic and metal organic compounds. All these crystal structures have been obtained using X-ray or neutron diffraction technique. For each entry in the CSD there are three distinct types of information stored. These are categorized as bibliographic information, chemical connectivity information and the three-dimensional coordinates. The annotation data field incorporates all of the bibliographic material for the particular entry and summarized the structural and experimental information for the crystal structure.

[Derived or Secondary databases of bimolecular structures](#)

NDB stands for Nucleic acid data bases. It is a relational database of three-dimensional structures containing nucleic acid. This encompasses DNA and RNA fragments, including those with unusual chemistry such as NDB, and collections of patterns and motifs such as SCOP, PALI etc. The structures are the same as those found in the PDB and therefore the NDB qualifies to be called a specialized sub collection. However a substantial amount, and, unlike the PDB, the NDB is much more than just a collection of files. The structure of DNA has been classified into A, B and Z polymorphic forms, based on the information specified by authors. Other classes include RNA structures, unusual structures and protein-nucleic acid complexes. These classes of structures are arranged in the form of an ATLAS of Nucleic Acid Containing Structures, which can be browse and searched to obtain the structure or structures required. Each entry in the atlas has information on the sequence, crystallisation condition, references and details of the parameters and the figures of the merit used in structure solution. The entry has links not only to the coordinated but also to automatically generated graphical views of the molecule. NDB also has also have archives of structural geometries calculated for all the structures or for a subset of them. And finally, the database stores average geometrical parameters for nucleic acids, obtained by statistical analysis of the structures. These parameters are widely used in computer simulations of nucleic acids and their interactions.

The SCOP database (Structural Classification of Proteins) is a manual classification of protein structures in a hierarchical scheme with many levels. The principal classes are the family, the super family and the fold. SCOP is a searchable and browsable

database. In other words, one may either enter SCOP at the top of the hierarchy or examine different folds and families as one pleases, or one may supply a keyword or a phrase to be search the database and retrieve corresponding entries. Once a structure, or a set of structures, has been selected, they may be obtained or viewed wither as graphical images. Each entry also has other annotation regarding function, etc., and links to other databases, including other structural classification such as CATH.

CATH stands for Class, Architecture, Topology and Homologous super family. The name reflects the classification hierarchy used in the database. The structures chosen for classification are a subset of PDB, consisting of those that have been determined to a high degree of accuracy.

Types of databases :

Flat file databases: the easiest way to store data for non· experts. It is not a true database but the ordered collection of similar files. Data can be useful by indexing and ordering the information from the files. In past databases were stored using this concept like PDB began with flat file system with FORTRAN programs, now it uses object·oriented databases.

Relational databases: Relational databases are storing information in the form of tables and tables store information in the form of rows and columns. Tables may relate with each other on more than one entity based on referential integrity concepts. Management of database is easier for manipulating, searching and retrieving the related information. GenBank uses relational database model for storing sequences at NCBI.

Object oriented databases: Object oriented databases is a new concept introduced for biological information storage. The information is stored in the form of objects like, genetic maps and proteins with the set of attributes. This objects are related with each other with OOPS concept. Relationships amongst these objects are identified.

Data retrieval tools :

There are three data retrieval systems of particular

relevance to molecular biologist: Sequence Retrieval System (SRS), Entrez, DBGET.

These systems allow text searching of multiple molecular biology database and provide links to relevant information for entries that match the search criteria. The three systems differ in the databases they search and the links they have to other information.

Sequence Retrieval System (SRS):

SRS is a homogeneous interface to over 80 biological databases that had been developed at the European Bioinformatics Institute (EBI) at Hinxton, UK (see also SRS help . It includes databases of sequences, metabolic pathways, transcription factors, application results (like BLAST, SSEARCH, FASTA), protein 3-D structures, genomes, mappings, mutations, and locus specific mutations.

The web page listing all the databases contains a link to a description page about the database including the date on which it was last updated. You select one or more of the databases to search before entering your query.

After getting results you choose an alignment algorithm (like CLUSTALW, PHYLIP) enter parameters, and run it. The SRS is highly recommended for use.

Entrez:

Entrez is a molecular biology database and retrieval system. Developed by the National Center for Biotechnology information (NCBI) . It is entry point for exploring distinct but integrated databases. Of the three text-based database systems, Entrez is the easiest to use, but also offers more limited information to search.

DBGET:

DBGET is an integrated database retrieval system, developed at the university of Tokyo. Provided access to 20 databases, one at a time. Having more limited options, the DBGET is less recommended than the two others

Specificity and Sensitivity of the Search Tools

Sensitivity: the ability to detect "true positive" matches. The most sensitive search finds all true matches, but might have lots of false positives

Specificity : the ability to reject "false positive" matches. The most specific search will return only true matches, but might have lots of false negatives

There are three main search tools:FastA, BLAST and SW-search.

The FastA Software Package

FastA is a sequence comparison software that uses the method of Pearson and Lipman. The program compares a DNA sequence to a DNA database or a protein sequence to a protein database. Practically, FastA is a family of programs, which include: FastA, TFastA, Ssearch, etc.

Variants of FastA

FASTA - Compares a DNA query sequence to a DNA database, or a protein query to a protein database, detecting the sequence type automatically.

Versions 2 and 3 are in common use, version 3 having a highly improved score normalization method. It significantly reduces the overlap between the score distributions.

FASTX - Compares a DNA query to a protein database. It may introduce gaps only between codons.

FASTY - Compares a DNA query to a protein database, optimizing gap location, even within codons.

TFASTA - Compares a protein query to a DNA database.

BLAST - Basic Local Alignment Search Tool

Blast programs use a heuristic search algorithm. Blast programs were

designed for fast database searching, with minimal sacrifice of sensitivity for distantly related sequences. The programs search databases in a special compressed format. To use your own private database with Blast, you need to convert it to the blast format.

Variants of BLAST

BLASTN - Compares a DNA query to a DNA database. Searches both strands automatically. It is optimized for speed, rather than sensitivity.

BLASTP - Compares a protein query to a protein database.

BLASTX - Compares a DNA query to a protein database, by translating the query sequence in the 6 possible frames, and comparing each against the database (3 reading frames from each strand of the DNA) searching.

TBLASTN - Compares a protein query to a DNA database, in the 6 possible frames of the database.

TBLASTX - Compares the protein encoded in a DNA query to the protein encoded in a DNA database, in the 6*6 possible frames of both query and database sequences.

BLAST2 - Also called advanced BLAST. It can perform gapped alignments.

PSI-BLAST - (Position Specific Iterated) Performs iterative database searches

The Smith-Waterman Tool

Smith-Waterman (SW) searching method compare query to each sequence in database using the full Smith-Waterman algorithm for pairwise comparisons. It also uses search results to generate statistics. Since SW searching is exhaustive, it is the slowest method. Use a special hardware + software (Biocelerator) to execute the algorithm.

MODULE 3

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. Aligned sequences of nucleotide or amino acid residues are typically represented as rows within a matrix. Gaps are inserted between the residues so that identical or similar characters are aligned in successive columns. Sequence alignments are also used for non-biological sequences, such as calculating the edit distance cost between strings in a natural language or in financial data.

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.

- Alignment specifies which positions in two sequences match

acgtctag	acgtctag	acgtctag
actctag-	-actctag	ac-tctag

2 matches	5 matches	7 matches
5 mismatches	2 mismatches	0 mismatches
1 not aligned	1 not aligned	1 not aligned

Mutations: Insertions, deletions and substitutions

Indel: insertion or deletion of a base with respect to the ancestor sequence

a	c	g	t	c	t	a	g
-	a	c	t	c	t	a	g

Mismatch: substitution (point mutation) of a single base

Global and Local Alignments:

Very short or very similar sequences can be aligned by hand. However, most interesting problems require the alignment of lengthy, highly variable or extremely numerous sequences that cannot be aligned solely by human effort. Instead, human knowledge is applied in constructing algorithms to produce high-quality sequence alignments, and occasionally in adjusting the final results to reflect patterns that are difficult to represent algorithmically (especially in the case of nucleotide sequences). Computational approaches to sequence alignment generally fall into two categories: *global alignments* and *local alignments*. Calculating a global alignment is a form of global optimization that "forces" the alignment to span the entire length of all query sequences. By contrast, local alignments identify regions of similarity within long sequences that are often widely divergent overall. Local

alignments are often preferable, but can be more difficult to calculate because of the additional challenge of identifying the regions of similarity. A variety of computational algorithms have been applied to the sequence alignment problem. These include slow but formally correct methods like dynamic programming. These also include efficient, heuristic algorithms or probabilistic methods designed for large-scale database search, that do not guarantee to find best matches.

Global alignment



Local 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. 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.

Hybrid methods, known as semi-global or "glocal" (short for **global-local**) methods, search for the best possible partial alignment of the two sequences (a subset of one or both starts and one or both ends has to be chosen before aligning). This can be especially useful when the downstream part of one sequence overlaps with the upstream part of the other sequence. In this case, neither global nor local

alignment is entirely appropriate: a global alignment would attempt to force the alignment to extend beyond the region of overlap, while a local alignment might not fully cover the region of overlap. Another case where semi-global alignment is useful is when one sequence is short (for example a gene sequence) and the other is very long (for example a chromosome sequence). In that case, the short sequence should be globally (fully) aligned but only a local (partial) alignment is desired for the long sequence.

Pairwise alignment

Pairwise sequence alignment methods are used to find the best-matching piecewise (local or global) alignments of two query sequences. Pairwise alignments can only be used between two sequences at a time, but they are efficient to calculate and are often used for methods that do not require extreme precision (such as searching a database for sequences with high similarity to a query). The three primary methods of producing pairwise alignments are dot-matrix methods, dynamic programming, and word methods; however, multiple sequence alignment techniques can also align pairs of sequences. Although each method has its individual strengths and weaknesses, all three pairwise methods have difficulty with highly repetitive sequences of low information content - especially where the number of repetitions differ in the two sequences to be aligned. One way of quantifying the utility of a given pairwise alignment is the 'maximum unique match' (MUM), or the longest subsequence that occurs in both query sequences. Longer MUM sequences typically reflect closer relatedness.

Dot-matrix methods

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. None of these limitations apply to Miroppeats alignment diagrams but they have their own particular flaws.

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.

Dynamic programming

The technique of dynamic programming can be applied to produce global alignments via the Needleman-Wunsch algorithm, and local alignments via the Smith-Waterman algorithm. In typical usage, protein alignments use a substitution matrix to assign scores to amino-acid matches or mismatches, and a gap penalty for matching an amino acid in one sequence to a gap in the other. DNA and RNA alignments may use a scoring matrix, but in practice often simply assign a positive match score, a negative mismatch score, and a negative gap penalty. (In standard dynamic programming, the score of each amino acid position is independent of the identity of its neighbors, and therefore base stacking effects are not taken into account. However, it is possible to account for such effects by modifying the algorithm.) A common extension to standard linear gap costs, is the usage of two different gap penalties for opening a gap and for extending a gap. Typically the former is much larger than the latter, e.g. -10 for gap open and -2 for gap extension. Thus, the number of gaps in an alignment is usually reduced and residues and gaps are kept together, which typically makes more biological sense. The Gotoh algorithm implements affine gap costs by using three matrices.

Dynamic programming can be useful in aligning nucleotide to protein sequences, a task complicated by the need to take into account frameshift mutations (usually insertions or deletions). The framesearch method produces a series of global or local pairwise alignments between a query nucleotide sequence and a search set of protein sequences, or vice versa. Its ability to evaluate frameshifts offset by an arbitrary number of nucleotides makes the method useful for sequences containing large numbers of indels, which can be very difficult to align with more efficient heuristic methods. In practice, the method requires large amounts of computing power or a system whose architecture is specialized for dynamic programming. The BLAST and EMBOSS suites provide basic tools for creating translated alignments (though some of these approaches take advantage of side-effects of sequence searching capabilities of the tools). More general methods are available

from both commercial sources, such as *FrameSearch*, distributed as part of the Accelrys GCG package, and Open Source software such as Genewise.

The dynamic programming method is guaranteed to find an optimal alignment given a particular scoring function; however, identifying a good scoring function is often an empirical rather than a theoretical matter. Although dynamic programming is extensible to more than two sequences, it is prohibitively slow for large numbers of sequences or extremely long sequences.

Word methods

Word methods, also known as k -tuple methods, are heuristic methods that are not guaranteed to find an optimal alignment solution, but are significantly more efficient than dynamic programming. These methods are especially useful in large-scale database searches where it is understood that a large proportion of the candidate sequences will have essentially no significant match with the query sequence. Word methods are best known for their implementation in the database search tools FASTA and the BLAST family. Word methods identify a series of short, nonoverlapping subsequences ("words") in the query sequence that are then matched to candidate database sequences. The relative positions of the word in the two sequences being compared are subtracted to obtain an offset; this will indicate a region of alignment if multiple distinct words produce the same offset. Only if this region is detected do these methods apply more sensitive alignment criteria; thus, many unnecessary comparisons with sequences of no appreciable similarity are eliminated.

In the FASTA method, the user defines a value k to use as the word length with which to search the database. The method is slower but more sensitive at lower values of k , which are also preferred for searches involving a very short query sequence. The BLAST family of search methods provides a number of algorithms optimized for particular types of queries, such as searching for distantly related sequence matches. BLAST was developed to provide a faster alternative to FASTA without sacrificing much accuracy; like FASTA, BLAST uses a word search of length k , but evaluates only the most significant word matches, rather than every word match as does FASTA. Most BLAST implementations use a fixed default word length that is optimized for the query and database type, and that is changed only under special circumstances, such as when searching with repetitive or very short query sequences. Implementations can be found via a number of web portals, such as EMBL FASTA and NCBI BLAST. Wani.

Multiple sequence alignment

Multiple sequence alignment is an extension of pairwise alignment to incorporate more than two sequences at a time. Multiple alignment methods try to align all of the sequences in a given query set. Multiple alignments are often used in identifying conserved sequence regions across a group of sequences hypothesized to be evolutionarily related. Such conserved sequence motifs can be used in conjunction with structural and mechanistic information to locate the catalytic active sites of enzymes. Alignments are also used to aid in establishing evolutionary relationships by constructing phylogenetic trees. Multiple sequence alignments are computationally difficult to produce and most formulations of the problem lead to NP-complete combinatorial optimization problems. Nevertheless, the utility of these alignments in bioinformatics has led to the development of a variety of methods suitable for aligning three or more sequences.

Dynamic programming

The technique of dynamic programming is theoretically applicable to any number of sequences; however, because it is computationally expensive in both time and memory, it is rarely used for more than three or four sequences in its most basic form. This method requires constructing the n -dimensional equivalent of the sequence matrix formed from two sequences, where n is the number of sequences in the query. Standard dynamic programming is first used on all pairs of query sequences and then the "alignment space" is filled in by considering possible matches or gaps at intermediate positions, eventually constructing an alignment essentially between each two-sequence alignment. Although this technique is computationally expensive, its guarantee of a global optimum solution is useful in cases where only a few sequences need to be aligned accurately. One method for reducing the computational demands of dynamic programming, which relies on the "sum of pairs" objective function, has been implemented in the MSA software package.

Progressive methods

Progressive, hierarchical, or tree methods generate a multiple sequence alignment by first aligning the most similar sequences and then adding successively less related sequences or groups to the alignment until the entire query set has been incorporated into the solution. The initial tree describing the sequence relatedness is based on pairwise comparisons that may include heuristic pairwise alignment methods similar to FASTA. Progressive alignment results are dependent on the choice of "most related" sequences and thus can be sensitive to inaccuracies in the initial pairwise alignments. Most progressive multiple sequence alignment methods additionally weight the sequences in the query set according to their relatedness,

which reduces the likelihood of making a poor choice of initial sequences and thus improves alignment accuracy.

Many variations of the Clustal progressive implementation are used for multiple sequence alignment, phylogenetic tree construction, and as input for protein structure prediction. A slower but more accurate variant of the progressive method is known as T-Coffee.

Iterative methods

Iterative methods attempt to improve on the heavy dependence on the accuracy of the initial pairwise alignments, which is the weak point of the progressive methods. Iterative methods optimize an objective function based on a selected alignment scoring method by assigning an initial global alignment and then realigning sequence subsets. The realigned subsets are then themselves aligned to produce the next iteration's multiple sequence alignment. Various ways of selecting the sequence subgroups and objective function are reviewed in.

Motif finding

Motif finding, also known as profile analysis, constructs global multiple sequence alignments that attempt to align short conserved sequence motifs among the sequences in the query set. This is usually done by first constructing a general global multiple sequence alignment, after which the highly conserved regions are isolated and used to construct a set of profile matrices. The profile matrix for each conserved region is arranged like a scoring matrix but its frequency counts for each amino acid or nucleotide at each position are derived from the conserved region's character distribution rather than from a more general empirical distribution. The profile matrices are then used to search other sequences for occurrences of the motif they characterize. In cases where the original data set contained a small number of sequences, or only highly related sequences, pseudocounts are added to normalize the character distributions represented in the motif.

Techniques inspired by computer science

A variety of general optimization algorithms commonly used in computer science have also been applied to the multiple sequence alignment problem. Hidden Markov models have been used to produce probability scores for a family of possible multiple sequence alignments for a given query set; although early HMM-based methods produced underwhelming performance, later applications have found them especially effective in detecting remotely related sequences because they are less susceptible to noise created by conservative or semiconservative substitutions. Genetic algorithms and simulated annealing have also been used in optimizing multiple sequence alignment scores as judged by a scoring function like

the sum-of-pairs method. More complete details and software packages can be found in the main article [multiple sequence alignment](#)

NEEDLEMAN and WUNSCH Algorithm

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 uses the solutions to the smaller problems to reconstruct a 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.

Needleman-Wunsch

match = 1 mismatch = -1 gap = -1

		G	C	A	T	G	C	U
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	-1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

Results:

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

Interpretation of the initialization step:

One can interpret the leftmost column in the above figure like this (putting a "handle" before each sequence, annotated as + here):

```
Alignment:  +GCATGCU
             +GATTACA
Score:      0 // Handle matches handle, doesn't win any score

Alignment:  +GCATGCU
             +GATTACA
Score:      0 // 1 gap, score -1

Alignment:  +GCATGCU
             +GATTACA
Score:      0 // 2 gaps, score -2

Alignment:  +GCATGCU
             +GATTACA
Score:      0 // 3 gaps, score -3

Alignment:  +GCATGCU
             +GATTACA
Score:      0 // 4 gaps, score -4

...

The same thing can be done for the uppermost row.
```

This algorithm can be used for any two strings. This guide will use two small DNA sequences as examples as shown in the diagram:

GCATGCU

GATTACA

Constructing the grid

First construct a grid such as one shown in Figure 1 above. Start the first string in the top of the third column and start the other string at the start of the third row. Fill out the rest of the column and row headers as in Figure 1. There should be no numbers in the grid yet.

		G	C	A	T	G	C	U

G								
A								
T								
T								
A								
C								
A								

Choosing a scoring system

Next, decide how to score each individual pair of letters. Using the example above, one possible alignment candidate might be:

12345678

GCATG-CU

G-ATTACA

The letters may match, mismatch, or be matched to a gap (a deletion or insertion (indel):

- Match: The two letters at the current index are the same.
- Mismatch: The two letters at the current index are different.
- Indel (INsertion or DELetion): The best alignment involves one letter aligning to a gap in the other string.

Each of these scenarios is assigned a score and the sum of the score of each pairing is the score of the whole alignment candidate. Different systems exist for assigning scores; some have been outlined in the Scoring systems section below. For now, the system used by Needleman and Wunsch will be used:

- Match: +1
- Mismatch or Indel: -1

For the Example above, the score of the alignment would be 0:

GCATG-CU

G-ATTACA

+-----+ -> $-1*4 + 1*4 = 0$

Filling in the table

Start with a zero in the second row, second column. Move through the cells row by row, calculating the score for each cell. The score is calculated by comparing the scores of the cells neighboring to the left, top or top-left (diagonal) of the cell and adding the appropriate score for match, mismatch or indel. Calculate the candidate scores for each of the three possibilities:

- The path from the top or left cell represents an indel pairing, so take the score of the left and the top cell, and add the score for indel to each of them.
- The diagonal path represents a match/mismatch, so take the score of the top-left diagonal cell and add the score for match if the corresponding bases in the row and column are matching or the score for mismatch if they do not.

The resulting score for the cell is the highest of the three candidate scores.

Given there is no 'top' or 'top-left' cells for the second row only the existing cell to the left can be used to calculate the score of each cell. Hence -1 is added for each shift to the right as this represents an indelible from the previous score. This results in the first row being 0, -1, -2, -3, -4, -5, -6, -7. The same applies to the second column as only the existing score above each cell can be used. Thus the resulting table is:

		G	C	A	T	G	C	U
	0	-1	-2	-3	-4	-5	-6	-7
G	-1							
A	-2							
T	-3							

T	-4							
A	-5							
C	-6							
A	-7							

The first case with existing scores in all 3 directions is the intersection of our first letters (in this case G and G). The surrounding cells are below:

		G
	0	-1
G	-1	X

This cell has three possible candidate sums:

- The diagonal top-left neighbor has score 0. The pairing of G and G is a match, so add the score for match: $0+1 = 1$
- The top neighbor has score -1 and moving from there represents an indel, so add the score for indel: $(-1) + (-1) = (-2)$
- The left neighbor also has score -1, represents an indel and also produces (-2).

The highest candidate is 1 and is entered into the cell:

		G
	0	-1
G	-1	1

The cell which gave the highest candidate score must also be recorded. In the completed diagram in figure 1 above, this is represented as an arrow from the cell in row and column 3 to the cell in row and column 2.

In the next example, the diagonal step for both X and Y represents a mismatch:

		G	C
--	--	----------	----------

	0	-1	-2
G	-1	1	X
A	-2	Y	

X:

- Top: $(-2)+(-1) = (-3)$
- Left: $(+1)+(-1) = (0)$
- Top-Left: $(-1)+(-1) = (-2)$

Y:

- Top: $(1)+(-1) = (0)$
- Left: $(-2)+(-1) = (-3)$
- Top-Left: $(-1)+(-1) = (-2)$

For both X and Y, the highest score is zero:

		G	C
	0	-1	-2
G	-1	1	0
A	-2	0	

The highest candidate score may be reached by two or all neighboring cells:

	T	G
T	1	1
A	0	X

- Top: $(1)+(-1) = (0)$

- Top-Left: $(1)+(-1) = (0)$
- Left: $(0)+(-1) = (-1)$

In this case, all directions reaching the highest candidate score must be noted as possible origin cells in the finished diagram in figure 1, e.g. in the cell in row and column 7.

Filling in the table in this manner gives the scores for all possible alignment candidates, the score in the cell on the bottom right represents the alignment score for the best alignment.

Tracing arrows back to origin

Mark a path from the cell on the bottom right back to the cell on the top left by following the direction of the arrows. From this path, the sequence is constructed by these rules:

- A diagonal arrow represents a match or mismatch, so the letters of the column and the letter of the row of the origin cell will align.
- A horizontal or vertical arrow represents an indel. Horizontal arrows will align a gap ("-") to the letter of the row (the "side" sequence), vertical arrows will align a gap to the letter of the column (the "top" sequence).
- If there are multiple arrows to choose from, they represent a branching of the alignments. If two or more branches all belong to paths from the bottom left to the top right cell, they are equally viable alignments. In this case, note the paths as separate alignment candidates.

Following these rules, the steps for one possible alignment candidate in figure 1 are:

U → CU → GCU → -GCU → T-GCU → AT-GCU → CAT-GCU → **G**CATG-CU

A → CA → ACA → TACA → TTACA → ATTACA → -ATTACA → **G**-ATTACA

↓

(branch) → TGCU → ...

→ TACA → ...

Scoring systems

Basic scoring schemes

The simplest scoring schemes simply give a value for each match, mismatch and indel. The step-by-step guide above uses match = 1, mismatch = -1, indel = -1. Thus the lower the alignment score the larger the edit distance, for this scoring system one wants a high score. Another scoring system might be:

- Match = 0
- Indel = 1
- Mismatch = 1

For this system the alignment score will represent the edit distance between the two strings. Different scoring systems can be devised for different situations, for example if gaps are considered very bad for your alignment you may use a scoring system that penalises gaps heavily, such as:

- Match = 0
- Mismatch = 1
- Indel = 10

Identity and similarity- The ratio of identical amino acids residues to the total number of amino acids present in the entire length of the sequence is termed as identity (Figure 39.1). Where as ratio of similar amino acids in a sequence relative to the total number of amino acid present is termed as similarity. The extend of similarity between two amino acids is calculated with a similarity matrix. An alignment between two amino acid sequences is required to calculate identity or similarity score. In the process, two sequence are arbitrarily placed to each other and an alignment score is calculated. This process is repeated until best score is found. In few cases, the length of the amino acids can be enlarged or reduced by incorporating a residue or inserting a gap (Figure 39.1).

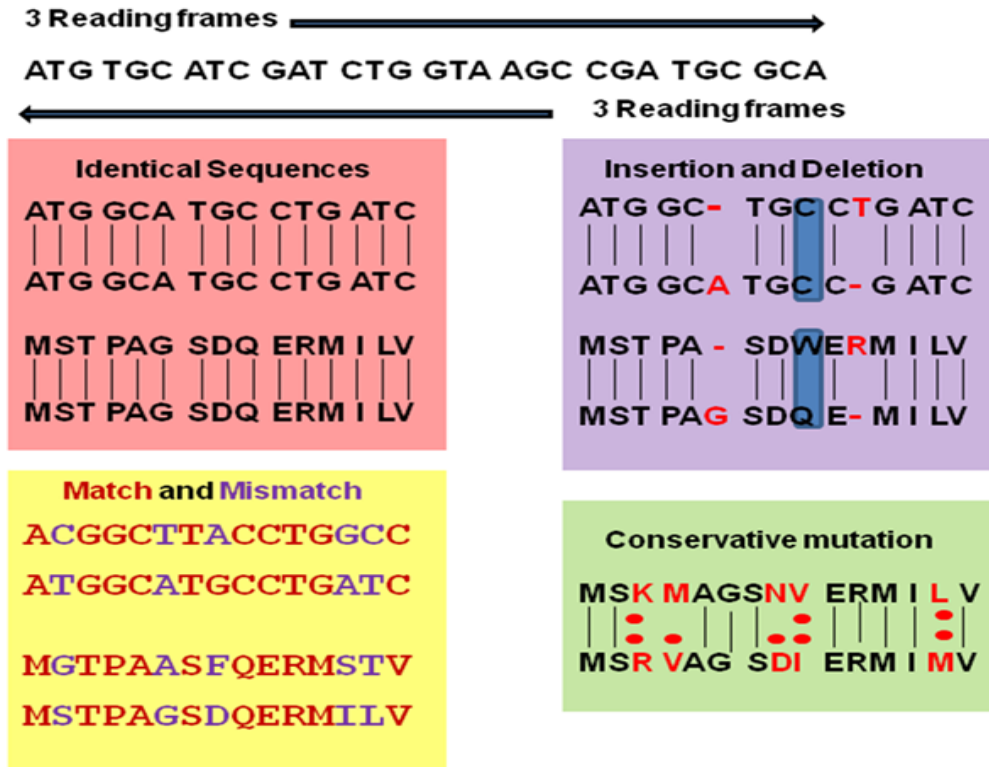


Figure 39.1: Sequence alignment of nucleotide and protein sequences.

The use of a nucleotide scoring matrix to obtain optimal alignment of two nucleotide sequence is given in Figure 39.2. In this case, an identity matrix is relevant as the four nucleotide will not show any similarity to each other. As given the alignment examples, the sliding of the sequences gives different scores (3 or 7 using identity matrix and the alignment with the best score is chosen.

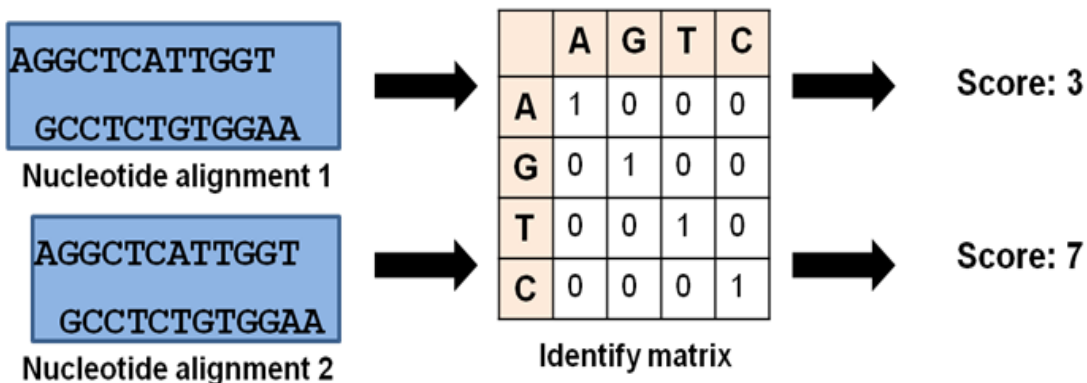


Figure 39.2: Sequence alignment of nucleotide sequences.

Opposite to the nucleotides, identity matrix is not sufficient to perform alignment of two protein sequences. Amino acids present in two sequences may have similar or different physiochemical properties. The probability to substitute one amino acid with other amino acids is also considered to give the score in the matrix. For example, aspartic acid is often observed with glutamic acid but substitution of aspartic acid with tryptophan is rare. This is due to the genetic codes of these amino acids (aspartate and glutamic acid has only 3rd codon different) and their properties (both aspartate and glutamic are negatively charged amino acids). In addition, the effect of substitution on the protein structure is also considered to provide score in the matrix. Aspartate (negatively charged) to tryptophan (aromatic) will have severe impact on the protein structure and hence will have lower score. The most commonly used scoring matrices are the PAM (position assisted matrix) and BLOSUM (blocks substitution matrix). The negative value in the matrix indicates that the occurrence is coincidental whereas positive values suggest a favorable substitution.

PAM:

A **point accepted mutation** — also known as a PAM — is the replacement of a single amino acid in the primary structure of a protein with another single amino acid, which is accepted by the processes of natural selection. This definition does not include all point mutations in the DNA of an organism. In particular, silent mutations are not point accepted mutations, nor are mutations which are lethal or which are rejected by natural selection in other ways.

A PAM matrix is a matrix where each column and row represents one of the twenty standard amino acids. In bioinformatics, PAM matrices are regularly used as substitution matrices to score sequence alignments for proteins. Each entry in a PAM matrix indicates the likelihood of the amino acid of that row being replaced with the amino acid of that column through a series of one or more point accepted mutations during a specified evolutionary interval, rather than these two amino acids being aligned due to chance. Different PAM matrices correspond to different lengths of time in the evolution of the protein sequence.

Each PAM matrix has twenty rows and twenty columns — one representing each of the twenty amino acids translated by the genetic code. The value in each cell of a PAM matrix is related to the probability of a row amino acid before the mutation being aligned with a column amino acid afterwards. From this definition, PAM matrices are an example of a substitution matrix.

BLOSUM:

In bioinformatics, the **BLOSUM (BLOcks SUBstitution Matrix)** matrix is a substitution matrix used for sequence alignment of proteins. BLOSUM matrices are used to score alignments between evolutionarily divergent protein sequences. They are based on local alignments. BLOSUM matrices were first introduced in a paper by Steven Henikoff and Jorja Henikoff. They scanned the BLOCKS database for very conserved regions of protein families (that do not have gaps in the sequence alignment) and then counted the relative frequencies of amino acids and their substitution probabilities. Then, they calculated a log-odds score for each of the 210 possible substitution pairs of the 20 standard amino acids. All BLOSUM matrices are based on observed alignments; they are not extrapolated from comparisons of closely related proteins like the PAM Matrices.

BLOSUM: Blocks Substitution Matrix, a substitution matrix used for sequence alignment of proteins.

Scoring metrics (statistical versus biological): When evaluating a sequence alignment, one would like to know how meaningful it is. This requires a scoring matrix, or a table of values that describes the probability of a biologically meaningful amino-acid or nucleotide residue-pair occurring in an alignment. Scores for each position are obtained frequencies of substitutions in blocks of local alignments of protein sequences.

Several sets of BLOSUM matrices exist using different alignment databases, named with numbers. BLOSUM matrices with high numbers are designed for comparing closely related sequences, while those with low numbers are designed for comparing distant related sequences. For example, BLOSUM80 is used for less divergent alignments, and BLOSUM45 is used for more divergent alignments. The matrices were created by merging (clustering) all sequences that were more similar than a given percentage into one single sequence and then comparing those sequences (that were all more divergent than the given percentage value) only; thus reducing the contribution of closely related sequences. The percentage used was appended to the name, giving BLOSUM80 for example where sequences that were more than 80% identical were clustered.

BLOSUM r: the matrix built from blocks with less than r% of similarity – E.g., BLOSUM62 is the matrix built using sequences with less than 62% similarity (sequences with $\geq 62\%$ identity were clustered) – Note: BLOSUM 62 is the default matrix for protein BLAST. Experimentation has shown that the BLOSUM-62 matrix is among the best for detecting most weak protein similarities.

Similarity Matrices:

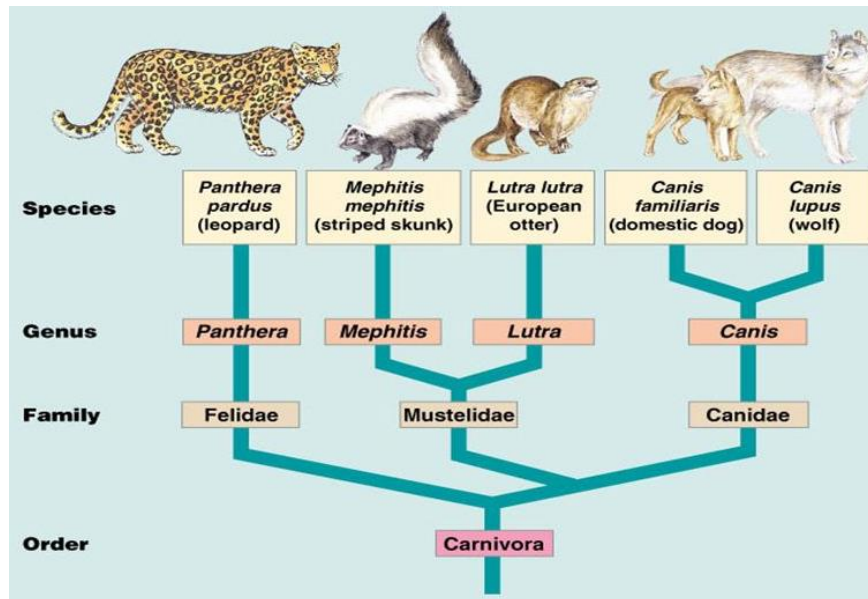
- Similarity matrix is a matrix of score which express the similarity between two data points
- Similarity matrix is used in sequence alignment, higher scores are given to more-similar characters and lower or negative score for dissimilar characters

Distance Matrices:

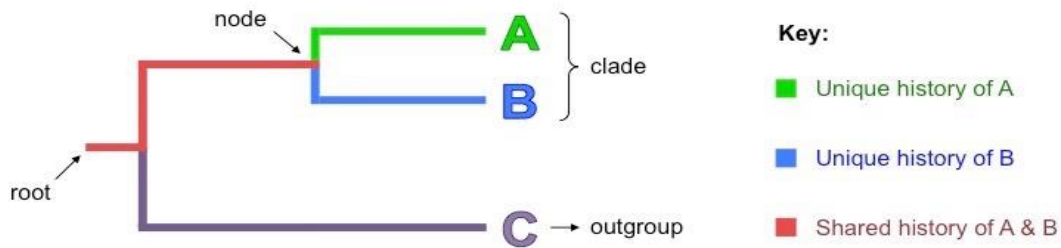
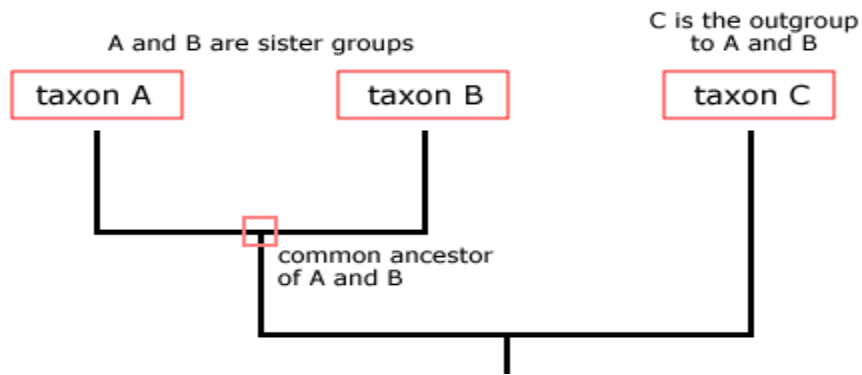
- It is a square matrix containing the distances taken pairwise between the elements of a set
- Distance matrices are used to represent protein structures in a coordinate independent manner, as well as the pairwise distance between two sequence space
- Distance is often defined as the fraction of mismatches at aligned positions with gaps either ignored or counted as mismatches
- Distance method attempt to construct an all to all matrix from the sequence query. Set describing the distances between each sequence pair

MODULE 4

A **phylogenetic tree** or **evolutionary tree** is a branching diagram or "tree" showing the evolutionary relationships among various biological species or other entities—their **phylogeny**—based upon similarities and differences in their physical or genetic characteristics. All life on Earth is part of a single phylogenetic tree, indicating common ancestry.



- A **phylogenetic tree** is a diagram that represents evolutionary relationships among organisms. Phylogenetic trees are hypotheses, not definitive facts.
- The pattern of branching in a phylogenetic tree reflects how species or other groups evolved from a series of common ancestors.
- In trees, two species are **more related** if they have a more recent common ancestor and **less related** if they have a less recent common ancestor.
- Phylogenetic trees can be drawn in various equivalent styles. Rotating a tree about its branch points doesn't change the information it carries.



Advantages :

- Understanding human origin
- Understanding biogeography
- Understanding the origin of particular traits
- Understanding the process of molecular evolution
- Origin of disease
- The aim of phylogenetic tree construction, is to find the tree which best describes the relationships between objects in a set. Usually the objects are species.

- The inference of phylogenies with computational methods has many important applications in medical and biological research, such as drug discovery and conservation biology.
 - Phylogenetic trees have already witnessed applications in numerous practical domains, such as in conservation biology (illegal whale hunting), epidemiology (predictive evolution), forensics (dental practice HIV transmission), gene function prediction and drug development.
-
- Other applications of phylogenies include multiple sequence alignment, protein structure prediction, gene and protein function prediction and drug design.
 - The computation of the tree-of life containing representatives of all living beings on earth is considered to be one of the grand challenges in Bioinformatics.

Tree Topology:

Rooted tree

A rooted phylogenetic tree is a directed tree with a unique node — the root — corresponding to the (usually imputed) most recent common ancestor of all the entities at the leaves of the tree. The root node does not have a parent node, but serves as the parent of all other nodes in the tree. The root is therefore a node of degree 2 while other internal nodes have a minimum degree of 3 (where "degree" here refers to the total number of incoming and outgoing edges).

The most common method for rooting trees is the use of an uncontroversial outgroup—close enough to allow inference from trait data or molecular sequencing, but far enough to be a clear outgroup.

Unrooted tree

Unrooted trees illustrate the relatedness of the leaf nodes without making assumptions about ancestry. They do not require the ancestral root to be known or inferred. Unrooted trees can always be generated from rooted ones by simply omitting the root. By contrast, inferring the root of an unrooted tree requires some means of identifying ancestry. This is normally done by including an outgroup in the input data so that the root is necessarily between the outgroup and the rest of the taxa in the tree, or by introducing additional assumptions about the relative rates of evolution on each branch, such as an application of the molecular clock hypothesis.

Bifurcating tree

Both rooted and unrooted phylogenetic trees can be either bifurcating or multifurcating, and either labeled or unlabeled. A rooted bifurcating tree has exactly two descendants arising from each interior node (that is, it forms a binary tree), and an unrooted bifurcating tree takes the form of an unrooted binary tree, a free tree with exactly three neighbors at each internal node. In contrast, a rooted multifurcating tree may have more than two children at some nodes and an unrooted multifurcating tree may have more than three neighbors at some nodes. A labeled tree has specific values assigned to its leaves, while an unlabeled tree, sometimes called a tree shape, defines a topology only.

Methods of phylogenetic analysis :

Distance matrices are used in phylogeny as non-parametric distance methods and were originally applied to phenetic data using a matrix of pairwise distances. These distances are then reconciled to produce a tree (a phylogram, with informative branch lengths). The distance matrix can come from a number of different sources, including measured distance (for example from immunological studies) or morphometric analysis, various pairwise distance formulae (such as euclidean distance) applied to discrete morphological characters, or genetic distance from sequence, restriction fragment, or allozyme data. For phylogenetic character data, raw distance values can be calculated by simply counting the number of pairwise differences in character states (Hamming distance).

Distance-matrix methods

Distance-matrix methods of phylogenetic analysis explicitly rely on a measure of "genetic distance" between the sequences being classified, and therefore they require an MSA (multiple sequence alignment) as an input. Distance is often defined as the fraction of mismatches at aligned positions, with gaps either ignored or counted as mismatches.[1] Distance methods attempt to construct an all-to-all matrix from the sequence query set describing the distance between each sequence pair. From this is constructed a phylogenetic tree that places closely related sequences under the same interior node and whose branch lengths closely reproduce the observed distances between sequences. Distance-matrix methods may produce either rooted or unrooted trees, depending on the algorithm used to calculate them. They are frequently used as the basis for progressive and iterative types of multiple sequence alignment. The main disadvantage of distance-matrix methods is their inability to efficiently use information about local high-variation regions that appear across multiple subtrees.[2]

Neighbor-joining

Neighbor-joining methods apply general data clustering techniques to sequence analysis using genetic distance as a clustering metric. The simple neighbor-joining method produces unrooted trees, but it does not assume a constant rate of evolution (i.e., a molecular clock) across lineages.

UPGMA and WPGMA

The UPGMA (Unweighted Pair Group Method with Arithmetic mean) and WPGMA (Weighted Pair Group Method with Arithmetic mean) methods produce rooted trees and require a constant-rate assumption - that is, it assumes an ultrametric tree in which the distances from the root to every branch tip are equal.

Fitch-Margoliash method

The Fitch-Margoliash method uses a weighted least squares method for clustering based on genetic distance.[3] Closely related sequences are given more weight in the tree construction process to correct for the increased inaccuracy in measuring distances between distantly related sequences. In practice, the distance correction is only necessary when the evolution rates differ among branches.[2] The distances used as input to the algorithm must be normalized to prevent large artifacts in computing relationships between closely related and distantly related groups. The distances calculated by this method must be linear; the linearity criterion for distances requires that the expected values of the branch lengths for two individual branches must equal the expected value of the sum of the two branch distances - a

property that applies to biological sequences only when they have been corrected for the possibility of back mutations at individual sites. This correction is done through the use of a substitution matrix such as that derived from the Jukes-Cantor model of DNA evolution.

The least-squares criterion applied to these distances is more accurate but less efficient than the neighbor-joining methods. An additional improvement that corrects for correlations between distances that arise from many closely related sequences in the data set can also be applied at increased computational cost. Finding the optimal least-squares tree with any correction factor is NP-complete,^[4] so heuristic search methods like those used in maximum-parsimony analysis are applied to the search through tree space.

Using outgroups

Independent information about the relationship between sequences or groups can be used to help reduce the tree search space and root unrooted trees. Standard usage of distance-matrix methods involves the inclusion of at least one outgroup sequence known to be only distantly related to the sequences of interest in the query set.^[1] This usage can be seen as a type of experimental control. If the outgroup has been appropriately chosen, it will have a much greater genetic distance and thus a longer branch length than any other sequence, and it will appear near the root of a rooted tree. Choosing an appropriate outgroup requires the selection of a sequence that is moderately related to the sequences of interest; too close a relationship defeats the purpose of the outgroup and too distant adds noise to the analysis.^[1] Care should also be taken to avoid situations in which the species from which the sequences were taken are distantly related, but the gene encoded by the sequences is highly conserved across lineages. Horizontal gene transfer, especially between otherwise divergent bacteria, can also confound outgroup usage.

Character Based Methods

What is a character?

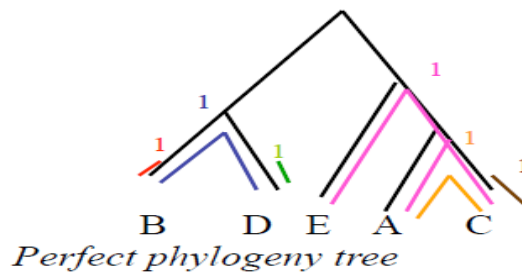
- finite number of states
- discrete
- each character fits on one branch of a phylogenetic tree changes in character happen only once
- species with the same character are all under the same subtree

Parsimony method

- The parsimony score is the number of changes of state on the evolutionary tree. The most parsimonious tree is that which minimizes the amount of evolutionary change.
- The topology is given, parsimony is a method for finding the tree with the least amount of state changes.
- The highest scoring tree minimizes the number of changes.
- Each taxa is described by a set of characters
- Each character can be in one of finite number of states
- In one step certain changes are allowed in character states
- Goal: find evolutionary tree that explains the states of the taxa with minimal number of Changes
- Character states
 - Binary: states are 0 and 1 usually interpreted as presence or absence of an attribute (eg. character is a gene and can be present or absent in a genome)

Example: characters = genes; 0 = absent ; 1 = present
 Taxa: genomes (A,B,C,D,E)

	<i>genes</i>							
A	0	0	0	1	1	0		
B	1	1	0	0	0	0		
C	0	0	0	1	1	1		
D	1	0	1	0	0	0		
E	0	0	0	1	0	0		



Goal: For a given **character state matrix** construct a tree topology that provides perfect phylogeny.

Maximum Likelihood

- Basic idea of **Maximum Likelihood** method is building a tree based on mathematical model.
- This method finds a tree based on probability calculations that best accounts for the large amount of variations of the data (sequences) set.
- **Maximum Likelihood method** (like the Maximum Parsimony method) performs its analysis on each position of the multiple alignment. This is why this method is very heavy on CPU.

The maximum likelihood method uses standard statistical techniques for inferring probability distributions to assign probabilities to particular possible phylogenetic trees. The method requires a substitution model to assess the probability of particular mutations; roughly, a tree that requires more mutations at interior nodes to explain the observed phylogeny will be assessed as having a lower probability. This is broadly similar to the maximum-parsimony method, but maximum likelihood allows additional statistical flexibility by permitting varying rates of evolution across both lineages and sites. In fact, the method requires that evolution at different sites and along different lineages must be statistically independent. Maximum likelihood is thus well suited to the analysis of distantly related sequences, but it is believed to be computationally intractable to compute due to its NP-hardness.

The "pruning" algorithm, a variant of dynamic programming, is often used to reduce the search space by efficiently calculating the likelihood of subtrees. The method calculates the likelihood for each site in a "linear" manner, starting at a node whose only descendants are leaves (that is, the tips of the tree) and working backwards toward the "bottom" node in nested sets. However, the trees produced by the method are only rooted if the substitution model is irreversible, which is not generally true of biological systems. The search for the maximum-likelihood tree also includes a branch length optimization component that is difficult to improve upon algorithmically.

HIDDEN MARKOV MODELS (HMM)

A Markov process is a process that moves from state to state depending on the previous n states. The process is called an ordered n model where n is the number of states affecting the choice of next state.

A HMM is a variation of markov chain in which states in the chain are hidden.

A multinomial model of DNA

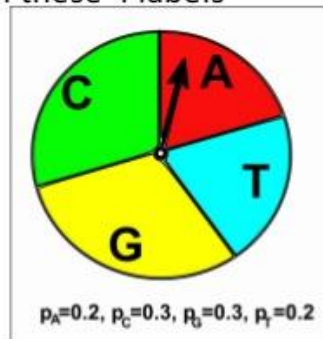
- The **multinomial model** is probably the simplest model of DNA sequence evolution

Assumes the sequence was produced by a process that randomly chose 1 of the 4 bases at each position

The 4 nucleotides are chosen with probabilities p_A, p_C, p_G, p_T

Like having a roulette wheel divided into "A", "C", "G", "T"

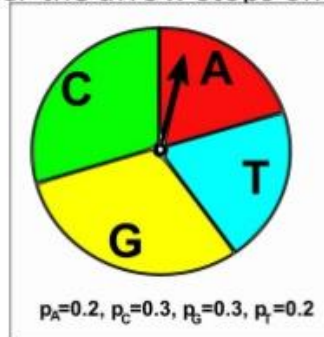
slices, where p_A, p_T, p_G, p_C are the fractions of the wheel taken up by the slices with these 4 labels



- We can generate a sequence using a multinomial model with a certain p_A, p_T, p_G, p_C

eg. if we set $p_A=0.2, p_C=0.3, p_G=0.3, \text{ and } p_T=0.2$, we use the model to generate a sequence of length n , by selecting n bases according to this probability distribution

This is like **spinning the roulette wheel n times in a row**, and recording which letter the arrow stops on each time

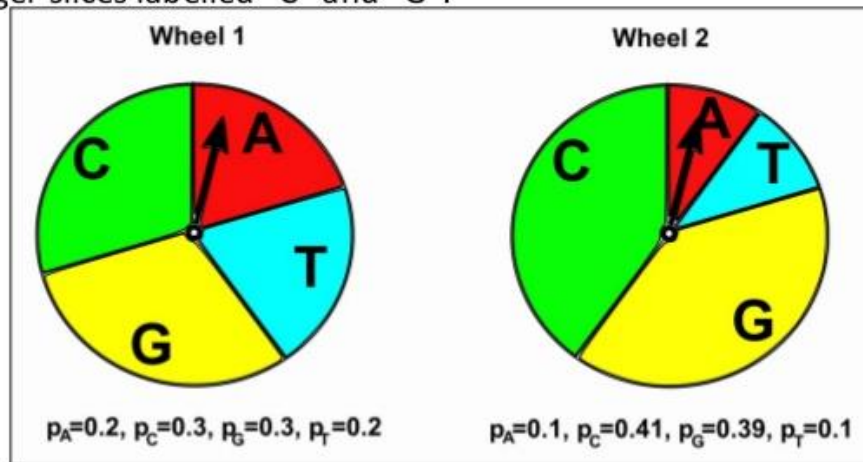


eg. if we make a 1000-bp random sequence using the model, we expect it to be ~20% A, ~30% C, ~30% G, ~20% T

In the same way, we can make a sequence using a multinomial model with $p_A=0.1, p_C=0.41, p_G=0.39, p_T=0.1$

The sequences generated using this 2nd model will have a higher fraction of Cs & Gs compared to those generated using the 1st model (which had $p_C=0.30$ & $p_G=0.30$)

That is, in the 2nd model we are using a roulette wheel that has larger slices labelled "C" and "G":



A Markov model of DNA

- For some DNA sequences, a multinomial model is not an accurate representation of how the sequences have evolved

A multinomial model **assumes each part of the sequence** (eg. bases 1-100, 101-200, etc.) **has the same probability of each of the 4 nucleotides** (same p_A, p_C, p_G, p_T)

May not hold for a particular sequence, if base frequencies differ a lot between different parts of the sequence

Also assumes the probability of choosing a base (eg. "A") at a particular position only depends on the predetermined probability of that base (p_A), & **does not depend on the bases found at adjacent positions in the sequence**

However, this assumption is incorrect for some sequences!

- A **Markov model** takes into account that the probability of a base at a position **depends on what bases are found at adjacent positions**

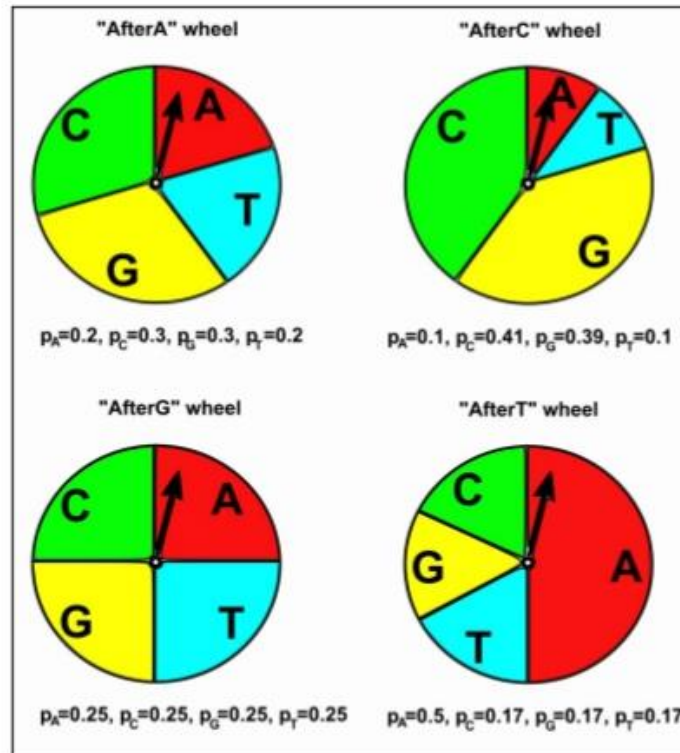
Assumes the sequence was produced by a process where the **probability of choosing a particular base at a position depends on the base chosen for the previous position**

If "A" was chosen at the previous position, a base is chosen at the current position with probabilities p_A, p_C, p_G, p_T , eg. $p_A=0.2$, $p_C=0.3$, $p_G=0.3$, and $p_T=0.2$

If "C" was chosen at the previous position, a base is chosen at the current position with different probabilities eg. $p_A=0.1$, $p_C=0.41$, $p_G=0.39$, and $p_T=0.1$

A Markov model is like having 4 roulette wheels, "afterA", "afterT", "afterG", "afterC", for the cases when "A", "T", "G", or "C" were chosen at the previous position in a sequence, respectively

Each of the 4 roulette wheels has a different p_A , p_T , p_G and p_C

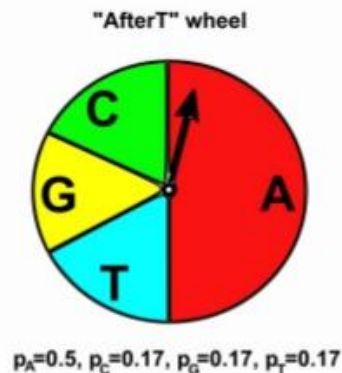


If you are generating a sequence using a Markov model, to choose a base for a particular position in the sequence, you spin the roulette wheel & see where the arrow stops

The particular roulette wheel you use at a particular position in the sequence **depends on the base chosen for the previous position in the sequence**

eg., if "T" was chosen at the previous position, we spin the "afterT" roulette wheel to choose the nucleotide for the current position

The probability of choosing a particular nucleotide at the current position (eg. "A") then depends on the fraction of the "afterT" roulette wheel taken up by the slice labelled with that nucleotide (p_A):



A multinomial model of DNA sequence evolution just has **four parameters**: the **probabilities $p_A, p_C, p_G,$ and p_T**

A Markov model has **16 parameters**: 4 sets of probabilities

$p_{AA}, p_{AC}, p_{AG}, p_{AT}$, that differ according to whether the previous nucleotide was "A", "G", "T" or "C"

$p_{AA}, p_{AC}, p_{AG}, p_{AT}$ are used to represent the probabilities for the case where the previous base was "A"; $p_{CA}, p_{CC}, p_{CG}, p_{CT}$ where the previous base was "C", and so on

We store the probabilities **current position** **ansition matrix**:

		current position			
		A	C	G	T
previous position	A	0.20	0.30	0.30	0.20
	C	0.10	0.41	0.39	0.10
	G	0.25	0.25	0.25	0.25
	T	0.50	0.17	0.17	0.17

The rows represent the base at the previous position in the sequence, the columns the base at the current position

		current position			
		A	C	G	T
previous position	A	0.20	0.30	0.30	0.20
	C	0.10	0.41	0.39	0.10
	G	0.25	0.25	0.25	0.25
	T	0.50	0.17	0.17	0.17

Rows 1, 2, 3, 4 give the probabilities p_A, p_C, p_G, p_T for the cases where the previous base was "A", "C", "G", or "T"

Row 1 gives probabilities $p_{AA}, p_{AC}, p_{AG}, p_{AT}$

Row 2 gives probabilities $p_{CA}, p_{CC}, p_{CG}, p_{CT}$

Row 3 gives probabilities $p_{GA}, p_{GC}, p_{GG}, p_{GT}$

Row 4 gives probabilities $p_{TA}, p_{AT}, p_{TG}, p_{TT}$

This transition matrix will generate random sequences with many "A"s after "T" (because p_{TA} is high, ie. 0.5)

But the sequences will have few "A"s after "C"s ($p_{CA} = 0.1$)

When you are generating a sequence using a Markov model, the **base chosen at each position at the sequence depends on the base chosen at the previous position**

There is no previous base at the 1st position in the sequence, so we define the probabilities of choosing "A", "C", "G" or "T" for the 1st position

Symbols π_A , π_C , π_G , π_T represent the probabilities of choosing "A", "C", "G", or "T" at the 1st position

The probabilities of choosing each of the four nucleotides **at the first position** in the sequence are π_A , π_C , π_G , and π_T

The probabilities of choosing each of the 4 bases **at the 2nd position** depend on the particular base that was chosen at the 1st position in the sequence

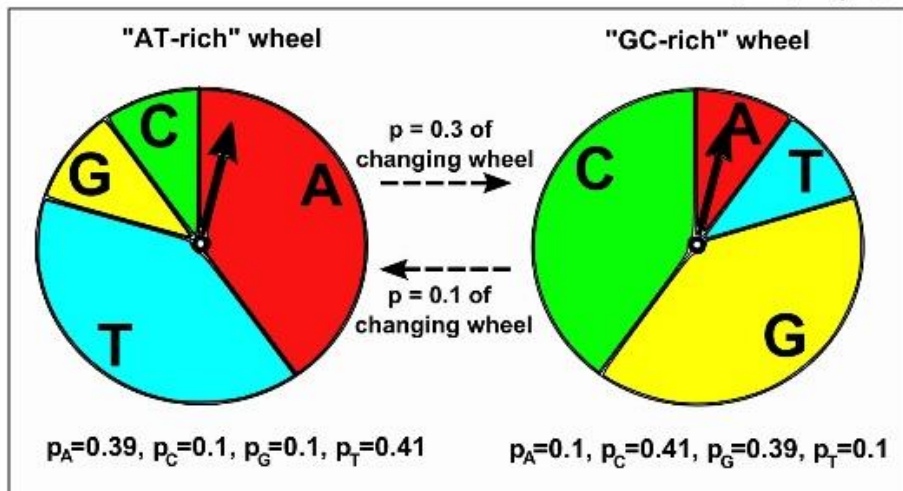
The probabilities of choosing each of the 4 bases **at the 3rd position** depend on the base chosen at the 2nd position

...

A Hidden Markov Model of DNA

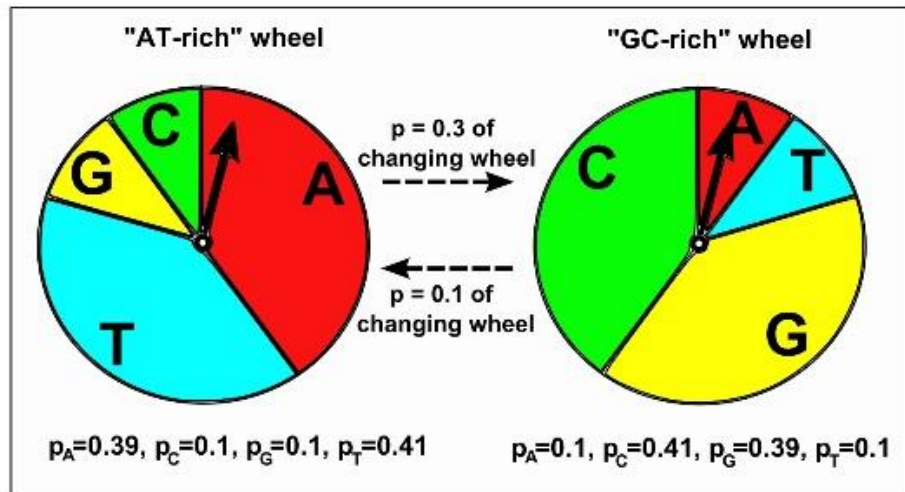
- In a **Markov model**, the base at a particular position in a sequence **depends on the base found at the previous position**
- In a **Hidden Markov model (HMM)**, the base found at a particular position in a sequence depends on the **state** at the previous position
The **state** at a sequence position is a property of that position of the sequence
eg. a HMM may model the positions in a sequence as belonging to one of 2 states, "**GC-rich**" or "**AT-rich**"
A more complex HMM may model the positions in a sequence as belonging to many different possible states, eg. "**exon**", "**intron**", and "**intergenic DNA**"

- A HMM is like having **several different roulette wheels, 1 for each state in the HMM**
 eg. a "GC-rich" roulette wheel, & an "AT-rich" roulette wheel
 Each wheel has "A", "T", "G", "C" slices, & a different % of each wheel is taken up by "A", "T", "G", "C"
 ie. "GC-rich" & "AT-rich" wheels have different p_A, p_T, p_G, p_C



- To generate a sequence using a HMM, choose a base at a position by spinning a particular roulette wheel, & see where the arrow stops
 How do we decide which roulette wheel to use?
 If there are 2 roulette wheels, **we tend to use the same roulette wheel that we used to choose the previous base** in the sequence
But there is also a certain small probability of **switching to the other roulette wheel**
 eg., if we used the "GC-rich" wheel to choose the previous base, there may be a 90% chance we'll use the "GC-rich" wheel to choose the base at the current position
 ie. a 10% chance that we will switch to using the "AT-rich" wheel to choose the base at the current position

If we used the "AT-rich" wheel to choose the base at the previous position, there may be 70% chance we'll use the "AT-rich" wheel again at this position
 ie. a 30% chance that we will switch to using the "GC-rich" roulette wheel to choose the nucleotide at this position



Emission & Transmission Matrices

- A HMM has two important matrices that hold its parameters
- Its **transition matrix** contains the probabilities of changing from 1 state to another

For a HMM with AT-rich & GC-rich states, the transition matrix holds the probabilities of switching from AT-rich state to GC-rich state, & from GC-rich to AT-rich state

eg. if the previous nucleotide was in the AT-rich state there may be a probability of 0.3 that the current base will be in the GC-rich state

Similarly, if the previous nucleotide was in the GC-rich state there may be a probability of 0.1 that the current nucleotide will be in the AT-rich state

That is, the transition matrix would be:

		current position	
		AT-rich	GC-rich
previous position	AT-rich	0.7	0.3
	GC-rich	0.1	0.9

There is a row for each of the possible states at the previous position in the sequence

eg. here the 1st row corresponds to the case where the previous position was in the "AT-rich" state

Here the second row corresponds to the case where the previous position was in the "GC-rich" state

The columns give the probabilities of switching to different states at the current position

eg. the probability of switching to the AT-rich state, if the previous position was in the GC-rich state, is 0.1

- The **emission matrix** holds the probabilities of choosing the 4 bases "A", "C", "G", and "T", in each of the states

In a HMM with AT-rich & GC-rich state, the emission matrix holds the probabilities of choosing "A", "C", "G", "T" in the AT-rich state, and in the GC-rich state

eg. $p_A=0.39$, $p_C=0.1$, $p_G=0.1$, $p_T=0.41$ for the AT-rich state,

$p_A=0.1$, $p_C=0.41$, $p_G=0.39$, $p_T=0.1$ for the GC-rich state

	A	C	G	T
AT-rich	0.39	0.10	0.10	0.41
GC-rich	0.10	0.41	0.39	0.10

There is a row for each state, & columns give probabilities of choosing each of the 4 bases in a particular state

eg. the probability of choosing "G" in the GC-rich state

(when using the GC-rich roulette wheel) is 0.39

- When generating a sequence using a HMM the base is chosen at a position **depending on the state at the previous position**

There is no previous position at the 1st position

You must specify probabilities of the choosing each of the states at the first position

eg. $\pi_{AT-rich}$ & $\pi_{GC-rich}$ being the probabilities of the choosing the “AT-rich” or “GC-rich” states at the 1st position

The probabilities of choosing each of the 2 states **at the first position** in the sequence are $\pi_{AT-rich}$ & $\pi_{GC-rich}$

The probabilities of choosing each of the 4 bases **at position 1** depend on the **particular state chosen at position 1**

The probability of choosing each of the 2 states at the 2nd position depends on the state chosen at position 1

The probabilities of choosing each of the 4 bases at the 2nd position depend on the state chosen at position 2 ...

eg. using the following emission & transmission matrices:

	A	C	G	T	
AT-rich	0.39	0.10	0.10	0.41	emission matrix
GC-rich	0.10	0.41	0.39	0.10	

		current position		
		AT-rich	GC-rich	
previous position	AT-rich	0.7	0.3	transmission matrix
	GC-rich	0.1	0.9	

The nucleotides generated by the GC-rich state will mostly but not all be “G”s & “C”s (because of high values of p_G and p_C for the GC-rich state in the emission matrix)

The nucleotides generated by the AT-rich state will mostly but not all be “A”s & “T”s (because of high values of p_T and p_A for the AT-rich state in the emission matrix)

eg. using the following emission & transmission matrices:

	A	C	G	T	
AT-rich	0.39	0.10	0.10	0.41	emission matrix
GC-rich	0.10	0.41	0.39	0.10	

		current position		
		AT-rich	GC-rich	
previous position	AT-rich	0.7	0.3	transmission matrix
	GC-rich	0.1	0.9	

Furthermore, there will tend to be runs of bases that are either all in the GC-rich state or all in the AT-rich state

Because the probabilities of switching from the AT-rich to GC-rich state (probability 0.3), or GC-rich to AT-rich state (probability 0.1) are relatively low

Inferring the states of a HMM that generated a DNA sequence

- If we have a HMM, & know the transmission & emission matrices, can we take some sequence & figure out **which state is most likely to have generated each base position?**

eg. for a HMM with "GC-rich" & "AT-rich" states, can we take some sequence, & figure out whether the GC-rich or AT-rich state probably generated each base?

- This is called the problem of finding the **most probable state path**

This problem essentially consists of **assigning the most likely state to each position in the DNA sequence**

- This problem is also sometimes called **segmentation**

eg. given a sequence of 1000 bp, you wish to use your HMM to **segment the sequence** into blocks that were probably generated by the "GC-rich" or "AT-rich" state

- The problem of **segmenting a sequence using a HMM** can be solved by an algorithm called the **Viterbi algorithm**

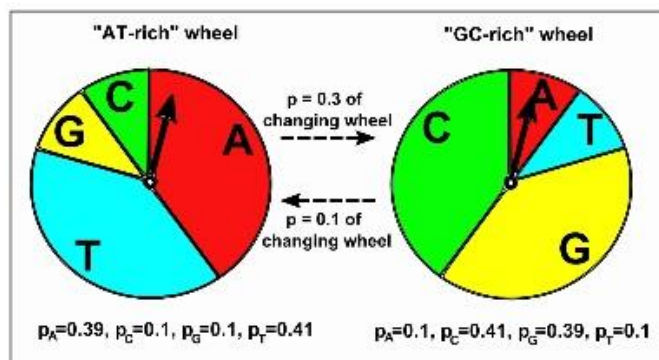
The Viterbi algorithm gives, for each base position in a sequence, the state of your HMM that most probably generated the nucleotide in that position

eg. if you segmented a 1000-bp sequence using a HMM with "AT-rich" & "GC-rich" states, it may tell you:

bases 1-343 were probably generated by the AT-rich state, 344-900 by GC-rich state, 901-1000 by AT-rich state

- A **HMM** is a **probabilistic model**
- We can use a HMM to **segment a genome** into long stretches that are AT-rich or GC-rich
- We use a HMM with AT-rich & GC-rich states
We specify probabilities p_A, p_T, p_G, p_C for each state

We also specify the probability of a transition from the AT-rich to the GC-rich state, or from GC-rich to AT-rich state

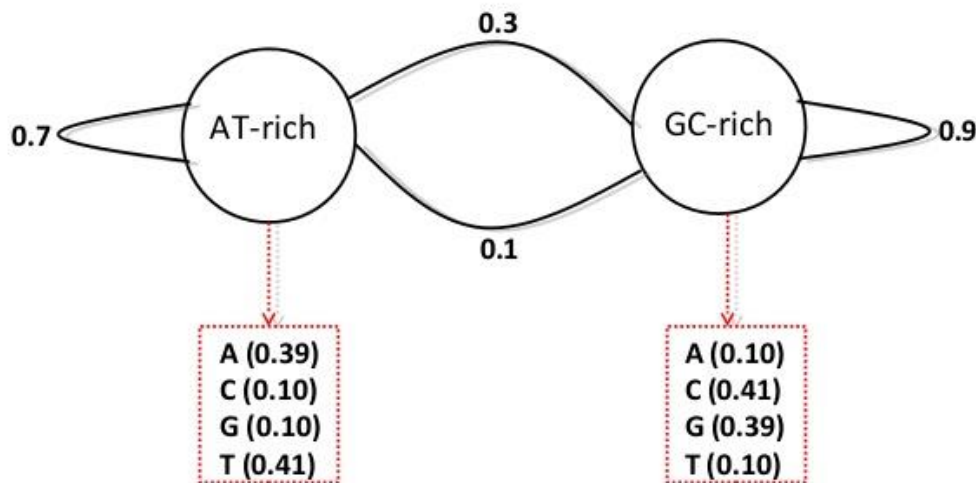


- We can display a HMM as a **state graph**

Each node represents a state in the HMM

Each edge represents a transition from one state to another

Each node has a certain set of emission probabilities, ie. probabilities p_A, p_C, p_G, p_T for that state

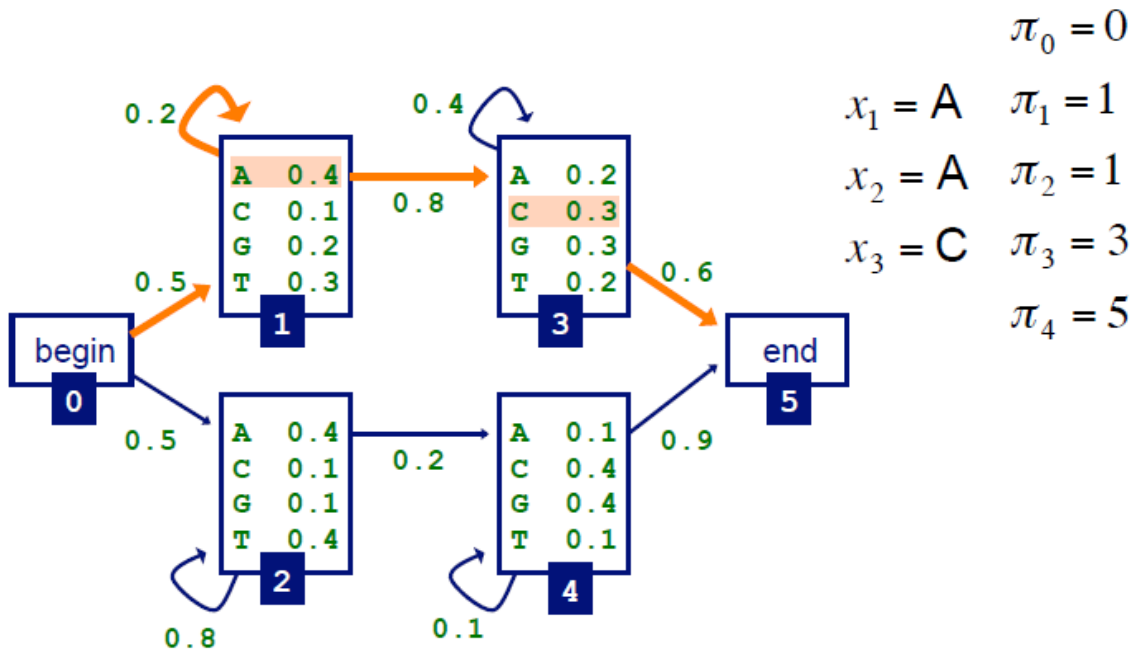


Three important questions

- How likely is a given sequence?
the **Forward algorithm**
- What is the most probable “path” for generating a given sequence?
the **Viterbi algorithm**
- How can we learn the HMM parameters given a set of sequences?
the **Forward-Backward (Baum-Welch) algorithm**

Path notation

- let π be a vector representing a path through the HMM



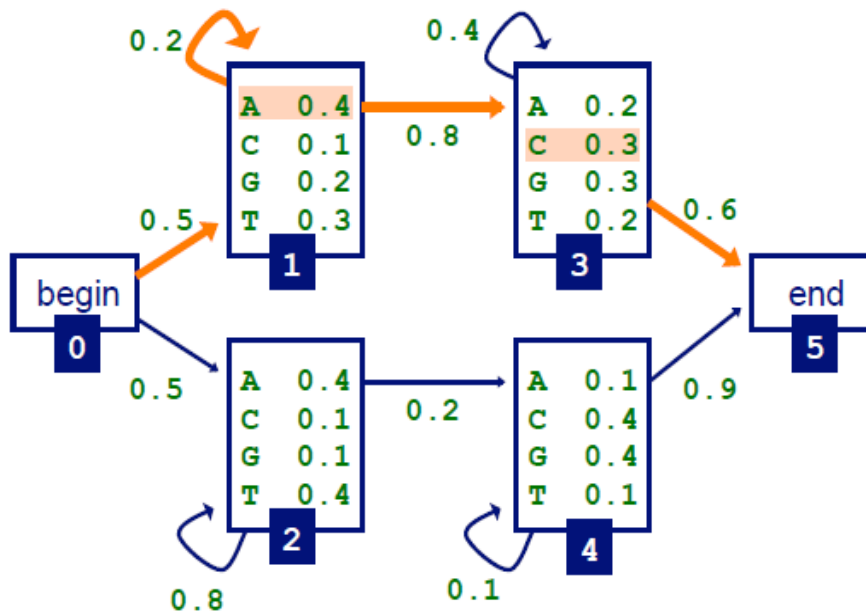
How likely is a given sequence?

- the probability that the path $\pi_0 \dots \pi_N$ is taken and the sequence $x_1 \dots x_L$ is generated:

$$P(x_1 \dots x_L, \pi_0 \dots \pi_N) = a_{0\pi_1} \prod_{i=1}^L e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}}$$

(assuming begin/end are the only silent states on path)

How likely is a given sequence?



$$\begin{aligned}
 P(\text{AAC}, \pi) &= a_{01} \times e_1(\text{A}) \times a_{11} \times e_1(\text{A}) \times a_{13} \times e_3(\text{C}) \times a_{35} \\
 &= 0.5 \times 0.4 \times 0.2 \times 0.4 \times 0.8 \times 0.3 \times 0.6
 \end{aligned}$$

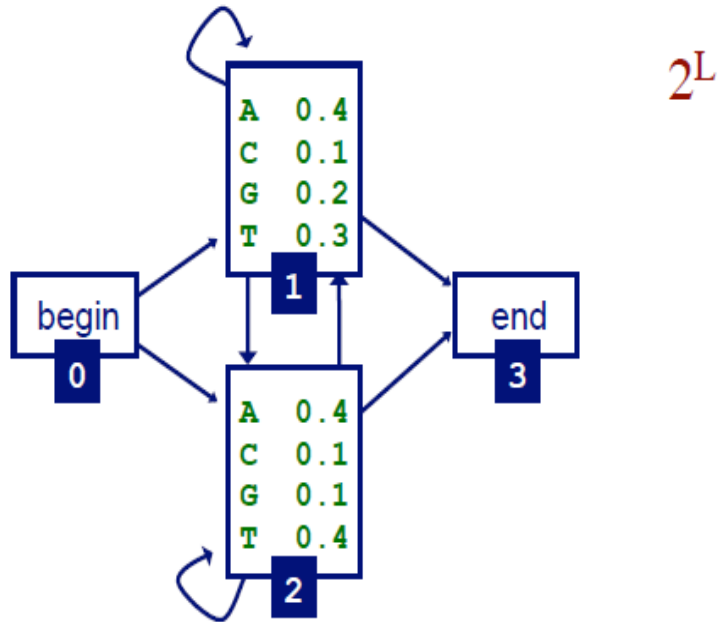
How likely is a given sequence?

- the probability over *all* paths is:

$$P(x_1 \dots x_L) = \sum_{\pi} P(x_1 \dots x_L, \underbrace{\pi_0 \dots \pi_N}_{\pi})$$

Number of paths

- for a sequence of length L , how many possible paths through this HMM are there?



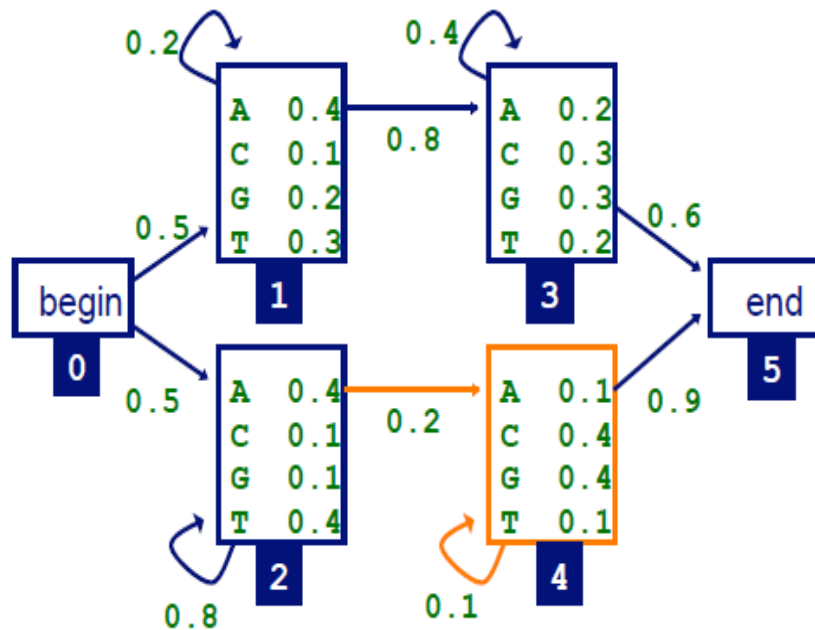
- the Forward algorithm enables us to compute $P(x_1 \dots x_L)$ efficiently

How likely is a given sequence: the Forward algorithm

- define $f_k(i)$ to be the probability of being in state k having observed the first i characters of x
- we want to compute $f_N(L)$, the probability of being in the end state having observed all of x
- can define this recursively

The Forward algorithm

- because of the Markov property, don't have to explicitly enumerate every path – use dynamic programming instead



- e.g. compute $f_4(i)$ using $f_2(i-1)$, $f_4(i-1)$

initialization:

$$f_0(0) = 1$$

probability that we're in start state and have observed 0 characters from the sequence

$$f_k(0) = 0, \quad \text{for } k \text{ that are not silent states}$$

recursion for emitting states ($i = 1 \dots L$):

$$f_l(i) = e_l(i) \sum_k f_k(i-1) a_{kl}$$

recursion for silent states:

$$f_l(i) = \sum_k f_k(i) a_{kl}$$

termination:

$$P(x) = P(x_1 \dots x_L) = f_N(L) = \sum_k f_k(L) a_{kN}$$

probability that we're in the end state and
have observed the entire sequence

Forward algorithm example

- given the sequence $x = \text{TAGA}$
- initialization

$$f_0(0) = 1 \quad f_1(0) = 0 \quad \dots \quad f_5(0) = 0$$

- computing other values

$$\begin{aligned} f_1(1) &= e_1(T) \times (f_0(0)a_{01} + f_1(0)a_{11}) = \\ &= 0.3 \times (1 \times 0.5 + 0 \times 0.2) = 0.15 \end{aligned}$$

$$f_2(1) = 0.4 \times (1 \times 0.5 + 0 \times 0.8)$$

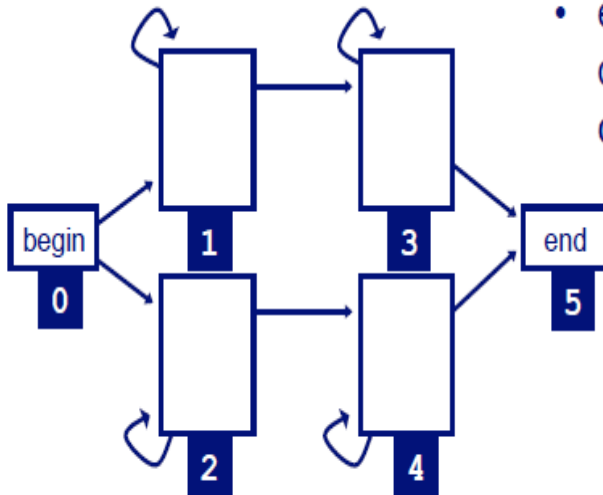
$$\begin{aligned} f_1(2) &= e_1(A) \times (f_0(1)a_{01} + f_1(1)a_{11}) = \\ &= 0.4 \times (0 \times 0.5 + 0.15 \times 0.2) \end{aligned}$$

• • •

$$P(\text{TAGA}) = f_5(4) = (f_3(4)a_{35} + f_4(4)a_{45})$$

Forward algorithm note

- in some cases, we can make the algorithm more efficient by taking into account the minimum number of steps that must be taken to reach a state



- e.g. for this HMM, we don't need to initialize or compute the values

$$f_3(0), f_4(0),$$

$$f_5(0), f_5(1)$$

Finding the most probable path: the Viterbi algorithm

- define $v_k(i)$ to be the probability of the most probable path accounting for the first i characters of x and ending in state k
- we want to compute $v_N(L)$, the probability of the most probable path accounting for all of the sequence and ending in the end state
- can define recursively, use DP to find $v_N(L)$ efficiently
- initialization:

$$v_0(0) = 1$$

$$v_k(0) = 0, \quad \text{for } k \text{ that are not silent states}$$

- recursion for emitting states ($i = 1 \dots L$):

$$v_l(i) = e_l(x_i) \max_k [v_k(i-1)a_{kl}]$$

$$\text{ptr}_l(i) = \arg \max_k [v_k(i-1)a_{kl}] \quad \text{keep track of most probable path}$$

- recursion for silent states:

$$v_l(i) = \max_k [v_k(i)a_{kl}]$$

$$\text{ptr}_l(i) = \arg \max_k [v_k(i)a_{kl}]$$

- termination:

$$P(x, \pi) = \max_k (v_k(L)a_{kN})$$

$$\pi_L = \arg \max_k (v_k(L)a_{kN})$$

- traceback: follow pointers back starting at π_L

HMM-APPLICATION

- DNA Sequence analysis
 - Protein family profiling
 - Predprediction
 - Splicing signals prediction
 - Prediction of genes
 - Horizontal gene transfer
 - Radiation hybrid mapping, linkage analysis
 - Prediction of DNA functional sites.
 - CpG island
- Prediction of protein-coding regions in genome sequences
 - Modelling families of related dna or protein sequences
 - Prediction of secondary structure elements from protein primary sequences

Module 5

General introduction to gene expression in eukaryotes and prokaryotes

Gene expression is a highly complex, regulated process that begins with DNA transcribed into RNA, which is then translated into protein.

- Every cell within an organism shares the same genome (with exceptions, i.e. mature red blood cells), but has variation between its proteomes.
- Gene expression involves the process of transcribing DNA into RNA and then translating RNA into proteins.

- Gene expression is a highly complex and tightly-regulated process

Prokaryotic versus Eukaryotic Gene Expression

Prokaryotes regulate gene expression by controlling the amount of transcription, whereas eukaryotic control is much more complex.

Key Points

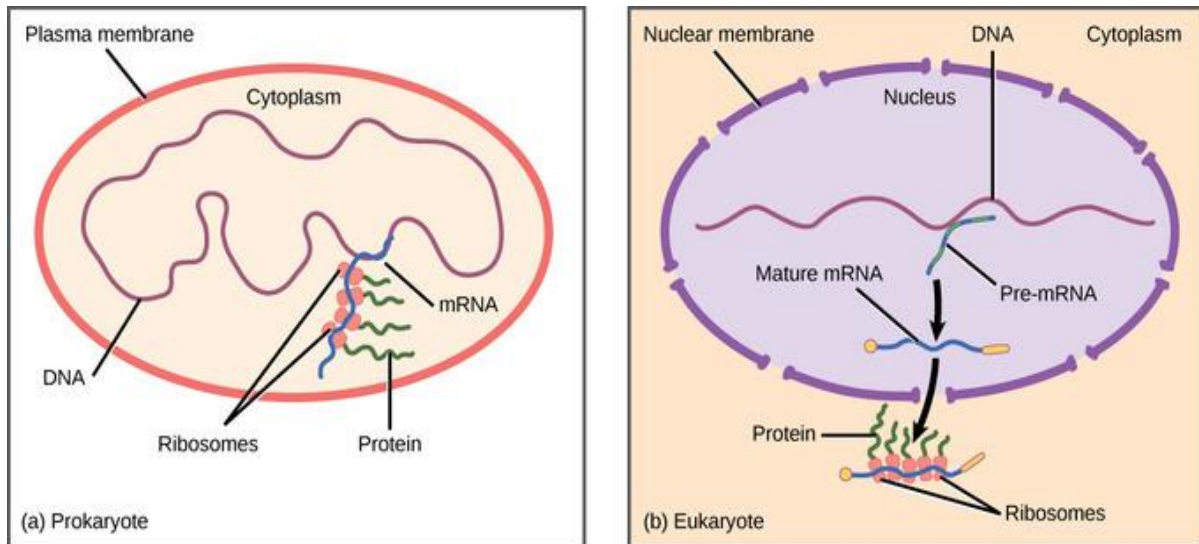
- Prokaryotic gene expression is primarily controlled at the level of transcription.
- Eukaryotic gene expression is controlled at the levels of epigenetics, transcription, post-transcription, translation, and post-translation.
- Prokaryotic gene expression (both transcription and translation) occurs within the cytoplasm of a cell due to the lack of a defined nucleus; thus, the DNA is freely located within the cytoplasm.
- Eukaryotic gene expression occurs in both the nucleus (transcription) and cytoplasm (translation).

To understand how gene expression is regulated, we must first understand how a gene codes for a functional protein in a cell. The process occurs in both prokaryotic and eukaryotic cells, just in slightly different manners.

Prokaryotic organisms are single-celled organisms that lack a defined nucleus; therefore, their DNA floats freely within the cell cytoplasm. To synthesize a protein, the processes of transcription (DNA to RNA) and translation (RNA to protein) occur almost simultaneously. When the resulting protein is no longer needed, transcription stops. Thus, the regulation of transcription is the primary method to control what type of protein and how much of each protein is expressed in a prokaryotic cell. All of the subsequent steps occur automatically. When more protein is required, more transcription occurs. Therefore, in prokaryotic cells, the control of gene expression is mostly at the transcriptional level.

Eukaryotic cells, in contrast, have intracellular organelles that add to their complexity. In eukaryotic cells, the DNA is contained inside the cell's nucleus where it is transcribed into RNA. The newly-synthesized RNA is then transported out of the nucleus into the cytoplasm where ribosomes translate the RNA into protein. The processes of transcription and translation are physically separated by the nuclear membrane; transcription occurs only within the nucleus, and translation occurs only outside the nucleus within the cytoplasm. The regulation of gene

expression can occur at all stages of the process. Regulation may occur when the DNA is uncoiled and loosened from nucleosomes to bind transcription factors (epigenetics), when the RNA is transcribed (transcriptional level), when the RNA is processed and exported to the cytoplasm after it is transcribed (post-transcriptional level), when the RNA is translated into protein (translational level), or after the protein has been made (post-translational level).

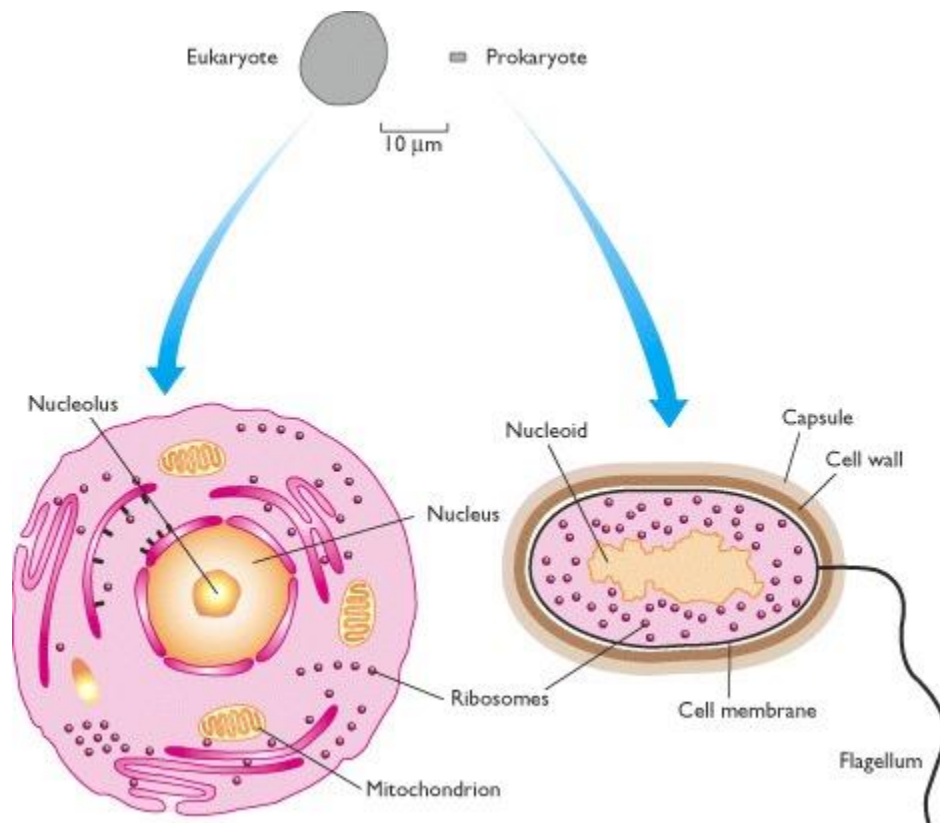


Prokaryotic vs Eukaryotic Gene Expression: Prokaryotic transcription and translation occur simultaneously in the cytoplasm, and regulation occurs at the transcriptional level. Eukaryotic gene expression is regulated during transcription and RNA processing, which take place in the nucleus, and during protein translation, which takes place in the cytoplasm. Further regulation may occur through post-translational modifications of proteins.

Prokaryotic Genomes :

Prokaryotes, whose cells lack extensive internal compartments. There are two very different groups of prokaryotes, distinguished from one another by characteristic genetic and biochemical features:

- a. the bacteria, which include most of the commonly encountered prokaryotes such as the gram-negatives (e.g. *E. coli*), the gram-positives (e.g. *Bacillus subtilis*), the cyanobacteria (e.g. *Anabaena*) and many more;
- b. the archaea, which are less well-studied, and have mostly been found in extreme environments such as hot springs, brine pools and anaerobic lake bottoms.



Cells of eukaryotes (left) and prokaryotes (right)

The top part of the figure shows a typical human cell and typical bacterium drawn to scale. The human cell is 10 μm in diameter and the bacterium is rod-shaped with dimensions of 1 × 2 μm. The lower drawings show the internal structures of eukaryotic and prokaryotic cells. Eukaryotic cells are characterized by their membrane-bound compartments, which are absent from prokaryotes. The bacterial DNA is contained in the structure called the nucleoid

- The genome of prokaryotic organisms generally is a circular, double-stranded piece of DNA, multiple copies of which may exist at any time.

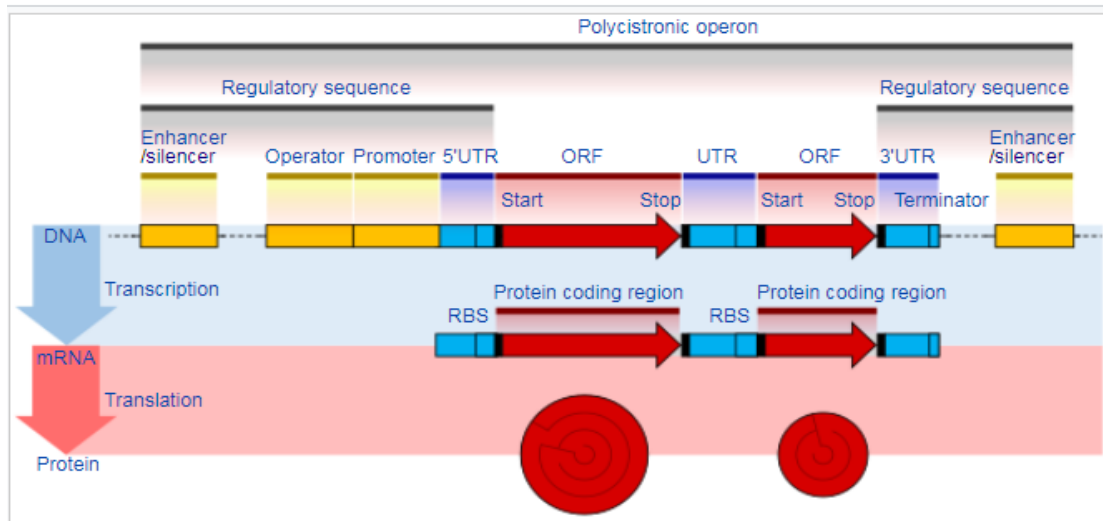
- The length of a genome varies widely, but is generally at least a few million base pairs.
- A genophore is the DNA of a prokaryote. It is commonly referred to as a prokaryotic chromosome

Gene Structure :

The overall organisation of prokaryotic genes is markedly different from that of the eukaryotes. The most obvious difference is that prokaryotic ORFs are often grouped into a polycistronic operon under the control of a shared set of regulatory sequences. These ORFs are all transcribed onto the same mRNA and so are co-regulated and often serve related functions. Each ORF typically has its own ribosome binding site (RBS) so that ribosomes simultaneously translate ORFs on the same mRNA. Some operons also display translational coupling, where the translation rates of multiple ORFs within an operon are linked. This can occur when the ribosome remains attached at the end of an ORF and simply translocates along to the next without the need for a new RBS.

Translational coupling is also observed when translation of an ORF affects the accessibility of the next RBS through changes in RNA secondary structure. Having multiple ORFs on a single mRNA is only possible in prokaryotes because their transcription and translation take place at the same time and in the same subcellular location.

The operator sequence next to the promoter is the main regulatory element in prokaryotes. Repressor proteins bound to the operator sequence physically obstructs the RNA polymerase enzyme, preventing transcription. Riboswitches are another important regulatory sequence commonly present in prokaryotic UTRs. These sequences switch between alternative secondary structures in the RNA depending on the concentration of key metabolites. The secondary structures then either block or reveal important sequence regions such as RBSs. Introns are extremely rare in prokaryotes and therefore do not play a significant role in prokaryotic gene regulation



The structure of a prokaryotic operon of protein-coding genes. Regulatory sequence controls when expression occurs for the multiple protein coding regions (red). Promoter, operator and enhancer regions (yellow) regulate the transcription of the gene into an mRNA. The mRNA untranslated regions (blue) regulate translation into the final protein products

Gene Density :

In genetics, the **gene density** of an organism's genome is the ratio of the number of genes per number of base pairs, usually written in terms of a million base pairs, or *megabase* (Mb).

Seemingly simple organisms, such as bacteria and amoebas, have a much higher gene density than humans. Bacterial DNA has a gene density on the order of 500-1000 genes/Mb. This is due several factors, including that the fact that bacterial DNA has no introns. There are also fewer codons in bacterial genes

GC content :

The GC content (percentage) is the number of GC nucleotides divided by the total nucleotides. However, it has been shown that there is a strong correlation between the prokaryotic optimal growth at higher temperatures and the GC content of structured RNAs (such as ribosomal RNA, transfer RNA, and many other non-coding RNAs)

Eukaryotic Genomes :

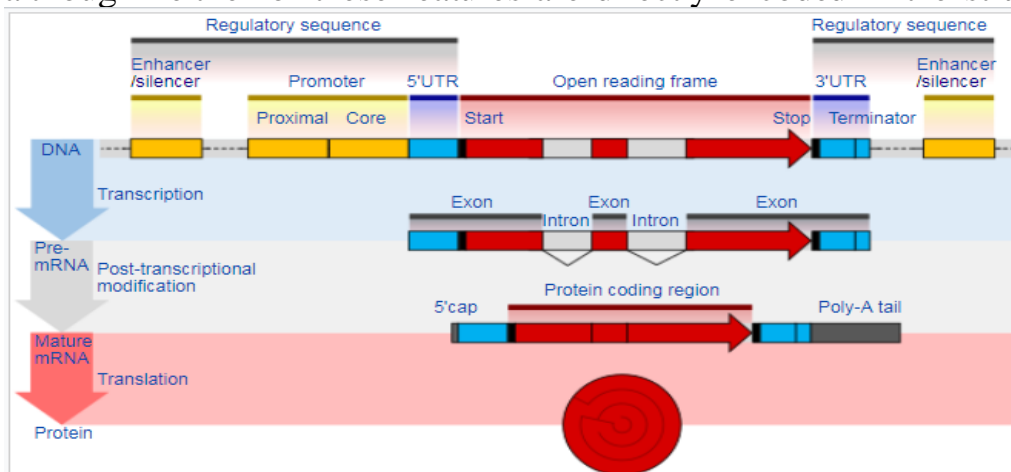
Eukaryotic genomes are composed of one or more linear DNA chromosomes. The number of chromosomes varies widely from Jack jumper ants and an asexual nemotode, which each have only one pair, to a fern species that has 720 pairs. A typical human cell has two copies of each of 22 autosomes, one inherited from

each parent, plus two sex chromosomes, making it diploid. Gametes, such as ova, sperm, spores, and pollen, are haploid, meaning they carry only one copy of each chromosome.

In addition to the chromosomes in the nucleus, organelles such as the chloroplasts and mitochondria have their own DNA. Mitochondria are sometimes said to have their own genome often referred to as the "mitochondrial genome". The DNA found within the chloroplast may be referred to as the "plastome". Like the bacteria they originated from, mitochondria and chloroplasts have a circular chromosome. Unlike prokaryotes, eukaryotes have exon-intron organization of protein coding genes and variable amounts of repetitive DNA. In mammals and plants, the majority of the genome is composed of repetitive DNA.

Gene Structure :

The structure of eukaryotic genes includes features not found in prokaryotes. Most of these relate to post-transcriptional modification of pre-mRNAs to produce mature mRNA ready for translation into protein. Eukaryotic genes typically have more regulatory elements to control gene expression compared to prokaryotes. This is particularly true in multicellular eukaryotes, humans for example, where gene expression varies widely among different tissues. A key feature of the structure of eukaryotic genes is that their transcripts are typically subdivided into exon and intron regions. Exon regions are retained in the final mature mRNA molecule, while intron regions are spliced out (excised) during post-transcriptional processing. Indeed, the intron regions of a gene can be considerably longer than the exon regions. Once spliced together, the exons form a single continuous protein-coding regions, and the splice boundaries are not detectable. Eukaryotic post-transcriptional processing also adds a 5' cap to the start of the mRNA and a poly-adenosine tail to the end of the mRNA. These additions stabilise the mRNA and direct its transport from the nucleus to the cytoplasm, although neither of these features are directly encoded in the structure of a gene



GC content :

The total GC content of the genome does not have the same variability among eukaryotic species, so as in prokaryotes

However, it seems to play a very significant role in gene recognition algorithms, because:

- eukaryotic ORFs are much more difficult to recognize
- the large-scale variation of GC content within eukaryotic genomes is the basis for useful correlations between genes and upstream promoter sequences, for the choice of codons, the length of genes and their density

Qualitatively, G (guanine) and C (cytosine) undergo a specific [hydrogen bonding](#), whereas A (adenine) bonds specifically with T (thymine, in DNA) or U (uracil, in RNA). Quantitatively, the GC pair is bound by three [hydrogen bonds](#), while AT and AU pairs are bound by two hydrogen bonds. To emphasize this difference in the number of hydrogen bonds, the base pairings can be represented as respectively $G\equiv C$ versus $A=T$ and $A=U$. DNA with low GC-content is less stable than DNA with high GC-content; however, the hydrogen bonds themselves do not have a particularly significant impact on stabilization, the stabilization is due mainly to interactions of base stacking. In spite of the higher [thermostability](#) conferred to the genetic material, it has been observed that at least some bacteria species with DNA of high GC-content undergo [autolysis](#) more readily, thereby reducing the longevity of the cell *per se*. Due to the thermostability given to the genetic materials in high GC organisms, it was commonly believed that the GC content played a necessary role in adaptation temperatures. However, it has been shown that there is a strong correlation between the prokaryotic optimal growth at higher temperatures and the GC content of structured RNAs (such as [ribosomal RNA](#), [transfer RNA](#), and many other [non-coding RNAs](#)). The AU base pairs are less stable than the GC [base pairs](#) previously attributed to GC bonds containing 3 hydrogen bonds and AU having only 2 hydrogen bonds, making high-GC-content RNA structures more resistant to the effects of high temperatures. More recently, it has been proved that the most stabilizing factor of thermal stability of double stranded nucleic acids is actually due to the base stackings of adjacent bases, rather than the number of hydrogen bonds between the bases. There is more favorable stacking energy for

G:C pairs because of the relative positions of exocyclic groups than in the A:U pairs. Additionally, there is a correlation between the order in which the bases stack and thermal stability.

Gene Density:

In genetics, the **gene density** of an organism's genome is the ratio of the number of genes per number of base pairs, usually written in terms of a million base pairs, or megabase (Mb). The human genome has a gene density of 12-15 genes/Mb,

Genes are far from each other, even in those regions of complex eukaryotes that are particularly rich of coding information.

- The average distance between human genes is around 65,000 base pairs, approximately equal to 10% of the genome size of a simple prokaryotic organism

Moreover:

- Mutational analyses have revealed that many genes encode proteins that perform multiple functions
- Many genes are present in multiple, redundant copies
- Simple eukaryotes tend to have a higher density of genes compared to more complex organisms, such as vertebrates
- Humans and other mammals have lowest gene density (# genes), in a given length of DNA
- Multicellular eukaryotes have many introns within genes and a large amount of noncoding DNA between genes

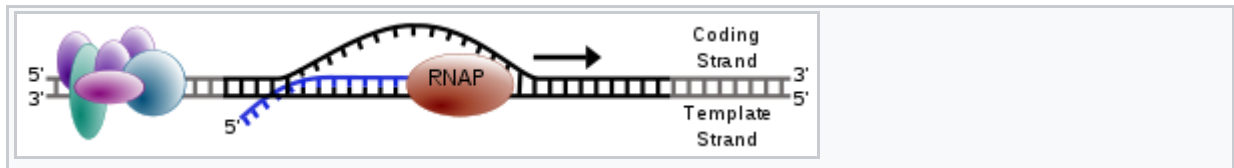
Gene Expression :

Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product. These products are often proteins, but in non-protein coding genes such as transfer RNA (tRNA) or small nuclear RNA (snRNA) genes, the product is a functional RNA.

The process of gene expression is used by all known life—eukaryotes

MECHANISMS :

Transcription



The process of transcription is carried out by RNA polymerase (RNAP), which uses DNA (black) as a template and produces RNA (blue).

A gene is a stretch of DNA that encodes information. Genomic DNA consists of two antiparallel and reverse complementary strands, each having 5' and 3' ends. With respect to a gene, the two strands may be labeled the "template strand," which serves as a blueprint for the production of an RNA transcript, and the "coding strand," which includes the DNA version of the transcript sequence. (Perhaps surprisingly, the "coding strand" is not physically involved in the coding process because it is the "template strand" that is read during transcription.)

The production of the RNA copy of the DNA is called transcription, and is performed in the nucleus by RNA polymerase, which adds one RNA nucleotide at a time to a growing RNA strand as per the complementarity law of the bases. This RNA is complementary to the template 3' → 5' DNA strand, which is itself complementary to the coding 5' → 3' DNA strand. Therefore, the resulting 5' → 3' RNA strand is identical to the coding DNA strand with the exception that Thymines are replaced with uracils (U) in the RNA. A coding DNA strand reading "ATG" is indirectly transcribed through the non-coding strand as "UAC" in RNA.

In prokaryotes, transcription is carried out by a single type of RNA polymerase, which needs a DNA sequence called a Pribnow box as well as a sigma factor (σ)

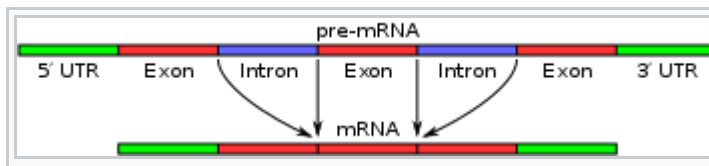
factor) to start transcription. In eukaryotes, transcription is performed by three types of RNA polymerases, each of which needs a special DNA sequence called the promoter and a set of DNA-binding proteins—transcription factors—to initiate the process. RNA polymerase I is responsible for transcription of ribosomal RNA (rRNA) genes. RNA polymerase II (Pol II) transcribes all protein-coding genes but also some non-coding RNAs (e.g., snRNAs, snoRNAs or long non-coding RNAs). Pol II includes a C-terminal domain (CTD) that is rich in serine residues. When these residues are phosphorylated, the CTD binds to various protein factors that promote transcript maturation and modification. RNA polymerase III transcribes 5S rRNA, transfer RNA (tRNA) genes, and some small non-coding RNAs (e.g., 7SK). Transcription ends when the polymerase encounters a sequence called the terminator.

RNA processing

While transcription of prokaryotic protein-coding genes creates messenger RNA (mRNA) that is ready for translation into protein, transcription of eukaryotic genes leaves a primary transcript of RNA (pre-mRNA), which first has to undergo a series of modifications to become a mature mRNA.

These include 5' capping, which is set of enzymatic reactions that add 7-methylguanosine (m^7G) to the 5' end of pre-mRNA and thus protect the RNA from degradation by exonucleases. The m^7G cap is then bound by cap binding complex heterodimer (CBC20/CBC80), which aids in mRNA export to cytoplasm and also protect the RNA from decapping.

Another modification is 3' cleavage and polyadenylation. They occur if polyadenylation signal sequence (5'- AAUAAA-3') is present in pre-mRNA, which is usually between protein-coding sequence and terminator. The pre-mRNA is first cleaved and then a series of ~200 adenines (A) are added to form poly(A) tail, which protects the RNA from degradation. Poly(A) tail is bound by multiple poly(A)-binding proteins (PABP) necessary for mRNA export and translation re-initiation.



Simple illustration of exons and introns in pre-mRNA and the formation of mature mRNA by splicing. The UTRs are non-coding parts of exons at the ends of the mRNA.

A very important modification of eukaryotic pre-mRNA is RNA splicing. The majority of eukaryotic pre-mRNAs consist of alternating segments called exons and introns. During the process of splicing, an RNA-protein catalytical complex known as spliceosome catalyzes two transesterification reactions, which remove an intron and release it in form of lariat structure, and then splice neighbouring exons together. In certain cases, some introns or exons can be either removed or retained in mature mRNA. This so-called alternative splicing creates series of different transcripts originating from a single gene. Because these transcripts can be potentially translated into different proteins, splicing extends the complexity of eukaryotic gene expression.

Extensive RNA processing may be an evolutionary advantage made possible by the nucleus of eukaryotes. In prokaryotes, transcription and translation happen together, whilst in eukaryotes, the nuclear membrane separates the two processes, giving time for RNA processing to occur.

Non-coding RNA maturation

In most organisms non-coding genes (ncRNA) are transcribed as precursors that undergo further processing. In the case of ribosomal RNAs (rRNA), they are often transcribed as a pre-rRNA that contains one or more rRNAs. The pre-rRNA is cleaved and modified (2'-O-methylation and pseudouridine formation) at specific sites by approximately 150 different small nucleolus-restricted RNA species, called snoRNAs. SnoRNAs associate with proteins, forming snoRNPs. While snoRNA part basepair with the target RNA and thus position the modification at a precise site, the protein part performs the catalytical reaction. In eukaryotes, in particular a snoRNP called RNase, MRP cleaves the 45S pre-rRNA into the 28S, 5.8S, and 18S rRNAs. The rRNA and RNA processing factors form large aggregates called the nucleolus.

In the case of transfer RNA (tRNA), for example, the 5' sequence is removed by RNase P, whereas the 3' end is removed by the tRNase Z enzyme and the non-templated 3' CCA tail is added by a nucleotidyl transferase. In the case of micro RNA (miRNA), miRNAs are first transcribed as primary transcripts or pri-miRNA with a cap and poly-A tail and processed to short, 70-nucleotide stem-loop structures known as pre-miRNA in the cell nucleus by the enzymes Drosha and Pasha. After being exported, it is then processed to mature miRNAs in the cytoplasm by interaction with the endonuclease Dicer, which also initiates the formation of the RNA-induced silencing complex (RISC), composed of the Argonaute protein.

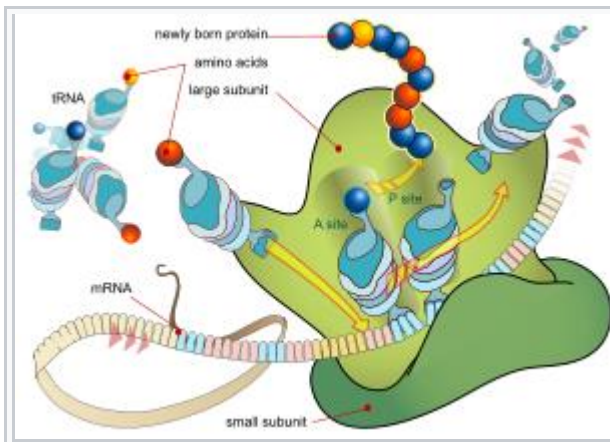
Even snRNAs and snoRNAs themselves undergo series of modification before they become part of functional RNP complex. This is done either in the

nucleoplasm or in the specialized compartments called Cajal bodies. Their bases are methylated or pseudouridinated by a group of small Cajal body-specific RNAs (scaRNAs), which are structurally similar to snoRNAs.

RNA export

In eukaryotes most mature RNA must be exported to the cytoplasm from the nucleus. While some RNAs function in the nucleus, many RNAs are transported through the nuclear pores and into the cytosol. Notably this includes all RNA types involved in protein synthesis. In some cases RNAs are additionally transported to a specific part of the cytoplasm, such as a synapse; they are then towed by motor proteins that bind through linker proteins to specific sequences (called "zipcodes") on the RNA.

Translation



During the translation, tRNA charged with amino acid enters the ribosome and aligns with the correct mRNA triplet. Ribosome then adds amino acid to growing protein chain.

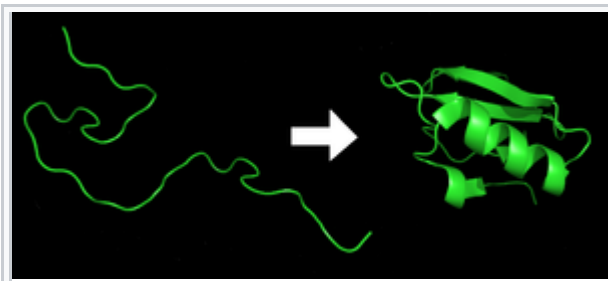
For some RNA (non-coding RNA) the mature RNA is the final gene product. In the case of messenger RNA (mRNA) the RNA is an information carrier coding for the

synthesis of one or more proteins. mRNA carrying a single protein sequence (common in eukaryotes) is monocistronic whilst mRNA carrying multiple protein sequences (common in prokaryotes) is known as polycistronic.

Every mRNA consists of three parts: a 5' untranslated region (5'UTR), a protein-coding region or open reading frame (ORF), and a 3' untranslated region (3'UTR). The coding region carries information for protein synthesis encoded by the genetic code to form triplets. Each triplet of nucleotides of the coding region is called a codon and corresponds to a binding site complementary to an anticodon triplet in transfer RNA. Transfer RNAs with the same anticodon sequence always carry an identical type of amino acid. Amino acids are then chained together by the ribosome according to the order of triplets in the coding region. The ribosome helps transfer RNA to bind to messenger RNA and takes the amino acid from each transfer RNA and makes a structure-less protein out of it. Each mRNA molecule is translated into many protein molecules, on average ~2800 in mammals.

In prokaryotes translation generally occurs at the point of transcription (co-transcriptionally), often using a messenger RNA that is still in the process of being created. In eukaryotes translation can occur in a variety of regions of the cell depending on where the protein being written is supposed to be. Major locations are the cytoplasm for soluble cytoplasmic proteins and the membrane of the endoplasmic reticulum for proteins that are for export from the cell or insertion into a cell membrane. Proteins that are supposed to be expressed at the endoplasmic reticulum are recognised part-way through the translation process. This is governed by the signal recognition particle—a protein that binds to the ribosome and directs it to the endoplasmic reticulum when it finds a signal peptide on the growing (nascent) amino acid chain.

Folding



Protein before (left) and after (right) folding

The polypeptide folds into its characteristic and functional three-dimensional structure from a random coil. Each protein exists as an unfolded polypeptide or

random coil when translated from a sequence of mRNA into a linear chain of amino acids. This polypeptide lacks any developed three-dimensional structure (the left hand side of the neighboring figure). Amino acids interact with each other to produce a well-defined three-dimensional structure, the folded protein (the right hand side of the figure) known as the native state. The resulting three-dimensional structure is determined by the amino acid sequence (Anfinsen's dogma).

The correct three-dimensional structure is essential to function, although some parts of functional proteins may remain unfolded. Failure to fold into the intended shape usually produces inactive proteins with different properties including toxic prions. Several neurodegenerative and other diseases are believed to result from the accumulation of *misfolded* proteins. Many allergies are caused by the folding of the proteins, for the immune system does not produce antibodies for certain protein structures.]

Enzymes called chaperones assist the newly formed protein to attain (fold into) the 3-dimensional structure it needs to function. Similarly, RNA chaperones help RNAs attain their functional shapes. Assisting protein folding is one of the main roles of the endoplasmic reticulum in eukaryotes.

Translocation

Secretory proteins of eukaryotes or prokaryotes must be translocated to enter the secretory pathway. Newly synthesized proteins are directed to the eukaryotic Sec61 or prokaryotic SecYEG translocation channel by signal peptides. The efficiency of protein secretion in eukaryotes is very dependent on the signal peptide which has been used.

Protein transport

Many proteins are destined for other parts of the cell than the cytosol and a wide range of signalling sequences or (signal peptides) are used to direct proteins to where they are supposed to be. In prokaryotes this is normally a simple process due to limited compartmentalisation of the cell. However, in eukaryotes there is a great variety of different targeting processes to ensure the protein arrives at the correct organelle.

Not all proteins remain within the cell and many are exported, for example, digestive enzymes, hormones and extracellular matrix proteins. In eukaryotes the export pathway is well developed and the main mechanism for the export of these proteins is translocation to the endoplasmic reticulum, followed by transport via the Golgi apparatus.

Transposition

A **transposable element** (**TE** or **transposon**) is a DNA sequence that can change its position within a genome, sometimes creating or reversing mutations and altering the cell's genetic identity and genome size. Transposition often results in duplication of the same genetic material.

Transposable elements make up a large fraction of the genome and are responsible for much of the mass of DNA in a eukaryotic cell. It has been shown that TEs are important in genome function and evolution. In *Oxytricha*, which has a unique genetic system, these elements play a critical role in development. Transposons are also very useful to researchers as a means to alter DNA inside a living organism.

There are at least two classes of TEs: Class I TEs or retrotransposons generally function via reverse transcription, while Class II TEs or DNA transposons encode the protein transposase, which they require for insertion and excision, and some of these TEs also encode other proteins

Transposable elements represent one of several types of mobile genetic elements. TEs are assigned to one of two classes according to their mechanism of transposition, which can be described as either *copy and paste* (Class I TEs) or *cut and paste* (Class II TEs).

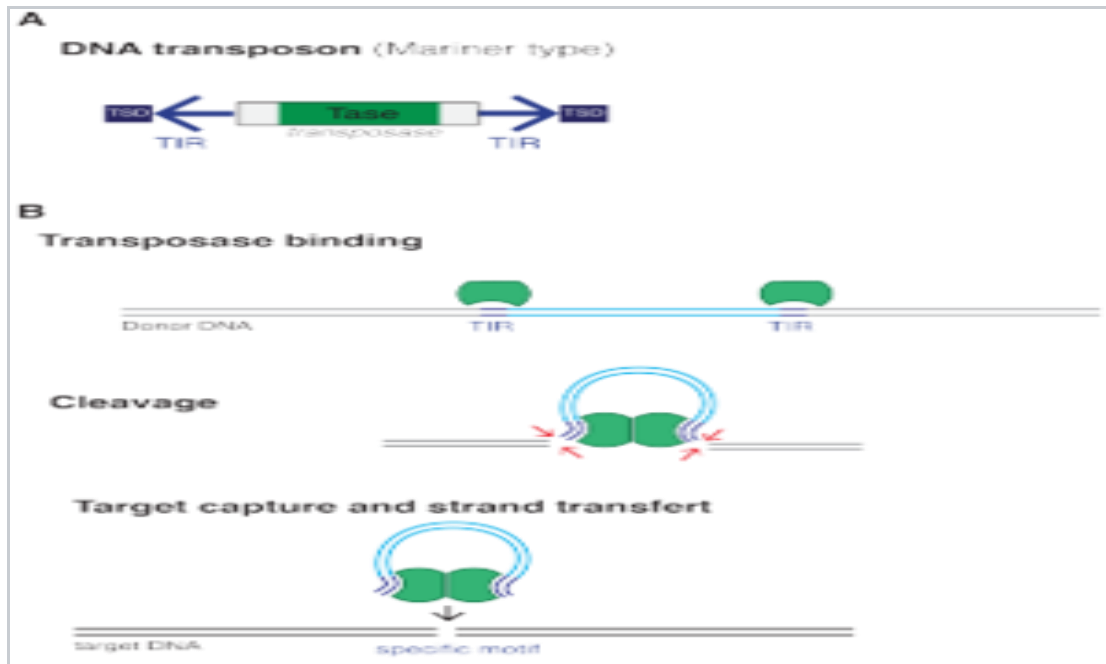
Class I (retrotransposons)

Class I TEs are copied in two stages: first, they are transcribed from DNA to RNA, and the RNA produced is then reverse transcribed to DNA. This copied DNA is then inserted back into the genome at a new position. The reverse transcription step is catalyzed by a reverse transcriptase, which is often encoded by the TE itself. The characteristics of retrotransposons are similar to retroviruses, such as HIV.

Retrotransposons are commonly grouped into three main orders:

- Retrotransposons, with long terminal repeats (LTRs), which encode reverse transcriptase, similar to retroviruses
- Retroposons, Long interspersed nuclear elements (LINEs, LINE-1s, or L1s), which encode reverse transcriptase but lack LTRs, and are transcribed by RNA polymerase II
- Short interspersed nuclear elements (SINEs) do not encode reverse transcriptase and are transcribed by RNA polymerase III

[Note : Retroviruses can also be considered TEs. For example, after conversion of retroviral RNA into DNA inside a host cell, the newly produced retroviral DNA is integrated into the genome of the host cell. These integrated DNAs are termed proviruses. The provirus is a specialized form of eukaryotic retrotransposon, which can produce RNA intermediates that may leave the host cell and infect other cells. The transposition cycle of retroviruses has similarities to that of prokaryotic TEs, suggesting a distant relationship between the two].



A. Structure of DNA transposons (Mariner type). Two inverted tandem repeats (TIR) flank the transposase gene. Two short tandem site duplications (TSD) are present on both sides of the insert. **B.** Mechanism of transposition: Two transposases recognize and bind to TIR sequences, join together and promote DNA double-strand cleavage. The DNA-transposase complex then inserts its DNA cargo at specific DNA motifs elsewhere in the genome, creating short TSDs upon integration

Class II (DNA transposons)

The cut-and-paste transposition mechanism of class II TEs does not involve an RNA intermediate. The transpositions are catalyzed by several transposase enzymes. Some transposases non-specifically bind to any target site in DNA, whereas others bind to specific target sequences. The transposase makes a staggered cut at the target site producing sticky ends, cuts out the DNA transposon and ligates it into the target site. A DNA polymerase fills in

the resulting gaps from the sticky ends and DNA ligase closes the sugar-phosphate backbone. This results in target site duplication and the insertion sites of DNA transposons may be identified by short direct repeats (a staggered cut in the target DNA filled by DNA polymerase) followed by inverted repeats (which are important for the TE excision by transposase).

Cut-and-paste TEs may be duplicated if their transposition takes place during S phase of the cell cycle, when a donor site has already been replicated but a target site has not yet been replicated. Such duplications at the target site can result in gene duplication, which plays an important role in genomic evolution.

Not all DNA transposons transpose through the cut-and-paste mechanism. In some cases, a replicative transposition is observed in which a transposon replicates itself to a new target site (e.g. helitron).

Class II TEs comprise less than 2% of the human genome, making the rest Class I.

Autonomous and non-autonomous

Transposition can be classified as either "autonomous" or "non-autonomous" in both Class I and Class II TEs. Autonomous TEs can move by themselves, whereas non-autonomous TEs require the presence of another TE to move. This is often because dependent TEs lack transposase (for Class II) or reverse transcriptase (for Class I).

Activator element (*Ac*) is an example of an autonomous TE, and dissociation elements (*Ds*) is an example of a non-autonomous TE. Without *Ac*, *Ds* is not able to transpose.

Gene Prediction Approaches:

A. Statistical or ab initio methods: These methods attempt to predict genes based on statistical properties of the given DNA sequence. Programs are e.g. Genscan, GeneID, GENIE and FGENEH.

B. Comparative methods: The given DNA string is compared with a similar DNA string from a different species at the appropriate evolutionary distance and genes are predicted in both sequences based on the assumption that exons will be well conserved, whereas introns will not. Programs are e.g. CEM (conserved exon method) and Twinscan.

C. Homology methods: The given DNA sequence is compared with known protein structures. Programs are e.g. TBLASTN or TBLASTX, Procrustes and GeneWise.

- **Ab Initio based-**

It joins the exons in correct order. Two signals->

- a) **Gene signals:** a small pattern within the genomic DNA including putative splice sites, start and stop sites of transcription or translation, branch points, transcription factor binding sites, recognizable consensus sequences.
- b) **Gene content:** a region of genomic DNA including nucleotide and amino acid distribution, Synonymous codon usage and **hexamer frequencies**.

- **Neural network based algorithm**

- Composed of network of mathematical variables.
- Multiple layers like input, output and hidden layers.
- GRAIL** (Splice junctions, start and stop codons, poly-A sites, promoters and CpG islands). It scans the query sequence with windows of variable lengths & scores.

- **Discriminant analysis**

- Linear Discriminant Analysis (LDA)** represents 2D graph of coding signals vs. all possible 3' splice site positions; a diagonal line.
- Quadratic Discriminant Analysis (QDA)** represents quadratic function; a curved line.
- FGENES** (LDA)

-**FGENESH [Find Genes]** (HMMs)

-**FGENESH_C** (Similarity based)

-**FGENESH+** (Combination of ab initio & similarity based)

-**MZEF [Michael Zhang's Exon Finder]** (QDA)

- **HMMs**

-**GENSCAN** (Fifth order HMMs); combination of hexamer frequencies with coding signals; probability score $P > 0.5$

-**HMMgene** (*Conditional Maximum Likelihood*); combination of ab initio & homology-based algorithm

▪ Homology-based-

Exon structures and sequences of related species are highly conserved.

Comparison of homologous sequences derived from cDNA or Expressed Sequence Tags (ESTs).

-**GenomeScan** (Combination of GENSCAN prediction results with BLASTX similarity searches)

-**EST2Genome** (Intron-exon boundaries); Comparison of an EST sequence with a genomic DNA sequence

-**SGP-1 [Syntenic Gene Prediction]** (Similar to EST2)

-**TwinScan** (gene-finding server; similar to GenomeScan)

▪ Consensus-based-

Combination of results of multiple programs based on consensus.

Improvement of specificity by correcting false positives & problem of overprediction.

Lowered sensitivity & missed predictions.

-**GeneComber** (Combination of HMMgene & GenScan prediction results)

-**DIGIT** (Combination of FGENESH, GENSCAN & HMMgene)

MODULE 6

Energy Minimisation method : ZUKER ALGORITHM

Zuker divides a secondary structure into elements that can be described as graphs, sometimes called loops, and assigns free energy based on the face of those graphs [ZS81]. The structure with the lowest free energy is the optimal structure. Figure 1.1 shows different types of loops.

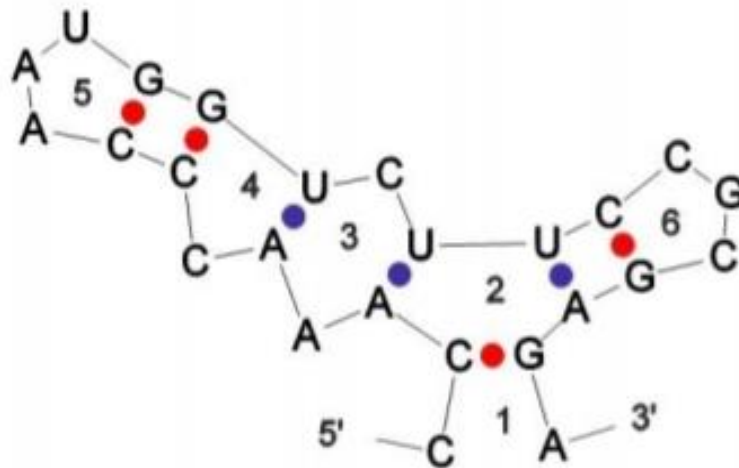


Figure 1.1 Zuker classifies a structure into five types of loops: hairpin loop (5, 6), stack region (consecutive base pairs between 5 and 4, or between 2 and 6), bulge loop (4), interior loop (3) and bifurcation loop (2).

Zuker defines two matrices $W(i,j)$ and $V(i,j)$ [ZS81]. $W(i,j)$ is the total free energy of subsequence i to j . $V(i,j)$ is also defined as the total free energy of subsequence i to j if i and j pairs, otherwise, $V(i,j) = \infty$. The recursion relations for $V(i,j)$ is defined as:

$$V(i,j) = \min \begin{cases} FH(i,j) \\ \min[FL(i,j,h,k) + V(h,k)] \text{ where } i < h < k < j \\ \min[W(i+1,k) + W(k+1,j-1)] \text{ where } i+1 < k < j-1 \end{cases}$$

Here, $FH(i,j)$ is the energy of hairpin loop $i \dots j$. $FL(i,j,h,k)$ is the energy of 2nd order loop such as stack region, bulge loop and interior loop $i \dots h \dots k \dots j$. The last item is the energy for bifurcation loop. The last item repeats over $i+1 < k < j-1$ because i and j must be a base pair, otherwise $V(i,j) = \infty$

The recursion relation for $W(i,j)$ is:

$$W(i,j) = \min \begin{cases} W(i+1, j) \\ W(i, j-1) \\ V(i, j) \\ \min[W(i,k) + W(k+1, j)] \text{ where } i < k < j \end{cases}$$

$W(1,L)$ gives the final total minimum free energy. The algorithm is $O(L^4)$ in time. The traceback stage in the Zuker algorithm is similar to the traceback stage in the Nussinov algorithm. The Zuker algorithm does not deal with pseudoknots. Figure 1.2 is a comparison of the secondary structures of the same sequence, folded by the Zuker algorithm and the Nussinov algorithm.

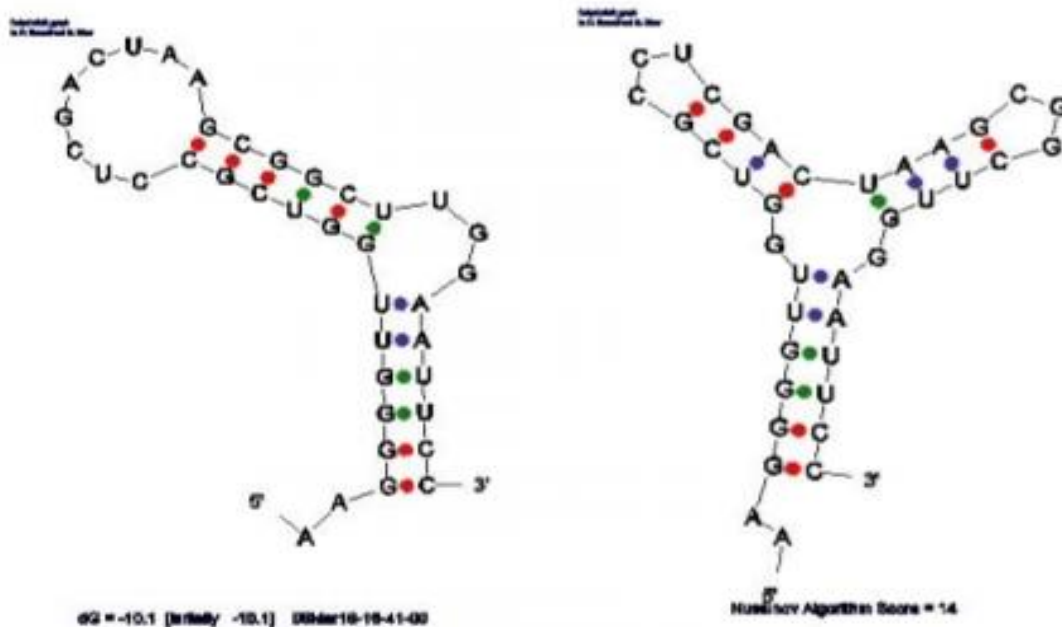
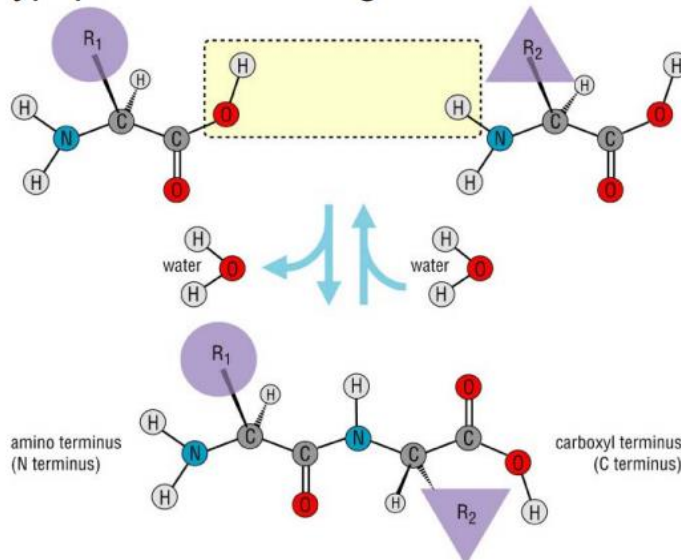


Figure 1.2 A short sequence folded by the Zuker algorithm (left) and the Nussinov algorithm (right). Hairpin loops with length less than 3 nucleotides (sharp U-turn) are not allowed. Note that there are less base pairs in the structure folded by Zuker algorithm than the structure folded by the Nussinov algorithm. Also note that the two algorithms [ZMT99] predict the same stack region near 5' and 3', but significantly different for the rest. The Nussinov algorithm obviously favors more base pairs, while the Zuker algorithm favors long stack region.

One of the problems with the Zuker algorithm is that the complex energy functions are not accurate enough. Therefore, a minimum free energy "optimal structure" can not be considered better than those structures with very close free energies [ZMT98]. For this reason, Zuker's MFOLD package calculates the optimal structure as well as the suboptimal structures within a user defined percentage.

The Peptide Bond

To make a protein, these amino acids are joined together in a polypeptide chain through the formation of a peptide bond.



Polypeptides

- Proteins are nothing more than long polypeptide chains.
- Chains that are less than 40-50 amino acids or **residues** are often referred to as polypeptide chains since they are too small to form a functional domain.
- Larger than this size, they are called proteins

Structure Prediction

INTRODUCTION :

- Protein three-dimensional structures are obtained using two popular experimental techniques, x-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy.
- There are many important proteins for which the sequence information is available, but their three-dimensional structures remain unknown.
- Therefore, it is often necessary to obtain approximate protein structures through computer modeling.

- Having a computer-generated three-dimensional model of a protein of interest has many ramifications, assuming it is reasonably correct.
- It may be of use for the rational design of biochemical experiments, such as site-directed mutagenesis, protein stability, or functional analysis.
- There are three computational approaches to protein three-dimensional structural modeling and prediction.

- They are homology modeling, threading, and ab initio prediction.
- The first two are knowledge-based methods; they predict protein structures based on knowledge of existing protein structural information in databases.
- The ab initio approach is simulation based and predicts structures based on physicochemical principles governing protein folding without the use of structural templates.

HOMOLOGY MODELLING :

- As the name suggests, homology modeling predicts protein structures based on sequence homology with known structures.
- It is also known as comparative modeling.
- The principle behind it is that if two proteins share a high enough sequence similarity, they are likely to have very similar three-dimensional structures.
- If one of the protein sequences has a known structure, then the structure can be copied to the unknown protein with a high degree of confidence.

- The overall homology modeling procedure consists of six major steps and one additional step.

1. Template Selection :-

- The template selection involves searching the Protein Data Bank (PDB) for homologous proteins with determined structures.
- The search can be performed using a heuristic pairwise alignment search program such as BLAST or FASTA.

- However, programming based search programmes such as SSEARCH or ScanPS can result in more sensitive search results.
- Homology models are classified into 3 areas in terms of their accuracy and reliability.

Midnight Zone: Less than 20% sequence identity. The structure cannot reliably be used as a template.

Twilight Zone: 20% - 40% sequence identity. Sequence identity may imply structural identity.

Safe Zone: 40% or more sequence identity. It is very likely that sequence identity implies structural identity

- Often, multiple homologous sequences may be found in the database. Then the sequence with the highest homology must be used as the template.

2. Sequence Alignment :

- Once the structure with the highest sequence similarity is identified as a template, the full-length sequences of the template and target proteins need to be realigned using refined alignment algorithms to obtain optimal alignment.

- Incorrect alignment at this stage leads to incorrect designation of homologous residues and therefore to incorrect structural models.
- Therefore, the best possible multiple alignment algorithms, such as Praline and T-Coffee should be used for this purpose.

3. Backbone Model Building :

- Once optimal alignment is achieved, the coordinates of the corresponding residues of the template proteins can be simply copied onto the target protein.

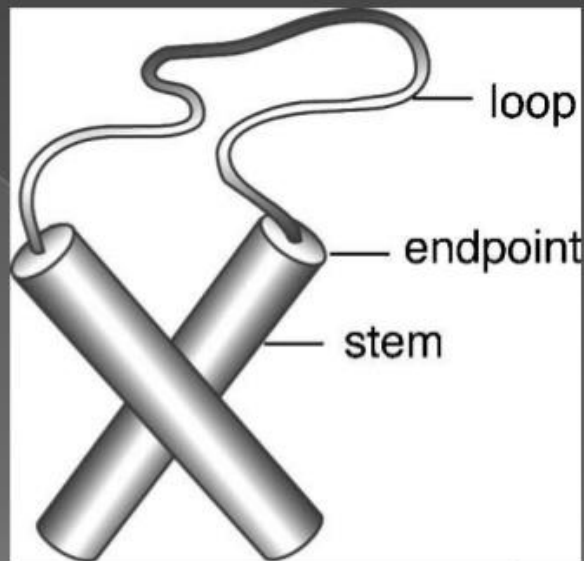
- If the two aligned residues are identical, coordinates of the side chain atoms are copied along with the main chain atoms.
- If the two residues differ, only the backbone atoms can be copied.

4. Loop Modelling :

- In the sequence alignment for modeling, there are often regions caused by insertions and deletions producing gaps in sequence alignment.

- The gaps cannot be directly modeled, creating “holes” in the model.
- Closing the gaps requires loop modeling which is a very difficult problem in homology modeling and is also a major source of error.
- Currently, there are two main techniques used to approach the problem: the database searching method and the ab initio method.
- The database method involves finding “spare parts” from known protein structures in a database that fit onto the two stem regions of the target protein.

- The stems are defined as the main chain atoms that precede and follow the loop to be modeled.
- The best loop can be selected based on sequence similarity as well as minimal steric clashes with the neighboring parts of the structure.
- The conformation of the best matching fragments is then copied onto the anchoring points of the stems.
- The ab initio method generates many random loops and searches for the one that does not clash with nearby side chains and also has reasonably low energy and ϕ and ψ angles in the allowable regions in the Ramachandran plot.



Schematic of loop modeling by fitting a loop structure onto the endpoints of existing stem structures represented by cylinders.

- FREAD is a web server that models loops using the database approach.
- PETRA is a web server that uses the ab initio method to model loops.
- CODA is a web server that uses a consensus method based on the prediction results from FREAD and PETRA.

5. Side Chain Refinement :

- Once main chain atoms are built, the positions of side chains that are not modeled must be determined.

- A side chain can be built by searching every possible conformation at every torsion angle of the side chain to select the one that has the lowest interaction energy with neighboring atoms.
- Most current side chain prediction programs use the concept of *rotamers*, which are favored side chain torsion angles extracted from known protein crystal structures.
- A collection of preferred side chain conformations is a rotamer library in which the rotamers are ranked by their frequency of occurrence.

- In prediction of side chain conformation, only the possible rotamers with the lowest interaction energy with nearby atoms are selected.
- A specialized side chain modeling program that has reasonably good performance is SCWRL, which is a UNIX program.

6. Model Refinement :

- In these loop modeling and side chain modeling steps, potential energy calculations are applied to improve the model.

- Modeling often produces unfavorable bond lengths, bond angles, torsion angles and contacts.
- Therefore, it is important to minimize energy to regularize local bond and angle geometry and to relax close contacts and geometric chain.
- The goal of energy minimization is to relieve steric collisions and strains without significantly altering the overall structure.
- However, energy minimization has to be used with caution because excessive energy minimization often moves residues away from their correct positions.

- GROMOS is a UNIX program for molecular dynamic simulation. It is capable of performing energy minimization and thermodynamic simulation of proteins, nucleic acids, and other biological macromolecules.
- The simulation can be done in vacuum or in solvents.
- A lightweight version of GROMOS has been incorporated in SwissPDB Viewer.

7. Model Evaluation :

- The final homology model has to be evaluated to make sure that the structural features of the model are consistent with the physicochemical rules.
- This involves checking anomalies in ϕ - ψ angles, bond lengths, close contacts, and so on.
- If structural irregularities are found, the region is considered to have errors and has to be further refined.

- Procheck is a UNIX program that is able to check general physicochemical parameters such as ϕ - ψ angles, chirality, bond lengths, bond angles, and so on.
- WHAT IF is a comprehensive protein analysis server that has many functions, including checking of planarity, collisions with symmetry axes, proline puckering, anomalous bond angles, and bond lengths.
- Few other programs for this step are ANOLEA, Verify3D, ERRAT, WHATCHECK, SOV etc.

THREADING/FOLD RECOGNITION:

- By definition, *threading or structural fold recognition predicts the structural fold of an unknown protein sequence by fitting the sequence into a structural database and selecting the best-fitting fold.*
- The comparison emphasizes matching of secondary structures, which are most evolutionarily conserved.
- The algorithms can be classified into two categories, pairwise energy based and profile based.

Pairwise Energy Method

- In the pairwise energy based method, a protein sequence is searched for in a structural fold database to find the best matching structural fold using energy-based criteria.
- The detailed procedure involves aligning the query sequence with each structural fold in a fold library.
- The alignment is performed essentially at the sequence profile level using dynamic programming or heuristic approaches.

- Local alignment is often adjusted to get lower energy and thus better fitting.
- The next step is to build a crude model for the target sequence by replacing aligned residues in the template structure with the corresponding residues in the query.
- The third step is to calculate the energy terms of the raw model, which include pairwise residue interaction energy, solvation energy, and hydrophobic energy.

- Finally, the models are ranked based on the energy terms to find the lowest energy fold that corresponds to the structurally most compatible fold.

Profile Method

- In the profile-based method, a profile is constructed for a group of related protein structures.
- The structural profile is generated by superimposition of the structures to expose corresponding residues.

- Statistical information from these aligned residues is then used to construct a profile.
- The profile contains scores that describe the propensity of each of the twenty amino acid residues to be at each profile position.
- To predict the structural fold of an unknown query sequence, the query sequence is first predicted for its secondary structure, solvent accessibility, and polarity.
- The predicted information is then used for comparison with propensity profiles of known structural folds to find the fold that best represents the predicted profile.

- Threading and fold recognition assess the compatibility of an amino acid sequence with a known structure in a fold library.
- If the protein fold to be predicted does not exist in the fold library, the method will fail.
- 3D-PSSM, GenThreader, Fugue are few web based programmes used for threading.

AB INITIO METHOD :

- When no suitable structure templates can be found, Ab Initio methods can be used to predict the protein structure from the sequence information only.
- As the name suggests, the ab initio prediction method attempts to produce all-atom protein models based on sequence information alone without the aid of known protein structures.

- Protein folding is modeled based on global free-energy minimization.
- Since the protein folding problem has not yet been solved, the *ab initio* prediction methods are still experimental and can be quite unreliable.
- One of the top *ab initio* prediction methods is called Rosetta, which was found to be able to successfully predict 61% of structures (80 of 131) within 6.0 Å RMSD (Bonneau et al., 2002).

➤ The basic idea of Rosetta is:

- To narrow the conformation searching space with local structure predictions &
- Model the structures of proteins by assembling the local structures of segments

➤ The Rosetta method is based on assumptions:

- Short sequence segments have strong local structural biases &
- Multiplicity of these local biases are highly sequence dependent

➤ **1st step of Rosetta:**

- Fragment libraries for each 3- & 9-residue segment of the target protein are extracted from the protein structure database using a sequence profile-profile comparison method

➤ **2nd step of Rosetta:**

- Tertiary structures are generated using a MC search of the possible combinations of likely local structures, &
- Minimizing a scoring function that accounts for nonlocal interactions such as:
 - ✓ compactness,
 - ✓ hydrophobic burial,
 - ✓ specific pair interactions (disulfides & electrostatics), &
 - ✓ strand pairing

CONTENT BEYOND THE SYLLABUS

Bioconductor is a [free](#), [open source](#) and [open development](#) software project for the analysis and comprehension of [genomic](#) data generated by [wet lab](#) experiments in [molecular biology](#).

Bioconductor is based primarily on the [statistical R programming language](#), but does contain contributions in other programming languages. It has two [releases](#) each year that follow the semiannual releases of R. At any one time there is a [release version](#), which corresponds to the released version of R, and a [development version](#), which corresponds to the development version of R. Most users will find the release version appropriate for their needs. In addition there are many [genome annotation](#) packages available that are mainly, but not solely, oriented towards different types of [microarrays](#).

While computational methods continue to be developed to interpret biological data, the Bioconductor project is an open source software repository that hosts a wide range of statistical tools developed in the R programming environment. Utilizing a rich array of statistical and graphical features in R, many Bioconductor packages have been developed to meet various data analysis needs. The use of these packages provides a basic understanding of the R programming / command language. As a result, R and Bioconductor packages, which have a strong computing background, are used by most biologists who will benefit significantly from their ability to analyze datasets. All these results provide biologists with easy access to the analysis of genomic data without requiring programming [expertise](#).

GenoCAD is one of the earliest [computer assisted design](#) tools for [synthetic biology](#).^[1] The software is a bioinformatics tool developed and maintained by [GenoFAB, Inc.](#) GenoCAD facilitates the design of protein expression vectors, artificial gene networks and other genetic constructs for [genetic engineering](#) and is based on the theory of [formal languages](#).^[2] GenoCAD can be used online by accessing the GenoFAB Client Portal at <https://genofab.com/>.

Apache Taverna is an [open source software](#) tool for designing and executing [workflows](#), initially created by the [myGrid](#) project under the name *Taverna Workbench*, now a project under the [Apache incubator](#). Taverna allows users to integrate many different software components, including [WSDL](#) SOAP or REST [Web services](#), such as those provided by the [National Center for Biotechnology Information](#), the [European Bioinformatics Institute](#), the [DNA Databank of Japan \(DDBJ\)](#), SoapLab, [BioMOBY](#) and [EMBOSS](#). The set of available services is not finite and users can import new service descriptions into the Taverna Workbench.^{[1][2][3][4][5][6][7][8]}

Taverna Workbench provides a desktop authoring environment and enactment engine for scientific workflows. The Taverna workflow engine is also available separately, as a Java API, command line tool or as a server.

Taverna is used by users in many domains, such as [bioinformatics](#),^{[9][10]} [cheminformatics](#),^[11] [medicine](#), [astronomy](#),^[12] [social science](#), [music](#), and [digital preservation](#).^[13]

Some of the services for the use in Taverna workflows can be discovered through the [BioCatalogue](#) - a public, centralised and curated registry of Life Science Web services. Taverna workflows can also be shared with other people through the [myExperiment social web](#) site for scientists.^[14] [BioCatalogue](#) and [myExperiment](#) are another two product from the [myGrid](#) consortium.

