BIF501 - Bioinformatics II

Topic - 1 Applications of Bioinformatics

Applications of Bioinformatics

1. Drug Development	10. Personalized Medicine
2. Crop Improvement	11. Preventive Medicine
3. Microbial Genome	12. Waste Cleanup
4. Gene Therapy	13. Antibiotic Resistance
5. Biotechnology	14. Alternate Energy Science
6. Comparative Study	15. Insect Resistance
7. Evolutionary Studies	16. Climate Change Studies
8. Veterinary Science	17. Nutritional Quality
9. Molecular Medicine	

Drug development

- ✓ Drugs target only about 500 proteins
- ✓ Disease mechanisms and using computational tools identify and validate new drug targets

Crop improvement

- ✓ Comparative genetics of the plant genomes
- ✓ Information obtained from the model crop systems can be used to suggest improvements to other food crops.
- ✓ At present the complete genomes of *Arabidopsis thaliana* (water cress) and *Oryza sativa* (rice) are available.

Microbial genome applications

- ✓ Complete genome sequences
- ✓ Environment, health, energy and industrial applications
- ✓ Isolation of the genes that give them their unique abilities to survive under extreme conditions.

Gene Therapy

- ➢ Gene therapy-used to treat, cure or even prevent disease
- ➢ Clinical trials

Biotechnology

- ➢ *Archaeoglobus fulgidus* and *Thermotoga maritima*
- ➢ *Corynebacterium glutamicum*
- ➢ *Xanthomonas campestris*
- ➢ *Lactococcus lactis*

Evolutionary studies

MUHAMMAD IMRAN

- The sequencing of genomes from all three domains of life; eukaryota, bacteria and archaea

Topic - 2 Applications of Bioinformatics

Veterinary Science

- Farm animals including cows and sheep have been sequenced

Molecular Medicine

- The human genome project
- 3000-4000 hereditary disease including Cystic Fibrosis and Huntingtons disease
- Response to an environmental stress causes cancers, heart disease, diabetes
- Human Genome Project Data Base

Personalized medicine

- Pharmacogenomics
- Sequence variants in DNA
- Trial and error to find the best drug
- Patient's genetic profile
- With the specific details of the genetic mechanisms of diseases being unraveled, the development of diagnostic tests to measure a persons susceptibility to different diseases may become a distinct reality.
- Preventative actions such as change of lifestyle or having treatment at the earliest possible stages when they are more likely to be successful, could result in huge advances in our struggle to conquer disease.

Waste cleanup

- *Deinococcus radiodurans*
- Potential usefulness in cleaning up waste sites that contain radiation and toxic chemicals

Antibiotic Resistance

- *Enterococcus faecalis*
- Virulence region-resistant genes
- The discovery of the region, known as a pathogenicity island

Alternative energy sources

- *Chlorobium tepidum*
- Capacity for generating energy from light

Insect resistance

- *Bacillus thuringiensis*
- Control serious pests of cotton, maize and potatoes
- Insecticides can be reduced and hence the nutritional quality of the crops is increased

Climate change Studies

- Increasing levels of carbon dioxide emission-global climate change.
- Study the genomes of microbes that use carbon dioxide as their sole carbon source.

Improve nutritional quality

MUHAMMAD IMRAN

- Genes transferred into rice to increase levels of Vitamin A, iron and other micronutrients
- Reducing occurrences of blindness and anaemia

Topic -3 Nucleotide Sequence Databases

Biological Databases:

Biological databases in general store biological data and their main goals are

1. **Data storage**
2. **Information retrieval**
3. **Knowledge discovery**

Classification:

Biological databases can be classified as

- **Primary databases (**that stores the Primary Sequences**)**
- **Secondary databases (**the primary sequences are annotated and kept in Secondary Databases**)**
- **Specialized Databases (**they are dedicated towards some specific organism or can have some disease data**)**

Biological databases can also be classified on the bases of types of data which they contain, such as:

- **Nucleotide databases**
- **Protein databases**
- **RNA databases**
- **Genome databases**
- **Expression databases (**Gene Expression Databases**)**

Issues:

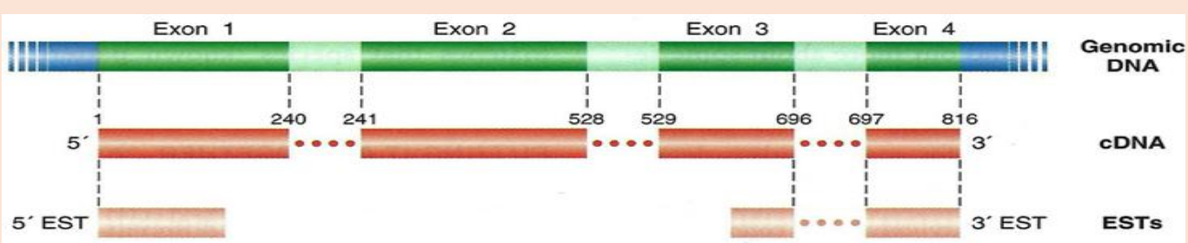
The issues which are present generally in other databases are also found to be in Biological databases that may be co-related with the relatively slow pace of quality assurance techniques as compared to the pace with which new data is emerging, so the issues are similar and are as follows:

Due to limited **Q/A**

- **Redundancy**
- **Inconsistency**
- **Incompatibility (format, terminology, data types, etc.)**

Nucleotide Sequence databases:

The Nucleotide Sequence Databases are one of the types of Biological Databases that contains nucleotide sequences in it, which can be DNA and cDNA or EST sequences.



Here, we have a diagram where we have a genomic DNA which has different *Exons*(we know that in Eukaryotes, we have exons and introns). So *exons*gets transcribed into mRNA and we can get cDNA from this mRNA through ***reverse transcription***and then we can store this cDNA into our databases whereas the ESTs are the subsets within those cDNA's.

MUHAMMAD IMRAN

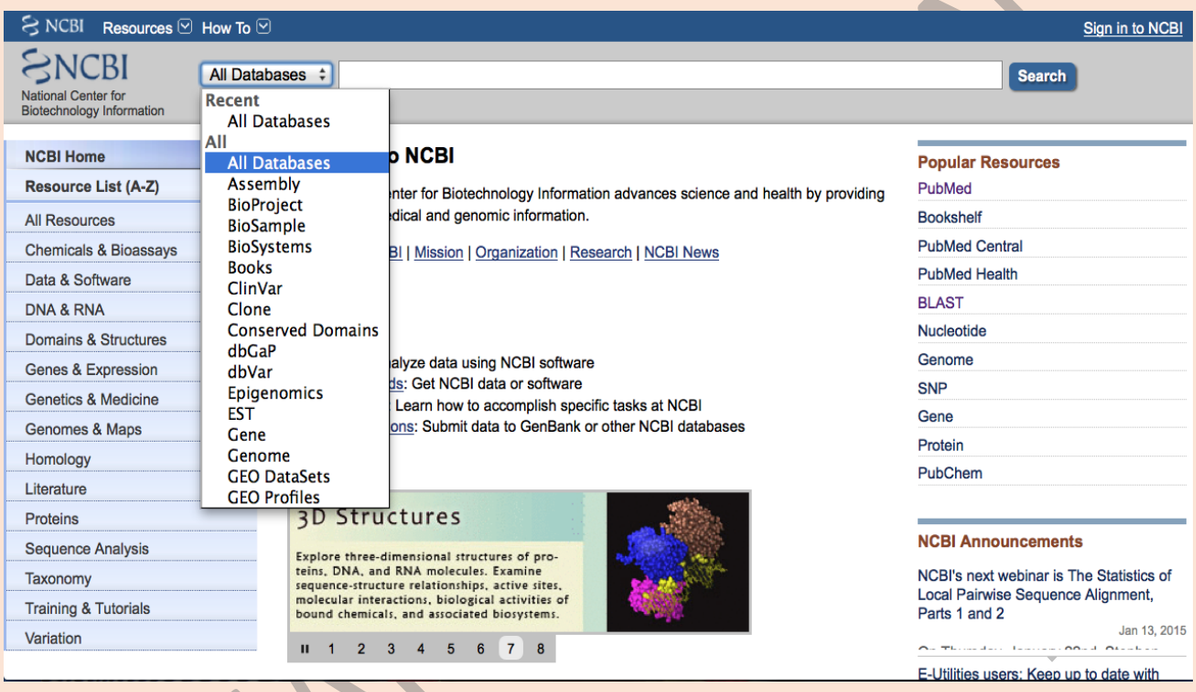
Origin:
The Nucleotide Sequence Databses were first assembled into Genebank (1982) at Los Alamos National Laboratory (LANL), New Maxico under the leadership of Walter Goad. **GeneBank**is now working under the umbrella of NCBI (National Center for Biotechnology Information).

NCBI is the central repository that stores multiple types of biological data that includes genomes, their assemblies, their sequencing data, their expression data and what not. In this diagram, we can see the page where you can search for any kind of data; a drop-down list which provides you with various options. The link to this page is http://www.ncbi.nlm.nih.gov/.

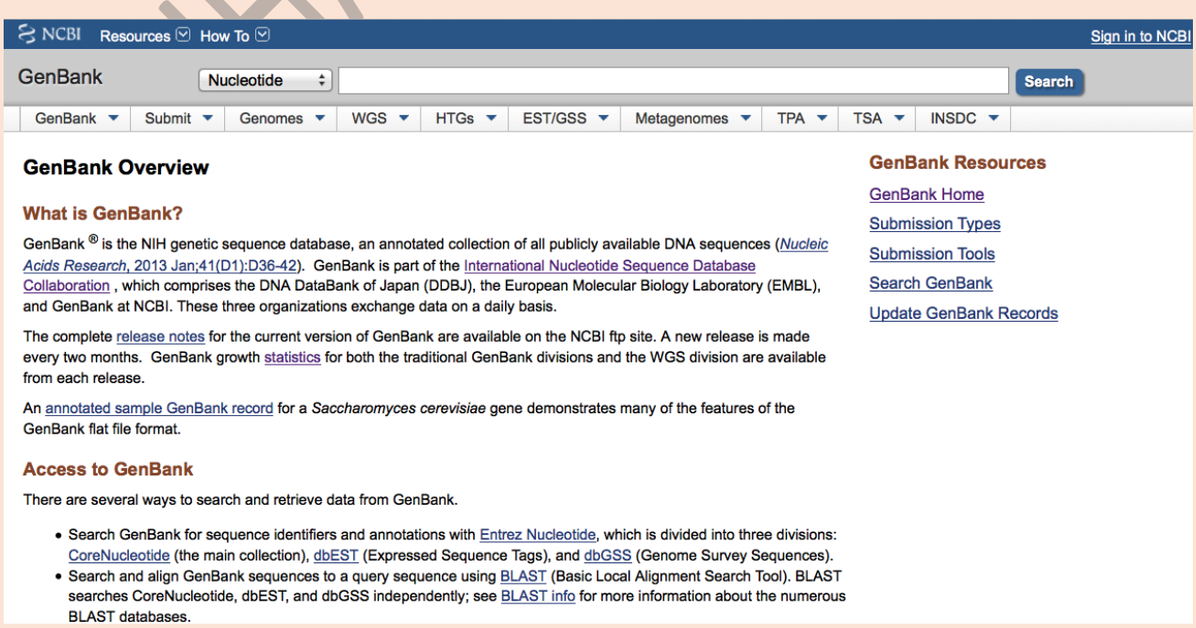
Here, is the page for GeneBank, so if you want search about nucleotides and genome sequences, this is the best resource

NCBI was established in United States.

National Center for Biotechnology Information



GeneBank



EMBL and DDBJ:

- European Molecular biology Lab (**EMBL** established1980) was established in Europe.

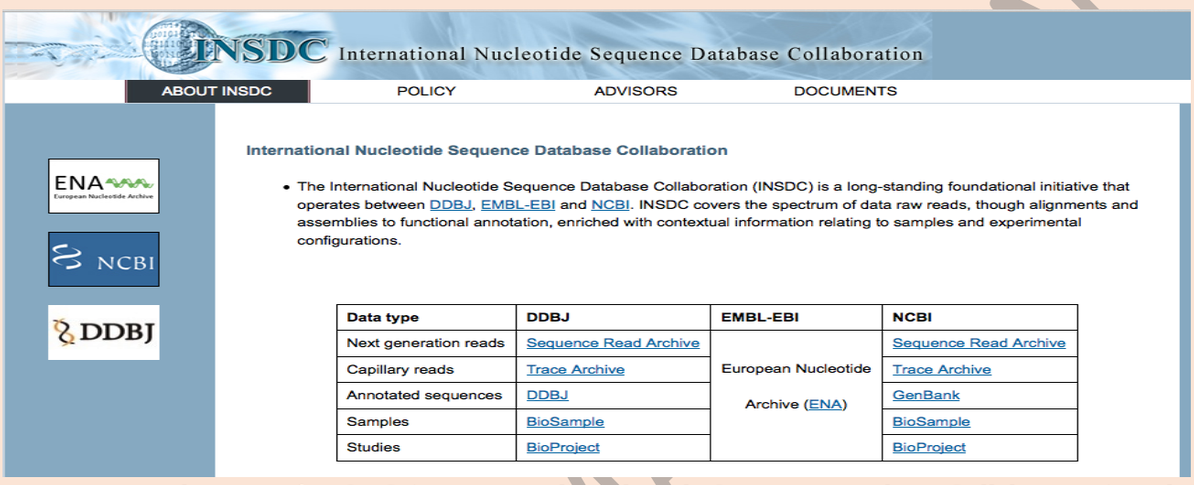
- DNA databank of Japan **(DDBJ)** established in Mishima japan (1984).

INSDC:

- Genebank, DDBJ and EMBL joined together in **I**nternational **N**ucleotide **S**equence **D**atabase **C**ollaboration (INSDC)

You can see in this diagram, the **NCBI** (**N**ational **C**enter for **B**iotechnology **I**nformation), **DDBJ**(**D**NA **D**atabank of **J**apan) and **EBI** (**E**uropean**B**ioinformatics**I**nstitute) /**ENA** (**E**uropean **N**ucleotide **A**rchive)forms an International collaboration known as **INSDC** (**I**nternational **N**ucleotide **S**equence **D**atabase **C**ollaboration).

Where **EMBL**established**EBI**, to deal with Bioinformatics kind of stuff and within them they have established **ENA** to maintain the DNA sequence datasets



Data type	DDBJ	EMBL-EBI	NCBI
Next generation reads	Sequence Read Archive	European Nucleotide Archive (ENA)	Sequence Read Archive
Capillary reads	Trace Archive		Trace Archive
Annotated sequences	DDBJ		GenBank
Samples	BioSample		BioSample
Studies	BioProject		BioProject

Here, is the page of INSDC (**I**nternational **N**ucleotide **S**equence **D**atabase **C**ollaboration), and you can observe that all three collaborators' logos are there. Similarly, if you look into the data, we can have Next Generation reads, Capillary reads and information about samples and annotated sequences all on this first page.

(We'll discuss it later).

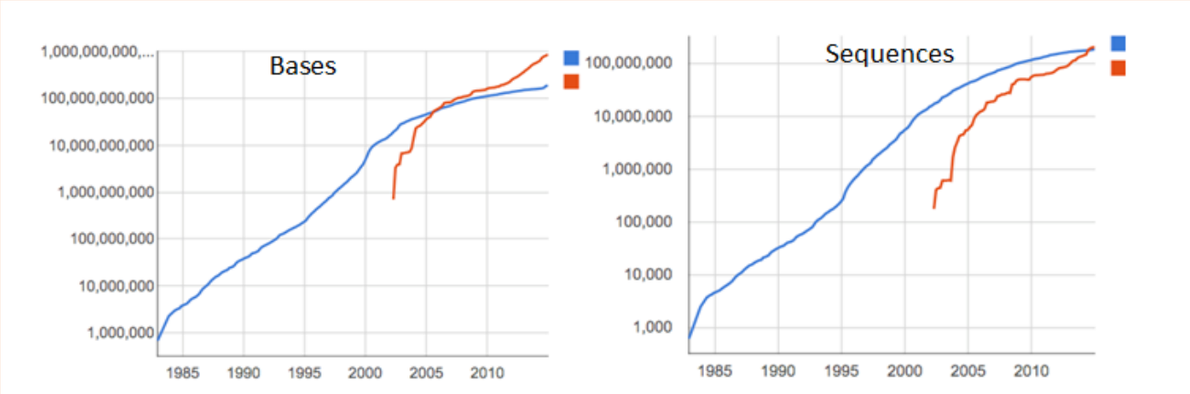
Growth of Genebank:

If we look into the growth of Gene Bank as shown in the figure below (left), we can see the number of **bases** in the GeneBank which are uncountable as they are in trillions which is a huge number starting somewhere in 1982 and if we look into these curves, blue is the growth of GeneBank and red one is the whole genome sequences (which we are comparing) which is starting somewhere in 2003 or 2004 after the publication of Human genome Project.

So, if you look into the number of bases, it seems like they double after every 18 month which means the growth is huge and is exponential.

Similarly, if we look into the sequences (right figure), are also around somewhere in 1000's in 1982 but now they are more than hundred million sequences in this GeneBank.

MUHAMMAD IMRAN



GeneBank

WGS

http://www.ncbi.nlm.nih.gov/genbank/statistics

Conclusions:

In the end, we conclude some of the followings:

- Biological databases store biological data.
- **INSDC** is joint venture of NCBI, EMBL and DDBJ.
- Growth of bases in **GeneBank** is exponential, doubling every 18 months.

Topic - 4 Protein Databases

Introduction:

Protein databases store protein data which may include the following:

- ✓ **Protein sequences**
- ✓ **Motif (**patterns of amino acids**)**
- ✓ **Structure**
- ✓ **Structure alignments (**aligned structures**)**

Origin:

First sequences to be collected were Proteins (before Nucleotide Sequences) using **Sanger and Tupy'**methods (1951) where Common protein families like cytochromes were sequenced (as in that era people were focusing on the sequences made from cytochrome molecules).

Atlas of protein sequences (mainly cytochromes) was assembled by Margret Dayhoff and her collaborators at **N**ational **B**iomedical **R**esearch **F**oundation (NBRF) in 1960s.

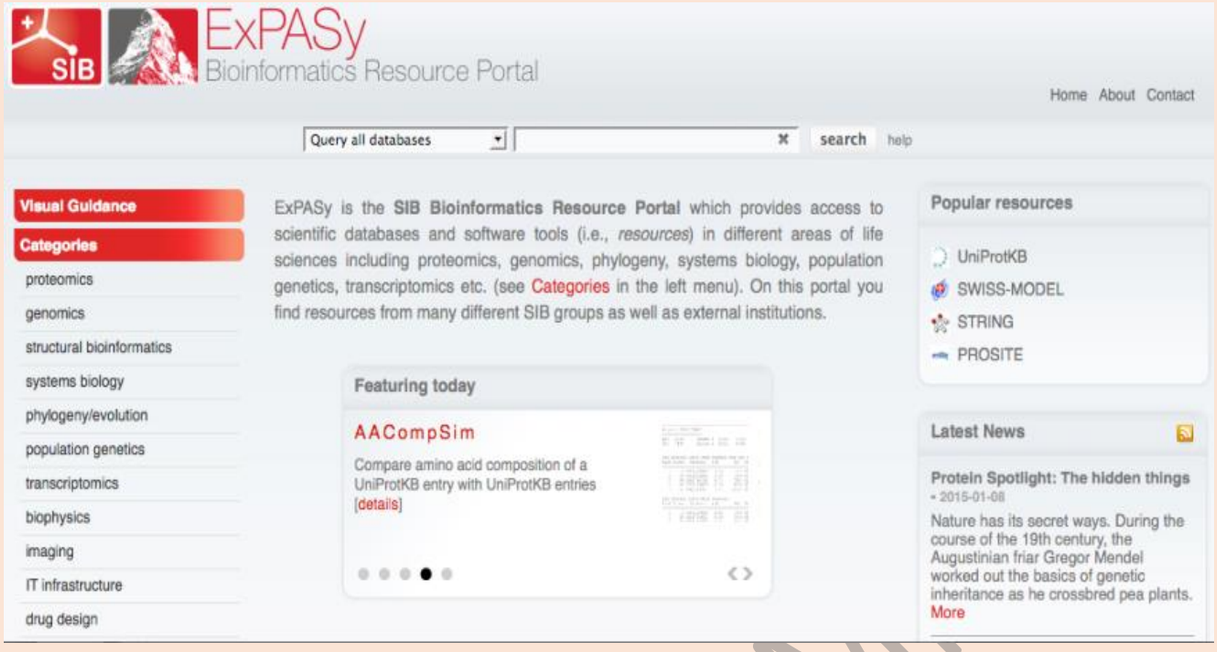
PIR (Protein Information Resource):

The collection (of ***Dayhoff*** and co) became **PIR (P**rotein **I**nformation **R**esource**)** which is now a collaboration of **NBRF**, **M**unich **C**enter for **P**rotein **S**equences (**MIPS**) and **J**apan **I**nternational **P**rotein **I**nformation **D**atabase (**JIPID**).

Protein Sequences:

MUHAMMAD IMRAN

Swiss-Protis a Collaboration between the SIB (**S**wiss**I**nstitue of **B**ioinformatics) and EBI (**E**uropean**B**ioinformatics**I**nstitute) and it weekly releases from about 50 servers across the



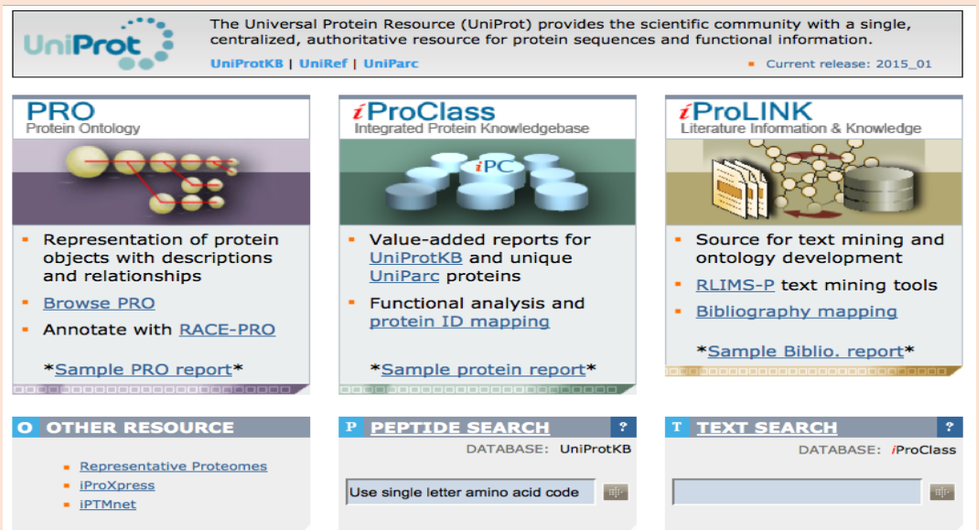
world, the main source beingExPASy in Geneva (i.e. it'smainlycontrolled by ExPASywhichis the main server located in Geneva).

Here, is the page for ExPASy, and you can find different structural alignments, proteomic data, genomic data.

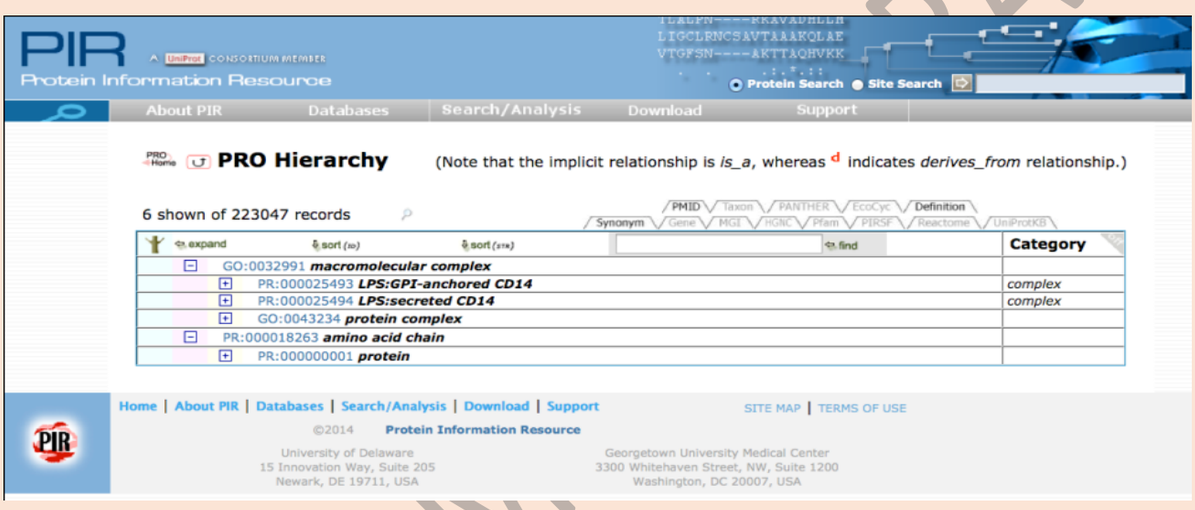
Protein Sequences:

International partnership between PIR, EBI and SIB created **UniProt**, by unifying the PIR-PSD (**P**rotein **I**nformation **R**esource – **P**rotein **S**tructure **D**atabase i.e. they kept the PSD of PIR into UniProt known as PIR-PSD), Swiss-Prot, and TrEMBL(is where we put the translated sequences from the DNA-where DNA is translated into the protein and the protein sequences are coming from different ***reading frames*** of those DNAs using all 6 reading frames- will be discussed later) databases.

MUHAMMAD IMRAN



Here, is the page of UniProt and you can see, we have 3 main sections i.e. Protein Ontologies labelled as PRO then we have ProClass where we can have the sequences and ProLINK tells us about the literature.

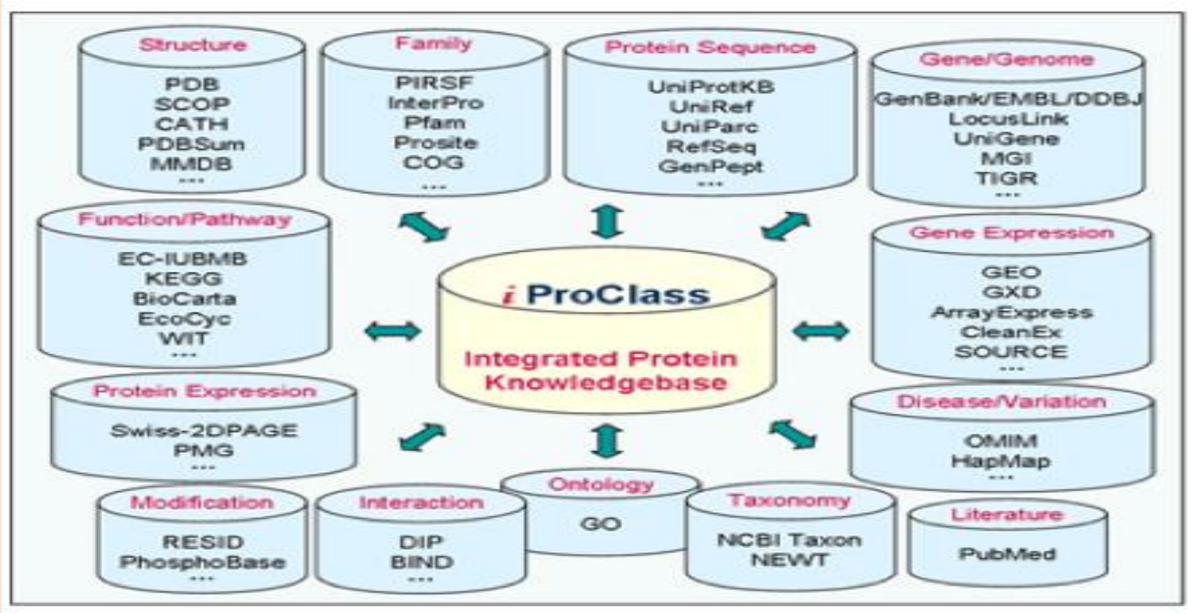


Here, we look into the PRO which is the Protein Ontologies- ontologies is where we can classify those proteins on the basis of their functions and different functions have their hierarchy so ontologies are labelled in form of different hierarchies, so there is a major function and a trend towards moving the specific function.

Here, we can see just a PRO Hierarchy Ontology in this example.

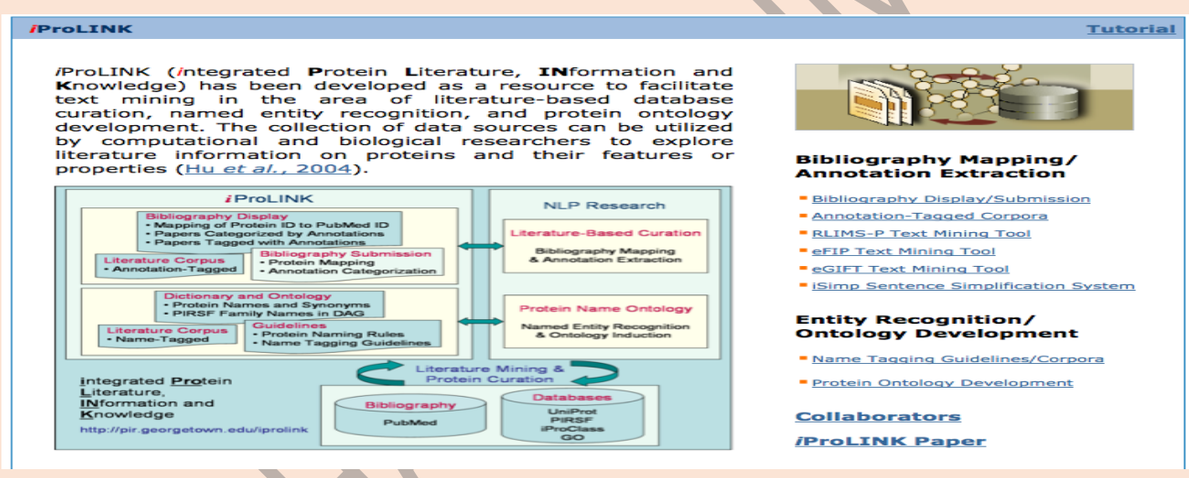
http://pir.georgetown.edu/

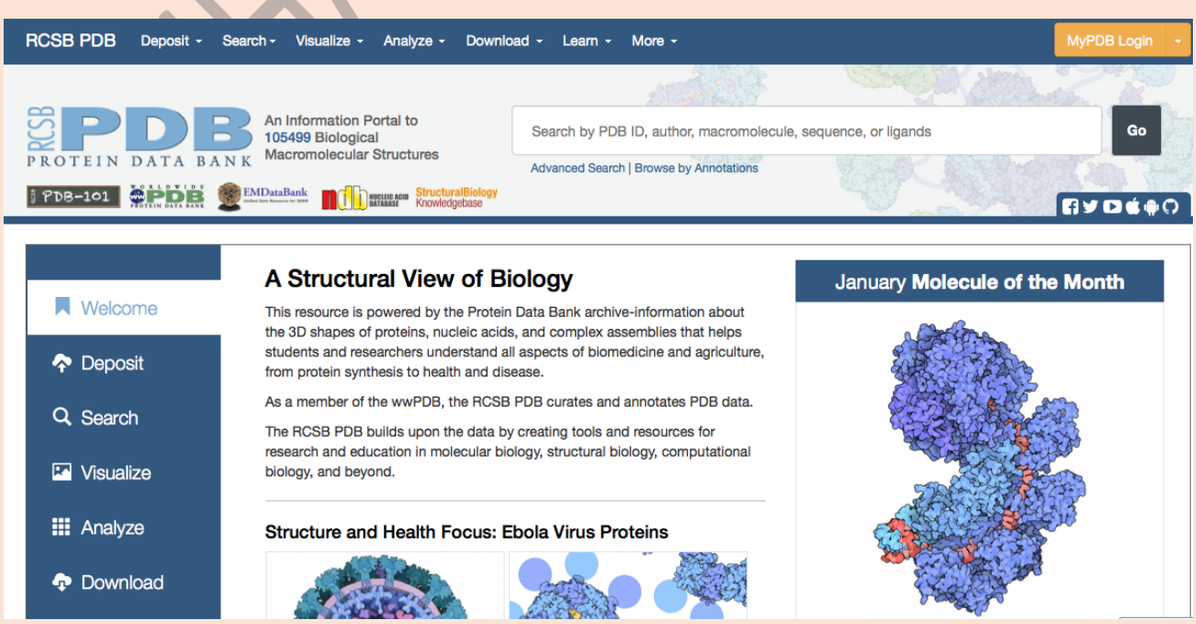
MUHAMMAD IMRAN



In this figure, as we mentioned earlier, *i*ProClass is the integration of different protein resources, so we can have sequences from here, protein expression data, and protein modifications. We can also integrate the genomic data with the proteomic data.

http://www.uniprot.org/



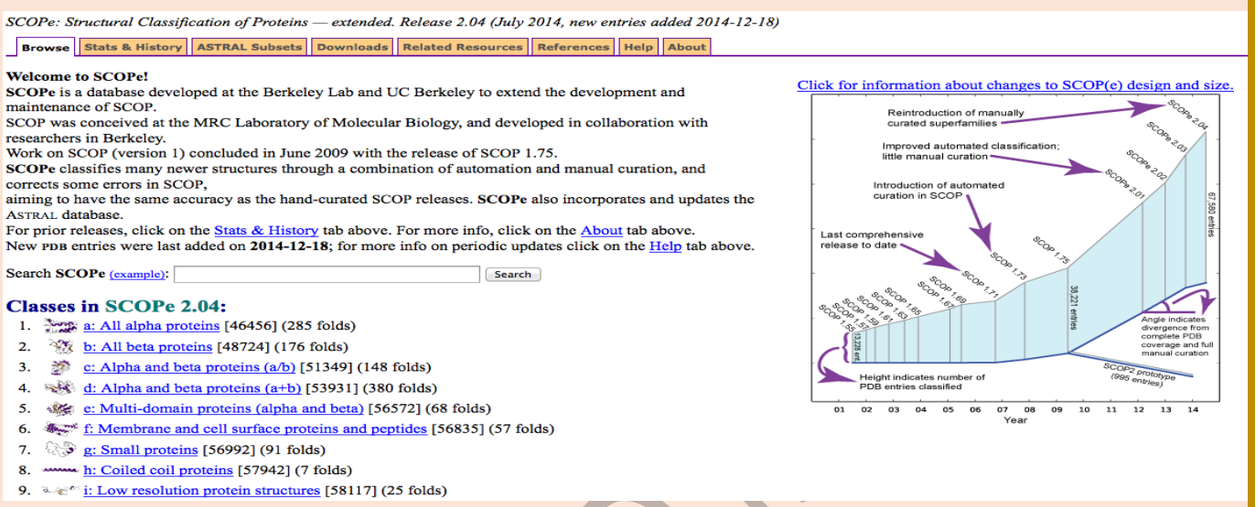


In this figure, we have *i*ProLINK which provides literature information and most of the research papers can be found here.

The link for it is: http://www.uniprot.org/.

Here, in this figure we have PDB and PDB stands for Protein Data Bank, basically it's a repository where we have the protein structures.

These structures are obtained by different chemistry and molecular biology techniques like X-ray Crystallography in the labs and then those structures are submitted into the PDB where researchers can get those structures and can compare their predicted structures with them, so it's a good resource if you are working on structural protein bioinformatics. The link for this page is http://www.rcsb.org/pdb/home/home.do

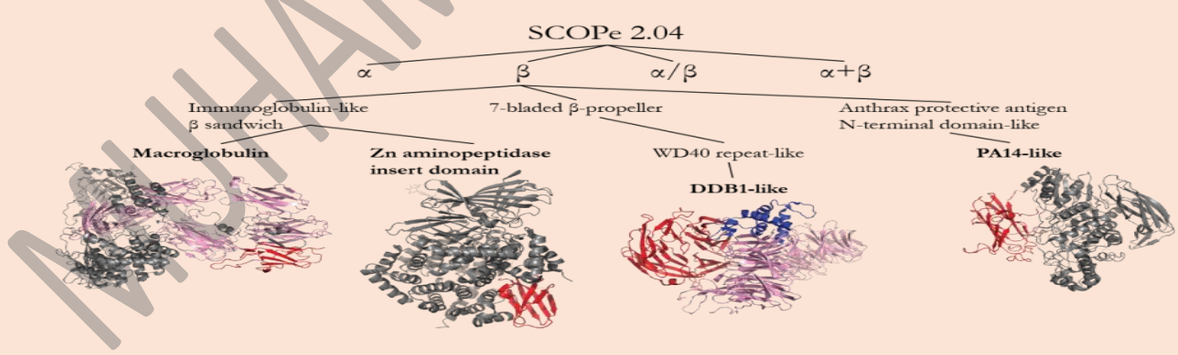


In this figure, we can see SCOP which is a similar effort that utilizes different structural elements on those proteins and it classify those proteins on the basis of their structural elements like family, fold super family, domains and then classes.

So, *Class* is the biggest in this SCOP hierarchy, there

are different major group of classes.

The link is : http://scop.mrc-lmb.cam.ac.uk/scop/



We can see here for an example, we have the *class* in which we have all the alpha helices; these helices are formed by special arrangement of amino-acids. Basically when the protein sequences- just a linear sequence of amino-acids when it turns around on it selves, it forms those secondary structures so those structures are then recognized as alpha and beta (we are not going into the details; you can go for molecular biology course or Google about alpha or beta). The main idea to present here is that SCOP actually classify the proteins on the basis of those structures so for an example, alpha (is that class where we have all those proteins that has alpha helices in them), we can also have beta (where we have all those proteins that has beta chains in them), alpha/beta (where we have alpha helices then comes beta then comes alpha so they are present one after the other), alpha + beta (we can have separate regions where we can have alpha helices stacked together and then we have beta chains stacked together). And the link for it is - http://scop.mrc-lmb.cam.ac.uk/scop/.

Conclusions:

We conclude that:

- ✓ First sequences to be collected were Protein sequences.
- ✓ Protein databases are classified on the basis of sequences, motifs, structures and different structural alignments.
- ✓ Growth of Sequence in Databases is exponential (just like as in Nucleotide Databases the growth of sequence is higher).

Topic - 5 Genome & Organism Specific Databases

Origin:

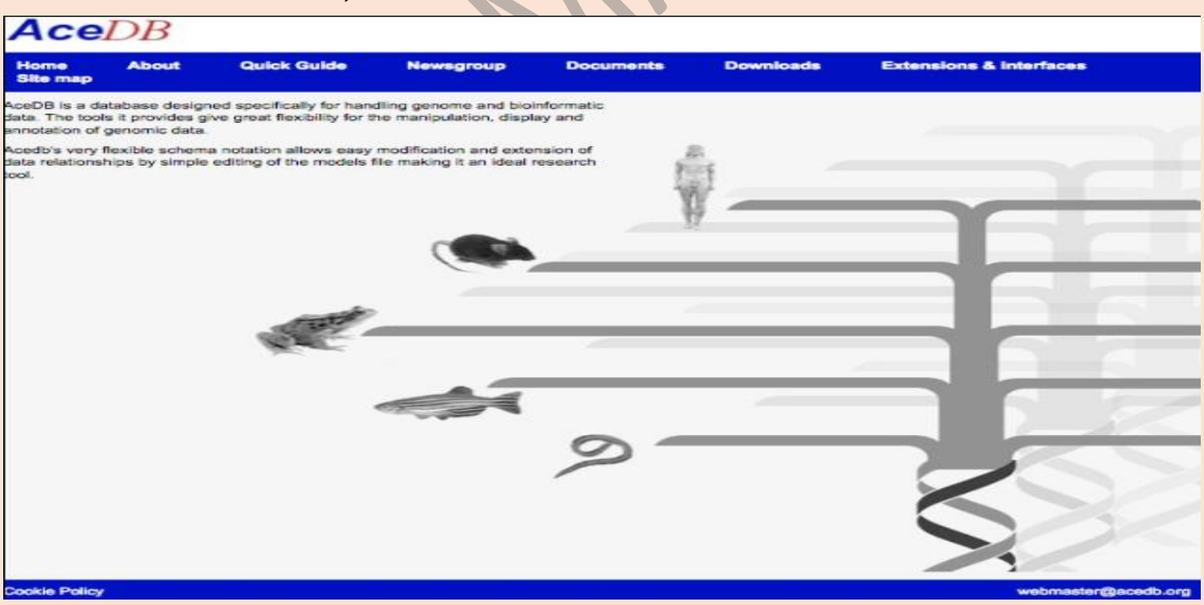
First attempt to sequence free living organism was launched in late 1990's (Blattner et al. 1997) and Viruses had already been sequenced (Fleischmann et al. 1995).

Haemophilusinfluenzaewas the first genome that was published and the project was initiated at **T**he **I**nstitute of **G**enome **R**esearch **(TIGR)** under the leadership of **Craig Ventor (**the same person who's name we'll see in the human genome project**)**. At that time, a method which was already established known as **shotgun sequencing method** was being tested by this project to verify its reliability and efficiency. And by utilizing this method they sequenced the genome which was about 1.8 million base pairs (bp), it took 9-months and the cost was around 1 million US dollars and this project Paved the way for sequencing of many other organisms.

Examples

- **AceDB** (**A***C. e*legans**D**ata**B**ase) was the first genome database for genome sequences was developed in 1989 and was established by **Richard durbin**and**Thierry-Miegi.**

  (Same Durbin whose book "***Biological Sequence Analysis***" we'll consider in the latter half of the course**).**



Here is the figure of AceDB webpage, you find the *sea elegans*; a worm and there are other organisms.

Link for this page is: http://www.acedb.org/.

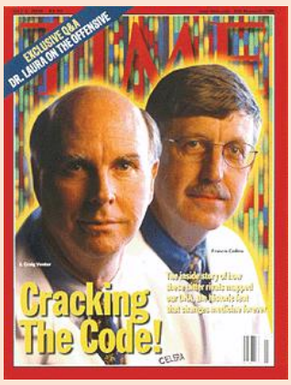
Examples:

TAIR (**T**he **A**rabidopsis **I**nformation **R**esource) which is a database for Arabidopsis (http://www.arabidopsis.org/) and **SGB** (**S**accharomyces **G**enome **D**atabase) actually uses the system of AceDB (http://www.yeastgenome.org/).

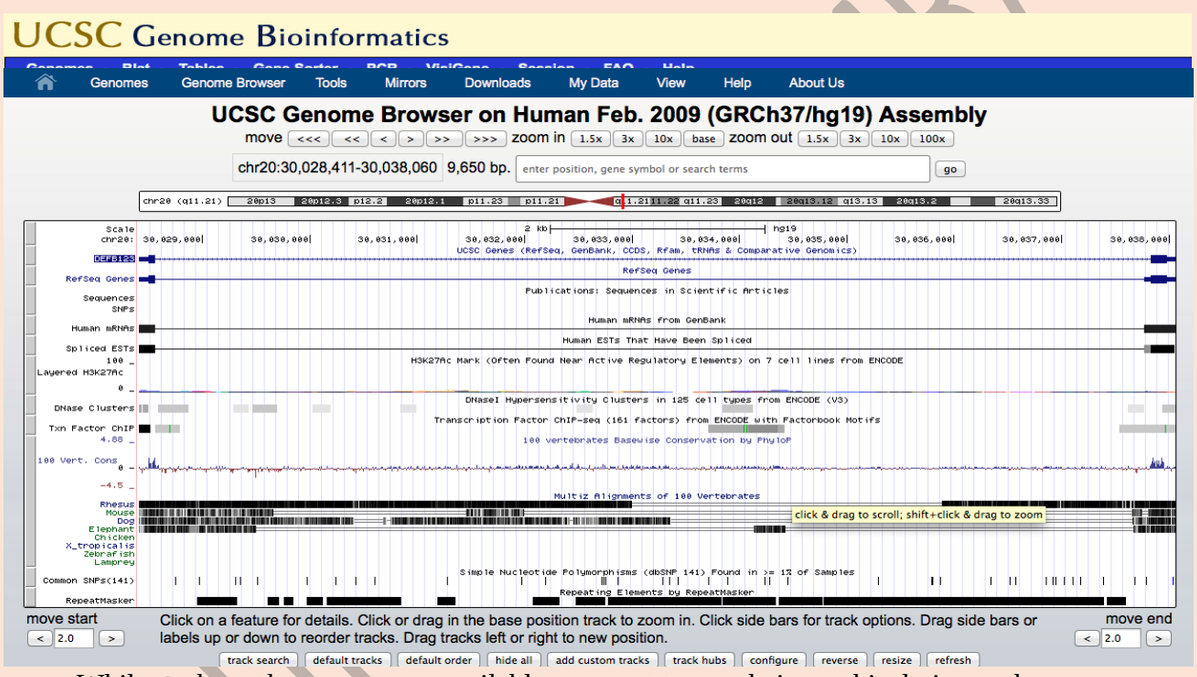
MAD IMRAN

Human Genome Project:

Human Genome Project started initially as a Pilot Project which begun by **D**epartment **O**f**E**nergy (**DOE**) of USA in 1986. Two organizations, one is **N**ational **H**uman **G**enome **R**esearch **I**nitiative (**NHGRI**), which was federally funded organization through NIH (**N**ational **I**nstitute of **H**ealth) that started in 1988 by **Francis Collin** which was joined to Commercial organization named ***Celera (Celera Genomics)***in 1998, a commercial under the leadership of **Craig Venter.**



Both of them claimed that they sequenced the full Human Genome and the issue was raised that who completed the project first, and who had a major role in it but later on it was resolved by President Bill Clinton at that time and they published it together back in year 2000. In the end, they concluded that there areTotal 3.4 billion bases which are sequenced at a cost of $1/base.



While we have those genomes available, we want to see their graphical views where we can get the reports, get the idea about where different genes are located, so in order to do that we needed to make something which we call it as genome browsers- are the webpages where we can look into the different features within our genomes so UCSC is one of the example (shown on the left) which is University of California Santa Cruz which is the biggest genome browser. The link to this browser is http://genome.ucsc.edu/.

The figure of UCSC Genome Browser, where we can have information, so on the top we see a chromosome and down below we see various lines which are known as different tracks (for snips, genes, EST's etc.) so we can look or zoom into different regions of the genome by using those genome browsers.

The link to this webpage is: http://genome.ucsc.edu/.

Conclusion:

In the end, we conclude the following:

- ✓ Success of ***Haemophilusinfluenzae***paved the way for other genome sequencing projects

- ✓ Human Genome Project was accomplished by NHGRI and Celera (they were working independently from one another).
- ✓ Genome browsers help in exploring different regions of the genome.

Topic -6 Gene Expression Databases

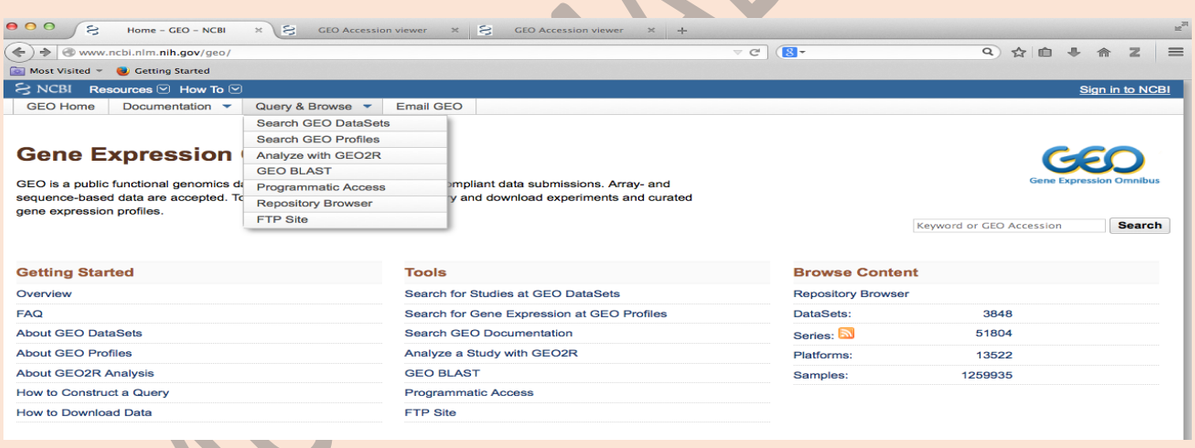
Gene Expression Omnibus (GEO):

Genes are expressed into mRNA, and whenever we talk about gene expression, we generally mean the mRNA sequences so we can normally get those mRNA from techniques like microarray and another famous technique nowadays which is being established is known as RNAseq. And microarray data and RNAseq can be classified into Gene Expression Data which is stored in Gene Expression Databases.

Basically, Gene Expression Databases are the public repository for the archiving and distribution of gene expression data submitted by the scientific community.

The gene expression data are MIAME compliant data (where MIAME is Minimum Information About a Microarray Experiment and can be searched on this link: http://www.mged.org/Workgroups/MIAME/miame.html). Whenever you need to submit a microarray experiment, you need to give descriptions like how exactly the samples were prepared, normalization methods which you have used and other counts and normalized files) so while keeping in view these datasets have different records in it.

Gene Expression Omnibus is convenient for deposition of gene expression data, as required by funding agencies and journals and it's also a curated resource for gene expression data where we can do Browsing, querying, analysis and retrieval of the data.



Here, is the webpage of GEO which is Gene Expression Omnibus running under NCBI (you can visit NCBI where you can get to the GEO Database) which are having different datasets, has expression profiles where we can see the change in expression of genes across different treatments and we can also analyze this expression data. There is a tool called as GEO2R, we can use BLAST in it. (We'll discuss later)

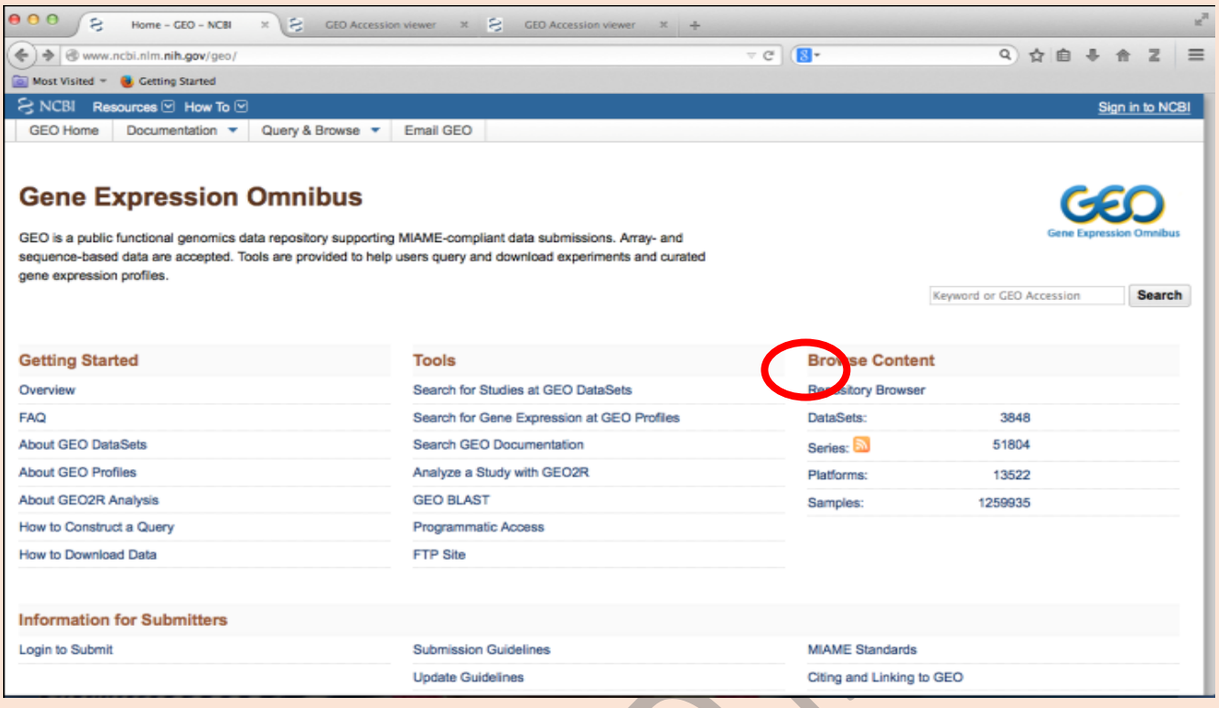
http://www.ncbi.nlm.nih.gov/geo/

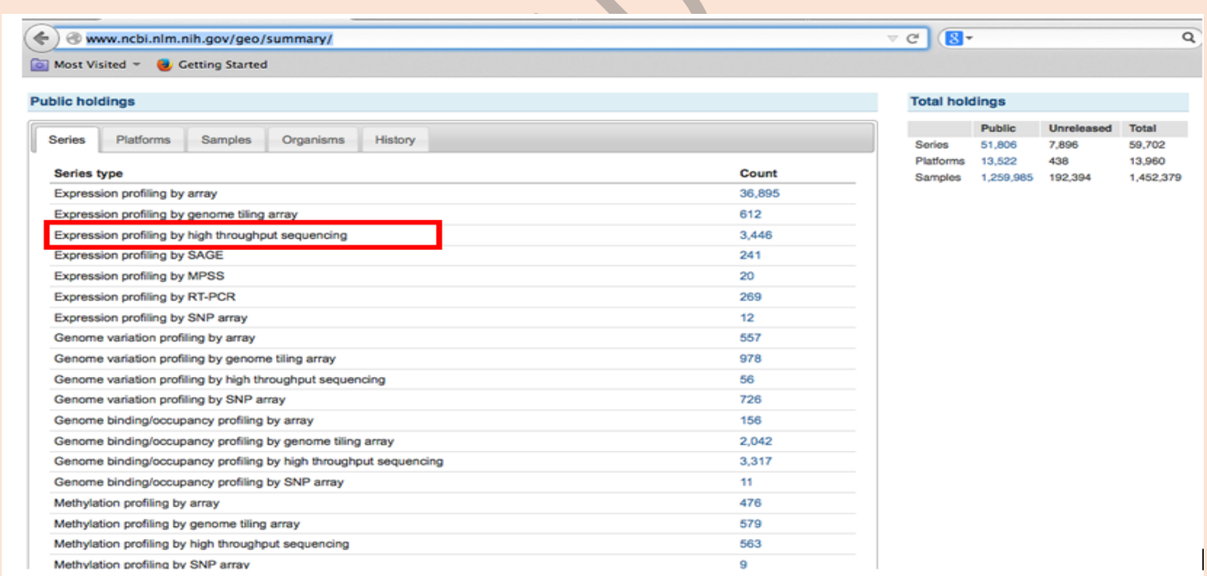
Gene Architecture:

GEO has four kinds of records or data files (keeping in view the MIAME rules) and are as follows:

- ✓ **Sample(GSM) –** these files stores the sample information like how the samples are prepared, how the treatments are given, how the experimental design was established.
- ✓ **Platform (GPL) –** The idea about platforms, they are stored in GPL files so here we can see whether it's a microarray data or RNAseq data (there are different protocols coming from different agencies so we can have that information).

- ✓ **Series (GSE) –** Sometimes different treatments are recorded as separate files so GSE are the files where we can have the similar treatment files and they are put together in a shape of series (are a set of samples and which are somehow related).
- ✓ **Datasets (GDS) –** Whereas the actual data is stored as GDS files which are the sample data collections and are assembled by GEO.

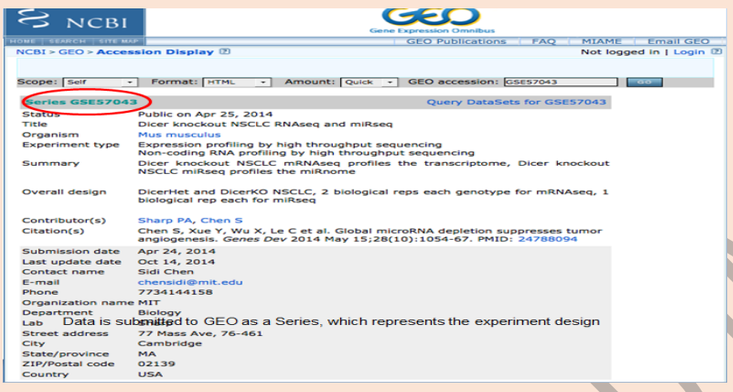




Here, is the Gene Expression Omnibus page and if we look into the different types of datasets it have, we can have *Series* (on the top left side of right figure), different records for the *Platform, Samples.* If you look into the types of series, you can see there are expression profiling by array, expression profiling by high throughput sequencing (in our course we'll be getting some RNAseq data which is under the expression profiling by high throughout sequencing), similarly there are other various techniques for getting the expression which are

listed below in the *Series* section as can be seen and number of datasets available are also present in the column called as *count.*

If you want to look into some dataset, you can simply type into search bar say for example, you write colon cancer RNAseq data which leads us to the sets of records it gets and when we click onto one of them the page appears (shown below).

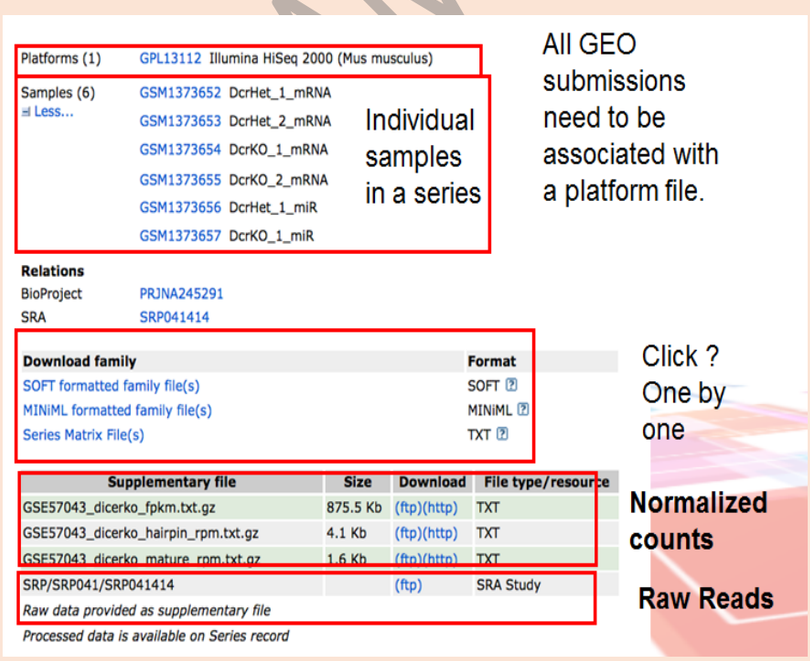


And we get this file, so here is the information of the particular dataset.

You can see on the top, that each dataset is submitted as a series with a unique number, say for example here the number is GSE57043 which is basically an id number for this dataset.

So, if we look into this page, we can have the idea about the experiment, the organism from which it's coming, type of experiment, a little bit summary of the experiment, we can also see the contributors name, their publications and addresses.

It is a huge page which is portioned and the other side of it is shown in the figure below

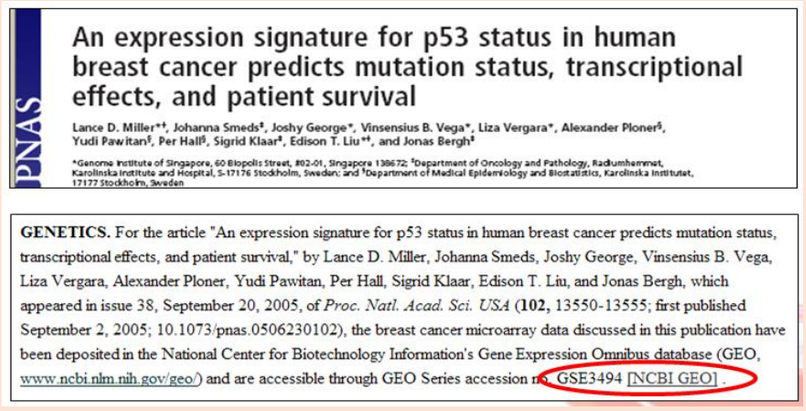


This is the bottom part of the same webpage shown above.

Here, we can see that we need to be associated with the platforms so the platform information is can be found on this page, and we get those sequences which are sequenced by machine Illumina HiSeq 2000 (we'll be discussed in upcoming lectures).

There are total six samples in this dataset, so individual samples are put together labelled as GSM.

We can also download the sequence expression counts or values in different formats and there are also some normalized counts as shown in the figure, these are compressed files. There are also raw reads data are present in the format, which we call as SRP or Sequencing Read Archive so that stores the raw read data.


PNAS

An expression signature for p53 status in human breast cancer predicts mutation status, transcriptional effects, and patient survival

Lance D. Miller*†, Johanna Smeds‡, Joshy George*, Vinsensius B. Vega*, Liza Vergara*, Alexander Ploner§, Yudi Pawitan§, Per Hall§, Sigrid Klaar‡, Edison T. Liu*†, and Jonas Bergh‡

*Genome Institute of Singapore, 60 Biopolis Street, #02-01, Singapore 138672; ‡Department of Oncology and Pathology, Radiumhemmet, Karolinska Institute and Hospital, S-17176 Stockholm, Sweden; and §Department of Medical Epidemiology and Biostatistics, Karolinska Institutet, 17177 Stockholm, Sweden

GENETICS. For the article "An expression signature for p53 status in human breast cancer predicts mutation status, transcriptional effects, and patient survival," by Lance D. Miller, Johanna Smeds, Joshy George, Vinsensius B. Vega, Liza Vergara, Alexander Ploner, Yudi Pawitan, Per Hall, Sigrid Klaar, Edison T. Liu, and Jonas Bergh, which appeared in issue 38, September 20, 2005, of *Proc. Natl. Acad. Sci. USA* (**102,** 13550-13555; first published September 2, 2005; 10.1073/pnas.0506230102), the breast cancer microarray data discussed in this publication have been deposited in the National Center for Biotechnology Information's Gene Expression Omnibus database (GEO, www.ncbi.nlm.nih.gov/geo/) and are accessible through GEO Series accession no. GSE3494 [NCBI GEO] .

Since, funding and the publication agencies demands that your data should be submitted and shared with the community so here is an example (in the figure shown) where we can see a publication and they have put GSE into their publication which helps other scientists to get access to this data using this ID number as highlighted (in the figure).If you are submitting your paper, you need to provide this information to the publication agencies which is an essential consideration.

Conclusion:

So we sum-up that **GEO** is a public repository for the archiving and distribution of gene expression data and is the Best resource to get microarray and Next Generation Sequencing (RNASeq) data.

Topic -7 Medical Databases

Introduction

Informatics in health care may be called as health informatics

- It deals with the resources, devices, and methods required to optimize the acquisition, storage, retrieval, and use of information in health and biomedicine.

(wiki)

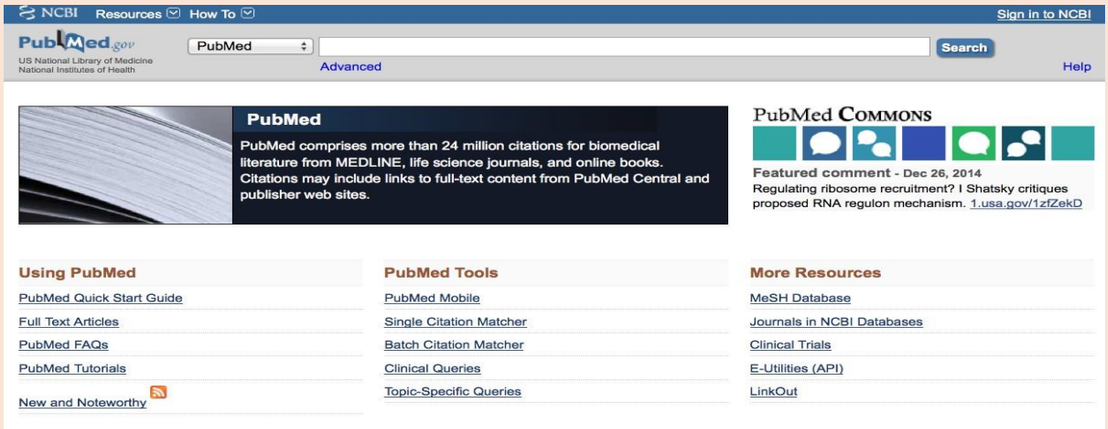
Medical databases store and provide medical information

- The premier database for biomedical literature is the National Library of Medicine (NLM)'s MEDLINE, accessible through PubMed

PUBMED

- Comprises of more than 24 million citations for biomedical literature from MEDLINE, life science journals, and online books
- Citations may include links to full-text content from PubMed Central and publisher web sites

MUHAMMAD IMRAN



MEDLINE

- MEDLINE is the primary resource for biomedical journal articles
- Millions of citations to articles in biomedical journals
- MEDLINE uses the MeSH vocabulary

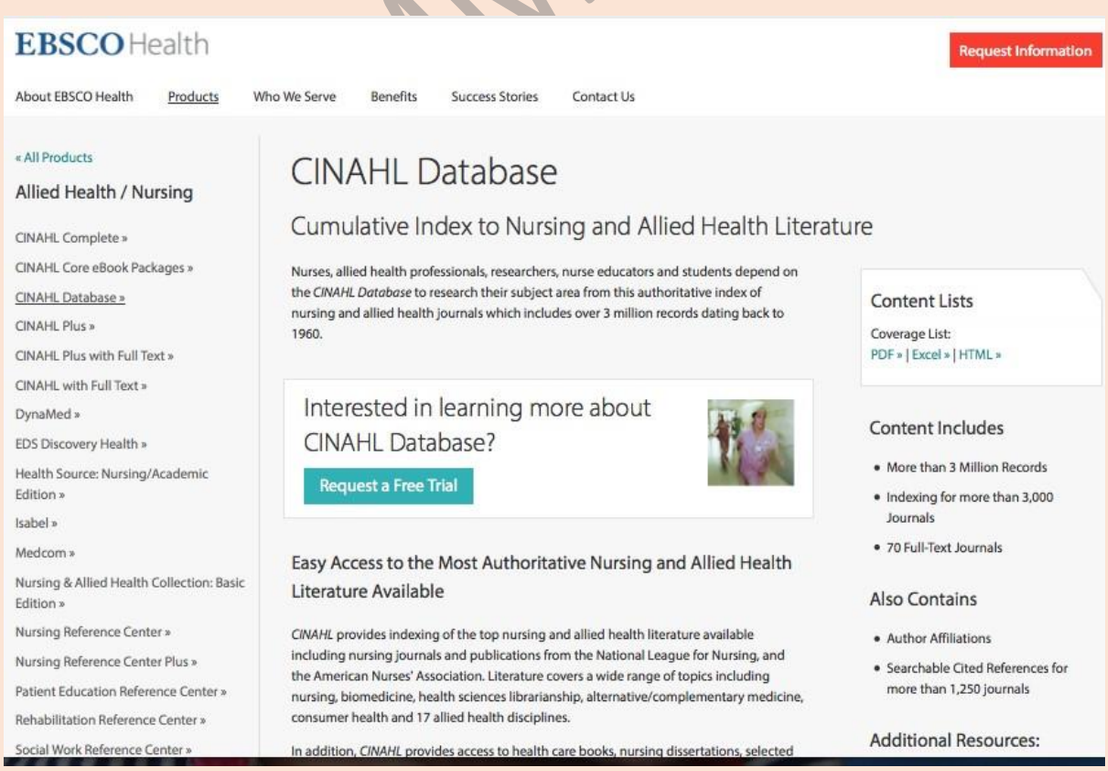
Other Databases

MEDLINE is the primary resource, but other databases may also be helpful

- Academic OneFile
- CINAHL (Cumulated Index of Nursing and Allied Health Literature)
- PsycINFO
- Web of Knowledge

Academic OneFile

- Academic OneFile lists articles from journals covering a broad range of subjects
- While it does not primarily focus on medical topics, useful articles can still be found here

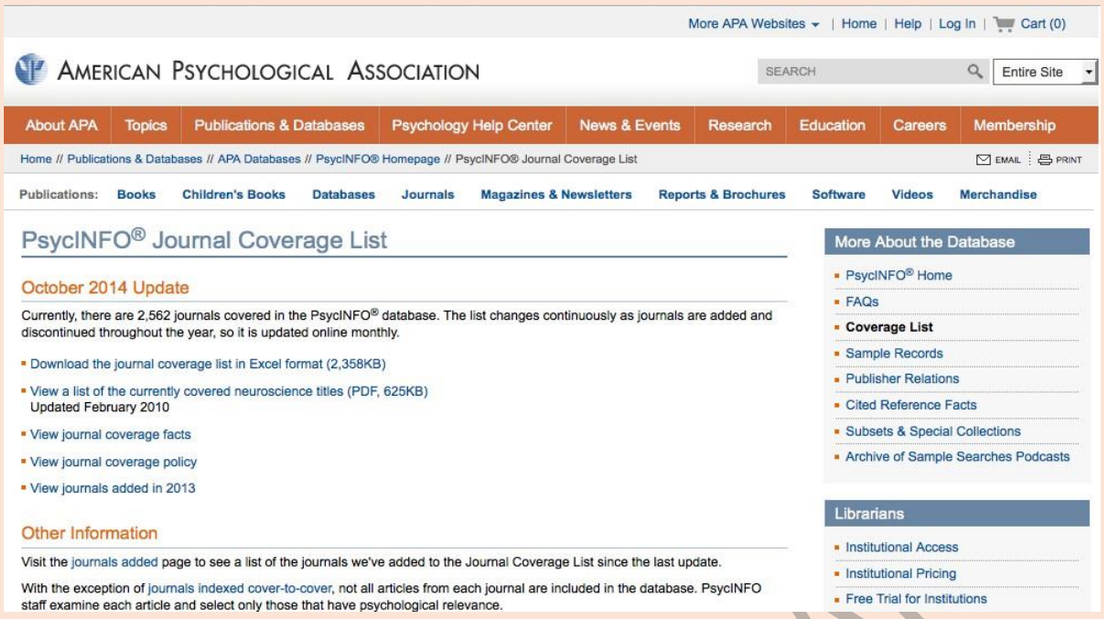


PsycINFO

- PsycINFO searches the psychological literature
- While it does not primarily focus on medical topics, useful articles can still be found here

MAD IMRAN

- http://www.apa.org/pubs/databases/psycinfo/coverage.aspx



http://www.apa.org/pubs/databases/psycinfo/coverage.aspx

Web of Science

- Major source for articles in a wide range of fields, including the sciences, social sciences, and humanities.
- Excellent place to find articles from scientific journals that may not be included in MEDLINE

Conclusions

Informatics in health care may be called as health informatics

- Medical databases deal with the acquisition, storage, retrieval, and use of information in health and biomedicine.

Topic -8 Sequence Submission Introduction

Introduction

Sequences are submitted to the databases in order to share them with the scientific community (sometimes they are also required by the Publication and funding agencies to submit them). Generally sequences are submitted at the time of publication and are reviewed by peers.

Caution

It is important to ensure that sequence files do not contain any special characters because sometimes the control characters can also be incorporated into or normal sequences, which can then mess-up the down-stream analysis or data-transfer.

MUHAMMAD IMRAN

Table 2.1. *Base–nucleic acid codes*

Symbol	Meaning	Explanation
G	G	Guanine
A	A	Adenine
T	T	Thymine
C	C	Cytosine
R	A or G	puRine
Y	C or T	pYrimidine
M	A or C	aMino
K	G or T	Keto
S	C or G	Strong interactions 3 h bonds
W	A or T	Weak interactions 2 h bonds
H	A, C or T not G	H follows G in alphabet
B	C, G or T not A	B follows A in alphabet
V	A, C or G not T (not U)	V follows U in alphabet
D	A, G or T not C	D follows C in alphabet
N	A,C,G or T	Any base

Adapted from NC-IUB (1984).

Mount, pg 28

So, there is an issue of how to put the ambiguous nucleotides or amino acids in the sequences (because at some places you are not sure whether it is 'A' or 'T' or 'G' or 'C' and you are restricted to put a single letter). So, there is an organization known as International Union of Biochemistry (IUB), it has established some standard codes to represent those ambiguous bases or amino acids.

For example, here we see that G, A, T or C are just Guanine, Adenine, Thymine and Cytosine respectively. If we see R, it can be either A or G and the word is derived from the group they are coming from i.e. the puRines. We see Y that is the pYrimidine, it can be C or T.

M stands for if they are having some amine group / amino group in them, K is if they have Keto group i.e. G or T.

S is if they have strong interactions (3 hydrogen bonds) like C and G, who forms triple bonds.

W is for weak interactions, A or T.

Since H follows G in Alphabet so it's everything except G, it can be A, C or T and similar procedure is followed for B,V, and D whereas N can be any base.

Table 2.2. *Table of standard amino acid code letters*

1-letter code	3-letter code	Amino acid
A[a]	Ala	alanine
C	Cys	cysteine
D	Asp	aspartic acid
E	Glu	glutamic acid
F	Phe	phenylalanine
G	Gly	glycine
H	His	histidine
I	Ile	isoleucine
K	Lys	lysine
L	Leu	leucine
M	Met	methionine
N	Asn	asparagine
P	Pro	proline
Q	Gln	glutamine
R	Arg	arginine
S	Ser	serine
T	Thr	threonine
V	Val	valine
W	Trp	tryptophan
X	Xxx	undetermined amino acid
Y	Tyr	tyrosine
Z[b]	Glx	either glutamic acid or glutamine

Adapted from IUPAC-IUB (1969, 1972, 1983).
[a] Letters not shown are not commonly used.
[b] Note that sometimes when computer programs translate DNA sequences, they will put a "Z" at the end to indicate the termination codon. This character should be deleted from the sequence.

Mount, pg 28

Similarly, for amino acids, we have single letter codes i.e. from A to Z. And we can see in the figure on the left that some letters are missing.

There are four amino acids that are starting with G, but we gave that G letter to Glycine and for rest of them, we might use some other letters like Glutamic acid is represented as E.

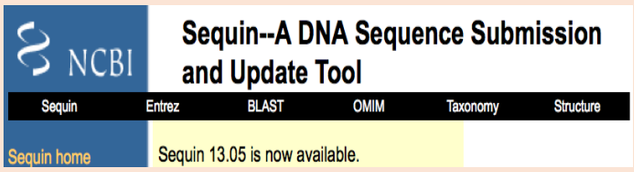
Y stands for Tyrosine (down below) and X can be any amino acid like N (in case of the nucleotide sequences).

NCBI:

NCBI has two options for sequence submission

BANKIt - for simple sequences (not related with down-stream analysis) and annotations and can be submitted through web (if the datasets are small) which does not requires any advanced tools.

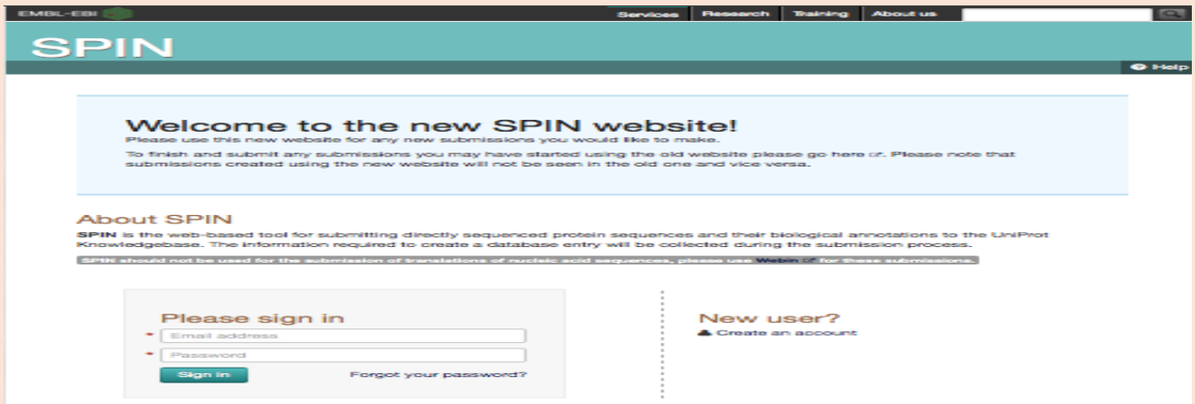
Sequin - For Complex sequences and annotations and is also good if we want to do some off-line submissions normally where we have our datasets which are huge ones and can be used in future with some advanced tools (for analysis) and graphical reports.



http://www.ncbi.nlm.nih.gov/WebSub/?tool=genbank

In the figures above, are the glances BankIt and Sequin webpages.

UniProt:



For protein sequences, just like NCBI tools, we have UniProt and the similar tool is called as**SPIN** which is a web-based tool for submitting directly sequenced protein sequences and biological annotations to the knowledgebase.

Shown in this figure, is the webpage of SPIN.

We can register here and then we can submit our data.

https://www.ebi.ac.uk/swissprot/Submissions/spin

Conclusion:

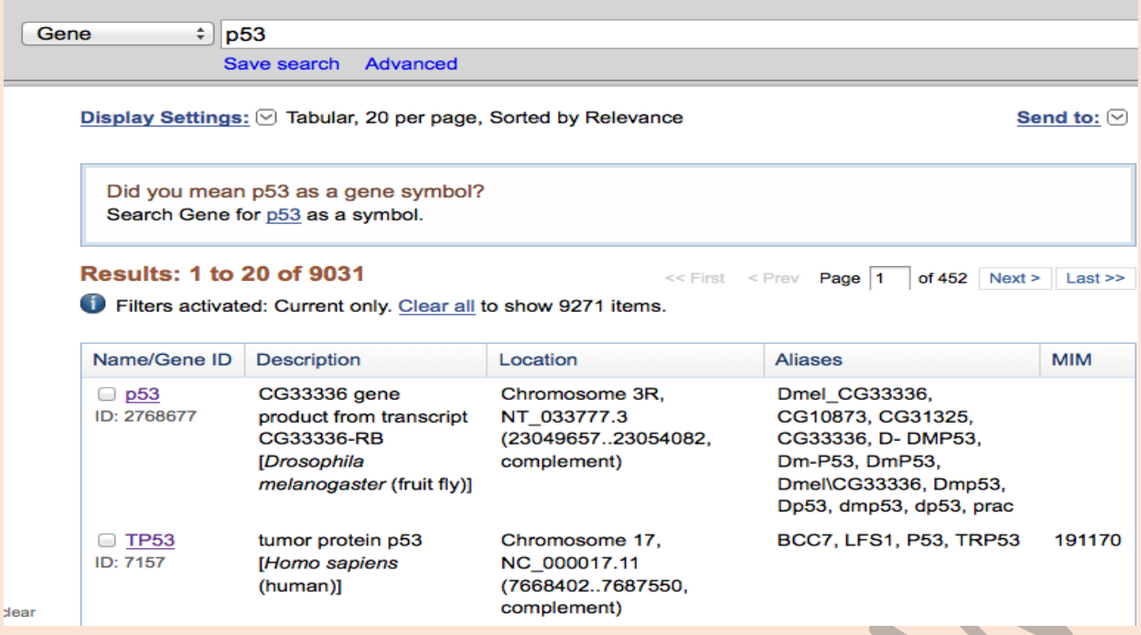
We conclude that sequences are stored in databases in specific format and when we want to submit them into a database then we need to follow the guidelines provided by those databases.

Topic #9 DNA Sequence Retrieval

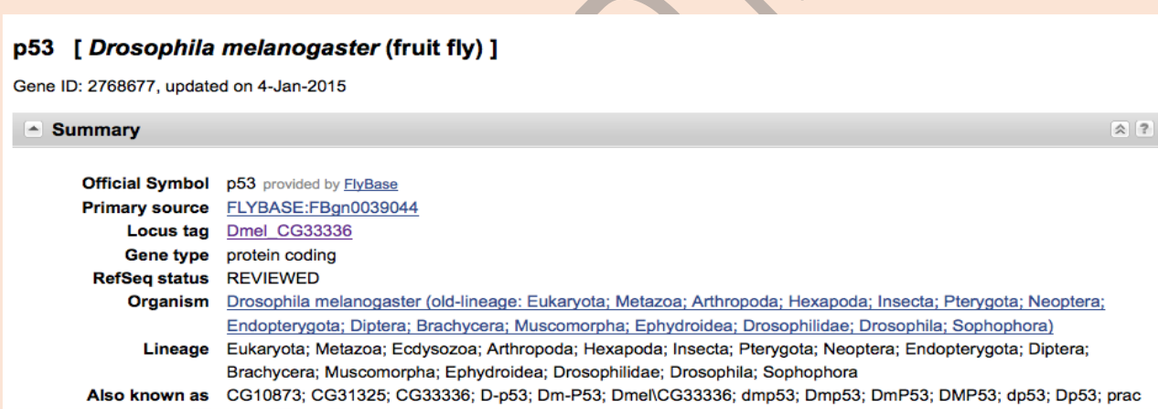
Introduction

Databases not merely collect and organize data (i.e. not only stores it) but allow intelligent data retrieval (we can do some down-stream analysis on those data sets). Let's see how we can get the DNA data from the NCBI.

MUHAMMAD IMRAN



So, here is the webpage of NCBI, for example you want to search for say p53 gene; tumour suppressor gene. We write p53 on the search bar, then we get then results, so here we can find many ID entries like 9000 entries are there, we are just looking into the first page in this we choose the first two. So let's click the first one, the p53 where the ID is 2768677, there is a description that what sort of gene is it, and its actually coming from *Drosophila melanogaster*, the location is Chromosome number 3 and we see some Aliases; the alternative names of this gene. The link to NCBI is http://www.ncbi.nlm.nih.gov/.

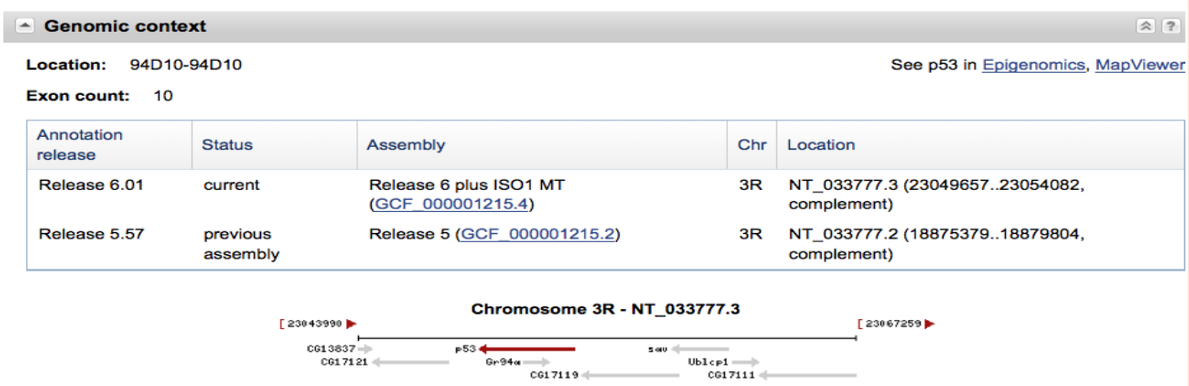


When we clicked on the first gene as shown in the figure above, we now come to this webpage which is a huge page that is portioned into different figures.

In this figure (on the left), we can see the **summary** of this gene.

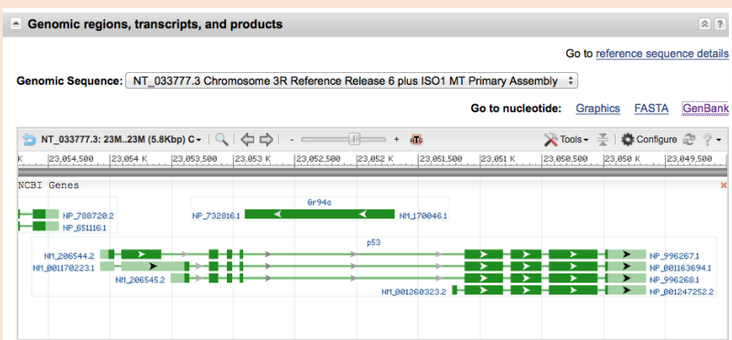
The official symbol is p53 provided by FlyBase which is also written in the *Primary source* (FlyBase is the databases that stores the genome of this fruit-fly *Drosophila*), then the *locus tag*, *gene type* is protein coding, RefSeq says reviewed (sometimes the genes are submitted and reviewed by some other scientist so it means that this gene has been REVIEWED). In the

organism section, we see the classification of that organism and the Aliases are written beneath



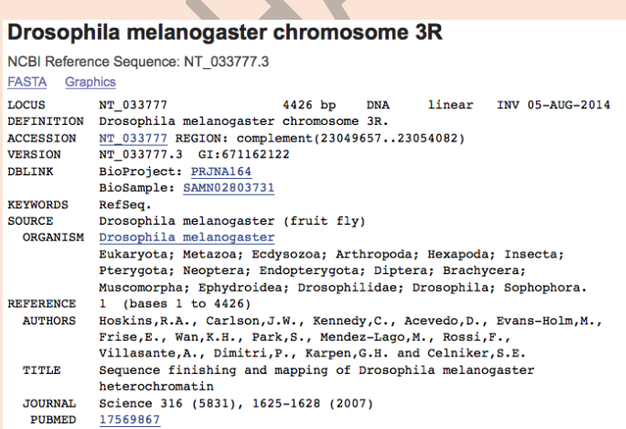
it.

In this figure, we can look into the structure of this gene and its coordinates (genomic coordinates), where we can see the location from where it is coming from, we can also see the orientations- the directions in which it is going (down below).



In this figure, we can see the genomic region, the transcripts and products tabs, we can look into the products of this gene (the gene when is expressed, the DNA is converted into the RNA). Since it's a eukaryotic genome where there is alternative splicing, so we can find different alternative splice variants of this gene.

On the upper right side of the figure, it is written as *Go to nucleotide, Graphics, FASTA and GeneBank* , so these are the different views with which we can get access to data files associated with this gene. When we click GeneBank, we are guided to another page, shown in the next figure.



Drosophila melanogaster chromosome 3R

NCBI Reference Sequence: NT_033777.3

FASTA Graphics

```
LOCUS       NT_033777               4426 bp    DNA     linear   INV 05-AUG-2014
DEFINITION  Drosophila melanogaster chromosome 3R.
ACCESSION   NT_033777 REGION: complement(23049657..23054082)
VERSION     NT_033777.3  GI:671162122
DBLINK      BioProject: PRJNA164
            BioSample: SAMN02803731
KEYWORDS    RefSeq.
SOURCE      Drosophila melanogaster (fruit fly)
  ORGANISM  Drosophila melanogaster
            Eukaryota; Metazoa; Ecdysozoa; Arthropoda; Hexapoda; Insecta;
            Pterygota; Neoptera; Endopterygota; Diptera; Brachycera;
            Muscomorpha; Ephydroidea; Drosophilidae; Drosophila; Sophophora.
REFERENCE   1  (bases 1 to 4426)
  AUTHORS   Hoskins,R.A., Carlson,J.W., Kennedy,C., Acevedo,D., Evans-Holm,M.,
            Frise,E., Wan,K.H., Park,S., Mendez-Lago,M., Rossi,F.,
            Villasante,A., Dimitri,P., Karpen,G.H. and Celniker,S.E.
  TITLE     Sequence finishing and mapping of Drosophila melanogaster
            heterochromatin
  JOURNAL   Science 316 (5831), 1625-1628 (2007)
   PUBMED   17569867
```

We can see the entry in GeneBank and how does it look.

Here, again we see the *name* of the gene, *locus* (where it's ID is written), length of the gene (it is 4426 BP), DNA, it is a linear type of DNA then we have the submission date.

Then the *definition* which is describing the organism's name, chromosome from which it is coming, then it has *accession* (the regions of the genome from which it is coming from), then we have the *version* (which is NT_033777.3, so there should have been .1 and .2 and since this is the third review, we can see .3 version here), we also see the reference (down below) and the authors from which this gene is coming and then their publications (it was seen to be published in Science).

```
FEATURES             Location/Qualifiers
     source          1..4426
                     /organism="Drosophila melanogaster"
                     /mol_type="genomic DNA"
                     /db_xref="taxon:7227"
                     /chromosome="3R"
                     /genotype="y[1]; Gr22b[1] Gr22d[1] cn[1] CG33964[R4.2]
                     bw[1] sp[1]; LysC[1] MstProx[1] GstD5[1] Rh6[1]"
     gene            1..4426
                     /gene="p53"
                     /locus_tag="Dmel_CG33336"
                     /gene_synonym="CG10873; CG31325; CG33336; D-p53; Dm-P53;
                     Dmel\CG33336; dmp53; Dmp53; DmP53; DMP53; dp53; Dp53;
                     prac"
                     /map="94D10-94D10"
                     /db_xref="FLYBASE:FBgn0039044"
                     /db_xref="GeneID:2768677"
     mRNA            join(1..118,178..501,884..964,1035..1071,1135..1161,
                     2959..3268,3333..3579,3642..4036,4096..4426)
                     /gene="p53"
                     /locus_tag="Dmel_CG33336"
                     /gene_synonym="CG10873; CG31325; CG33336; D-p53; Dm-P53;
                     Dmel\CG33336; dmp53; Dmp53; DmP53; DMP53; dp53; Dp53;
                     prac"
                     /product="p53, transcript variant B"
                     /note="p53-RB; Dmel\p53-RB; CG33336-RB; Dmel\CG33336-RB"
                     /transcript_id="NM_206544.2"
                     /db_xref="GI:281362333"
                     /db_xref="FLYBASE:FBtr0084360"
                     /db_xref="FLYBASE:FBgn0039044"
                     /db_xref="GeneID:2768677"
```

When we scrolled down, we can see in the figure on left, that there are features of this gene so the total length of the gene is 4426.

We can see mRNA (down below), and since it's a eukaryotic gene, so mRNA is coming from the exons and the regions from which it came are shown below with the word *join*.

```
CDS             join(75..118,178..501,884..964,1035..1071,1135..1161,
                2959..3268,3333..3579,3642..4036,4096..4118)
                /gene="p53"
                /locus_tag="Dmel_CG33336"
                /gene_synonym="CG10873; CG31325; CG33336; D-p53; Dm-P53;
                Dmel\CG33336; dmp53; Dmp53; DmP53; DMP53; dp53; Dp53;
                prac"
                /note="CG33336 gene product from transcript CG33336-RB;
                CG33336-PB; p53-PB; p53-like regulator of apoptosis and
                cell cycle; Dmp53; protein 53; drosophila p53"
                /codon_start=1
                /product="p53, isoform B"
                /protein_id="NP_996267.1"
                /db_xref="GI:45553461"
                /db_xref="FLYBASE:FBpp0083753"
                /db_xref="FLYBASE:FBgn0039044"
                /db_xref="GeneID:2768677"
                /translation="MSLHKSASFSLTFNQNTSIVSRSNSRTIFEAFKEFLDFWDIGNE
                VSAESAVRVSSNGAFNLPQSFGNESNEYAHLATPVDPAYGGNNTNNMMQFTNNLEILA
                NNNSDGNNKINACNKFVCHKGTDSEDDSTEVDIKEDIPKTVEVSGSELTTEPMAFLQG
                LNSGNLMQFSQQSVLREMMLQDIQIQANTLPKLENHNIGGYCFSMVLDEPPKSLWMYS
                IPLNKLYIRMNKAFNVDVQFKSKMPIQPLNLRVFLCFSNDVSAPVVRCQNHLSVEPLT
                ANNAKMRESLLRSENPNSVYCGNAQGKGISERFSVVVPLNMSRSVTRSGLTRQTLAFK
```

Then, within this mRNA we find the coding sequences (shown in the figure on the left), where coding sequences are the parts of the mRNA which are translated into the proteins so there are further sub-sets within those mRNA regions.

Down below, we see the translated version where we can see the word written as *translation*, and here we see the amino acid sequences coming from this gene.

```
ORIGIN
        1 cctggagcac ggaagattct tgcggacaca aatcgcaact gctaaataaa atttatttat
       61 ttgagtgcac agccatgagt cttcacaagt ccgcgtcgtt tagcttgact tttaaccagt
      121 gagcggagat attttattcg gtcttaccca acaaaataat gttgcgcctt tttgcagaaa
      181 cacttcgatt gtttcgcgta gcaatagtcg cacaattttt gaagctttca aggagttcct
      241 ggatttttgg gatatcggca acgaagtttc tgcagagtca gcagttcggg tctccagcaa
      301 cggagctttc aacttgccgc agagttttgg caacgaatcc aacgaatatg cccacctggc
      361 tacgcctgtg gatccagcct acggaggcaa caacacgaac aacatgatgc agttcacgaa
      421 caatctggaa attttggcca acaataattc cgatggcaat aacaaaatta atgcatgcaa
      481 caaattcgtc tgccacaagg ggtgagcaaa ttcaaaacac gcgctccaat cgataaacat
      541 tggctacggc gattgttcgc gctgcgtggc gaatggcaaa atccaaatag tcggtggcca
      601 ctacgattct gtagtttttt gttagcgaat ttttaatatt tagcctcctt ccccaacaag
      661 atcgcttgat cagatatagc cgactaagat gtatatatca cagccaatgt cgtggcacaa
      721 agaaaggtac agtgcggcaa caaattgatg atcgaacagt agaaaccttg catgtagcaa
          ----------------------------------------
     4261 ggcatgttcg atggccgaaa agaaaacatt tttatatttt tgatagtata ctgttgttaa
     4321 ctgcagttct atgtgactac gtaacttttg tctaccacaa caaacatact ctgtacaaaa
     4381 aagccaaaag tgaatttatt aaagagttgt catattttgc aaacat
//
```

In the end, till we reach the word called as *origin*, and here we can see the actual nucleotide sequences which are present starting from 1 until the last nucleotide and the sequence ends with a double slash sign (//).

Conclusions:

So, we conclude that DNA Sequences are stored in DNA sequence databases in specified formats and Genebank format is a standard format.

TOPIC no.10 Protein Sequence Retrieval

Protein Sequences:

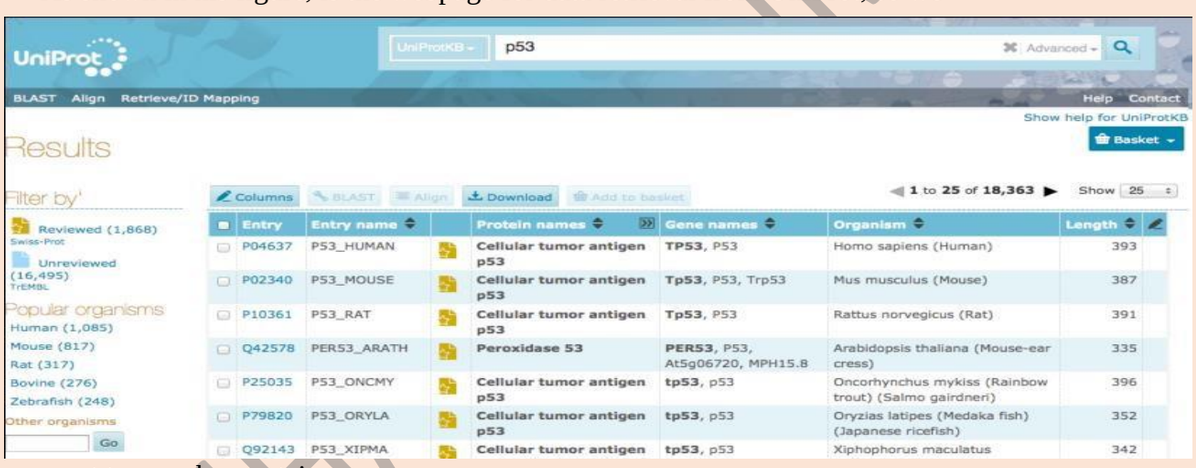
Now we talk about the data retrieval and first we'll talk about the protein sequence retrievals and structures. Protein data is of the following types:

- Actual sequences (from the proteomic data or some other experimental techniques) or translated sequences (sometimes, we go to nucleotides databases, we get those nucleotides and then we translate them by using some softwares, so these are kind of predicted protein sequences) .
- Structures (we can also make structures from those proteins that maybe predicted or the real structures coming from various X-ray Crystallography Techniques).
- Annotations (sometimes, we are interested in the functions of the proteins so those are stored as annotations).

UniProt (*It is an international partnership between PIR, EBI and SIB*):

Now as far as the resources are concerned, we have multiple resources for protein sequences but **UniProt**claims to be the biggest and integrated resource whereas for the structures **PDB** seems like a good resource.

As shown in his figure, is the webpage for data retrieval from UniProt, so we



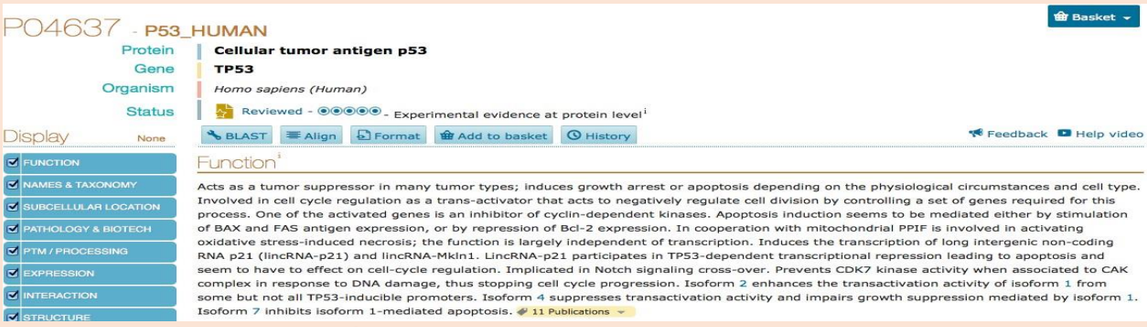
want to search a protein, say p53, where we put it into the search box and press enter which gives us the output. And we see that there are 18,000 different records and it is showing us the first 25 out of them.

We can have different columns for the output on this webpage so we can have *entry; it's ID, entry name* (the Suffix Human is written so it's coming from Human, it can be from mouse, rat and Arabidopsis), the *protein name* is Cellular tumour antigen, then *gene name* which is TP53 (where TP stands for Tumour Protein), the *organism* is obviously the human (here) and in the end we have it's length i.e. 393 bp (base pairs).

The link to this webpage is **http://www.uniprot.org/uniprot/.**

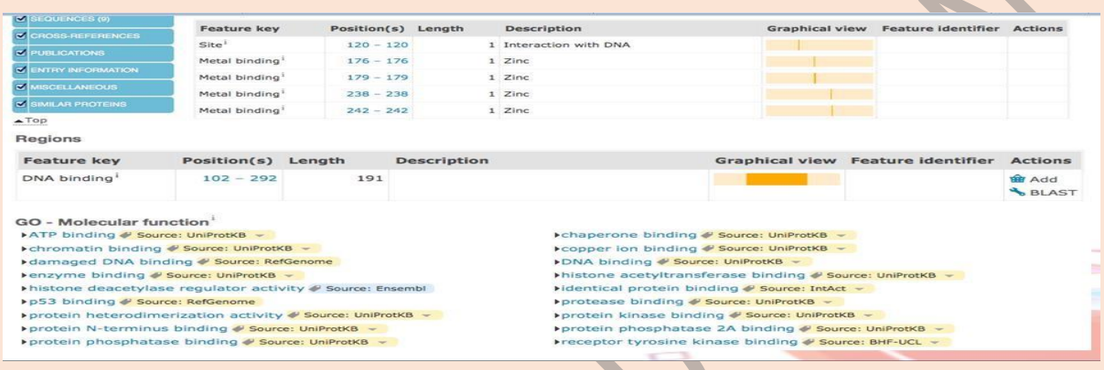
MUH. IMRAN

So, let's check the first one and here we reach on the record for this protein



(shown in the figure on the left) which is Cellular Tumour Antigen p53 protein, commonly known as TP53.

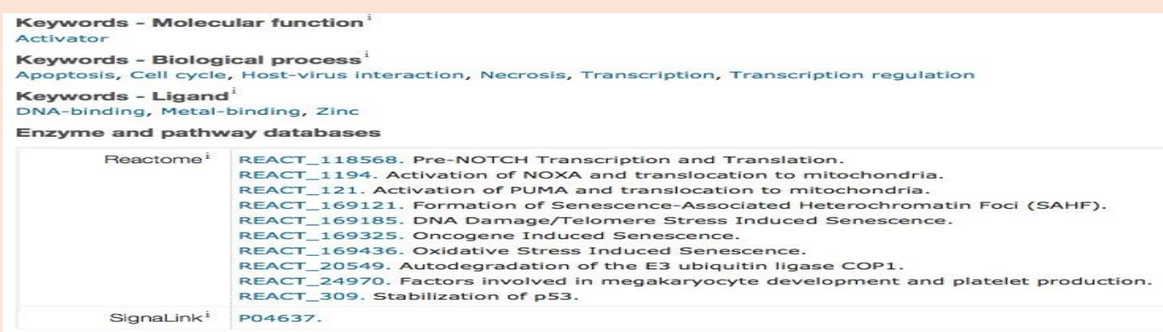
We can have different tabs, showing us the outputs. We can look into the *functions*, its *taxonomy*, and lot many other characteristics so if we look into the function so it gives us some description about what it's doing.



After scrolling the same webpage (shown in the figure on the left), we can see the *feature key* and in some *site* written (there are unique sites in different proteins which are having some specific properties in them so this is just one amino-acid present in this protein that *interacts with the DNA*). Similarly, there are different *metal binding sites* and we can see that it's mainly binding to the *Zinc*metal.The number of amino-acids is shown here so these are the regions where it interacts with the metal.

Down below, we can also see the *DNA binding* region, for example here, the amino acids are from 102 to 292 and that is also shown in the *Graphical view* as well.

GO-Molecular function or GO-Gene Ontologies, so gene ontologies are the different functional annotation term, there they define different functions, so amongst them we have molecular functions, biological processes, and we have cellular components. So here we just see a *Molecular function*, so it tells us that it performs the functions as shown in the figure , mainly it's a *ATP binding, it's p53 binding* with various other functions like *DNA binding.* So all those functions related to these proteins are present in the heading of GO-Molecular Function.



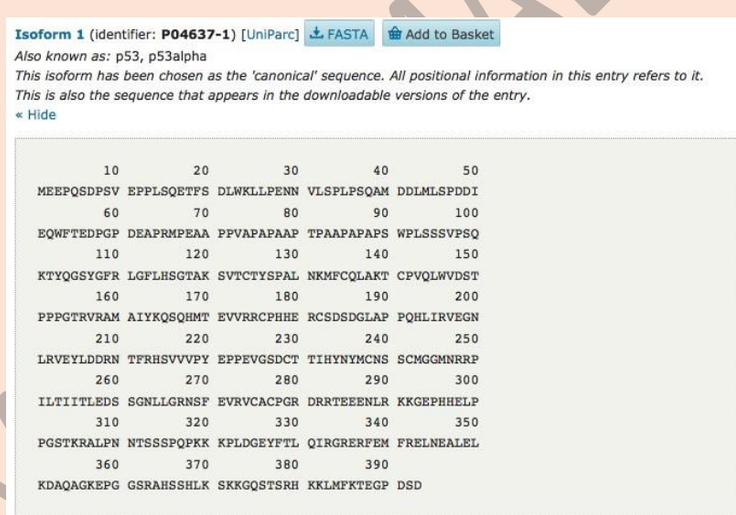
Next, we move on to some other functions, in the *Biological process* category (shown in the figure) we see that it is related to *Apoptosis* (which is a cell death and it is related to cell-cycle and some other components).

In the below section, we see some *enzymes and pathway databases*, and*Reactome*is a database in which we have a group of reactions which are categorized so these are the list of those reactions with which it is related.

When we move further (as shown in the figure on the left) till we reach its *Taxonomy.*

On the top, we can see something written as *Protein family* or *group databases* whichis TCDB. Basically, there is another classification in which the proteins are classified on the basis of being as transporter proteins so it is associated with the transportation across the membranes and there is 5-digit number, so there is a specific classification code which is given to each protein, and this protein has the specific code as shown in the figure.

So then we have the *names and taxonomies,* where there are *protein names,* and thetaxonomyof the individual can be seen in the *Taxonomic lineage* row. Let's see how we reach to its sequence and is shown in the figure below:



In this figure, we can see the sequence of the protein which is found to be at the end of the page.

Here, it says *Isoform 1*, so different proteins have different isoforms, different alternative splice variants so this is Isoform 1 as exhibited by its name which is P04637-1, and is the kind of first isoform. We can see the sequence of the protein and starts with a methionine (always a first amino acid in those proteins) and ending at 390TH amino acid. So, it's a 393 aa long protein and the sequence is right here. You can click on the FASTA button on the top and then you can get this output in FASTA format (we'll discuss it later).

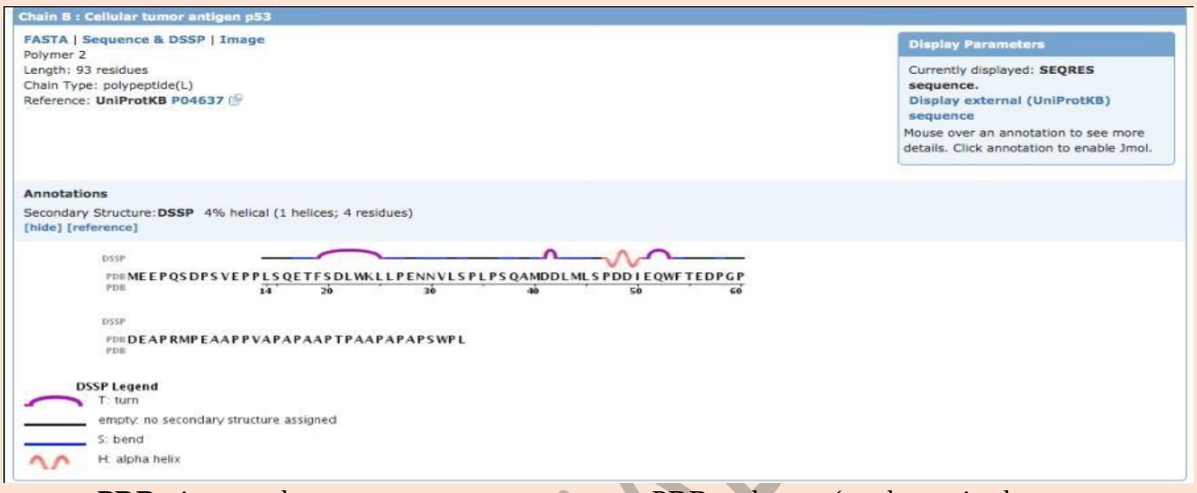
NCBI:

We can also get the same protein from NCBI (as shown in the figure on the left)

```
ORIGIN
        1 meepqsdpsv epplsqetfs dlwkllpenn vlsplpsqam ddlmlspddi eqwftedpgp
       61 deaprmpeaa ppvapapaap tpaapapaps wplsssvpsq ktyqgsygfr lgflhsgtak
      121 svtctyspal nkmfcqlakt cpvqlwvdst pppgtrvram aiykqsqhmt evvrrcphhe
      181 rcsdsdglap pqhlirvegn lrveylddrn tfrhsvvvpy eppevgsdct tihynymcns
      241 scmggmnrrp iltiitleds sgnllgrnsf evrvcacpgr drrteeenlr kkgephhelp
      301 pgstkralpn ntssspqpkk kpldgeyftl qirgrerfem frelnealel kdaqagkepg
      361 gsrahsshlk skkgqstsrh kklmfktegp dsd
//
```

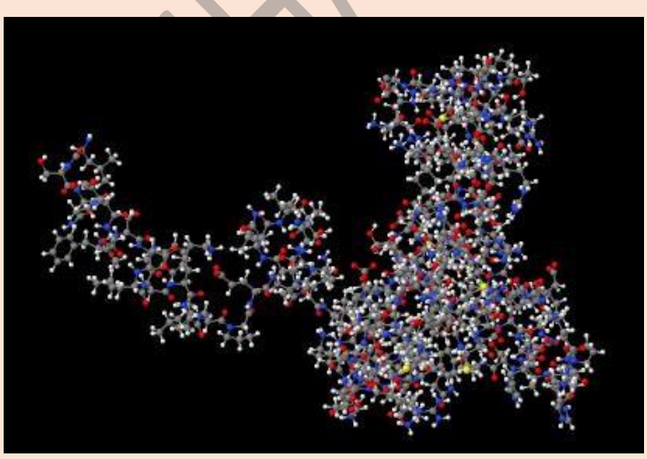
In NCBI, obviously the sequence is pretty similar and the arrangement is slightly different so it is *ORIGIN*, where the sequence starts and sequence ends at those two slashes (//). So, we can get the protein sequence from NCBI as well and the link to this website is http://www.ncbi.nlm.nih.gov/.

PDB:



PDB gives us the structures, so we can go to PDB webpage (as shown in the figure on the left) and search for the same ID i.e. P04637 and it gives us the sections or the regions from where it can make up some specific structures.

You can see the *turns* in Annotations section, the black ones are the empty lines where no secondary structure can be formed, blue ones show those bends and the orange ones are designated as alpha helices regions. So in PDB, we can have structures in this format as well as the 3D-Structures as shown in the figure below:



The link towards PDB is:
http://www.rcsb.org/pdb/explore/remediatedSequence.do?structureId=2LY4&bionumb er=1

Conclusions:

We conclude that:

- UniProt is the integrated resource between PIR, EBI and SIB and
- PDB is a good resource to get the protein structure.

TOPIC # 11 Sequence Formats

Sequences are stored in different formats in databases and since software requires those sequences in specific format so it's good to have an idea about what major formats are, we'll look into few of them.

FASTA Sequence Format

FASTA is the most recognized and well distributed format to present DNA and Protein sequences.

The sequence starts with a '*greater than*' sign (>), whereas the actual sequence is always on the next line. It is recommended that all lines of text should be shorter than 80 characters in length (generally we have 60 characters).

Example:

>gi|568815581:c7687550-7668402 Homo sapiens chromosome 17, GRCh38 Primary Assembly
GATGGGATTGGGGTTTTCCCCTCCCATGTGCTCATCTAGAGCCACCGTCCAGGGAG
CAGGTAGCTGCTGGGCTCTCCACGACGGTGACACGC--------
>gi|120407068|ref|NP_000537.3| cellular tumor antigen p53 isoform a [Homo sapiens]
MEEPQSDPSVEPPLSQETFSDLWKLLPENNVLSPLAPPVLGFLHSGTAKSVTCTYSPAL
NKMFCQLAKT--*
This sequence is of DNA in the *fasta* format (shown above), which starts with the '*greater than*' sign (>), same as in the case of the protein sequence in the *fasta* format (shown below) that also starts with the same symbol.

Then we have '*gi'* written which stands for 'gene identification' and the numbers shown are the 'ID' in both of the sequences.

In the DNA sequence, we have 'c' followed by the 'ID', this 'c' basically means the sequence is of the complementary strand and the regions from where it is coming are designated here; the base positions in between them, this gene is located. Then we have a short description of this gene that it belongs to '*Homo sapiens* ', 'chromosome 17' and the 'Primary Assembly' (assembly is where we get short sequence reads or small sequences and we put them together into a gene, known as assembly). Then finally, we have the actual sequence which is around 60 characters long in each line (as the sequence was quite long, we have used dashes to represent further characters).

In the protein sequence, we have 'ref' followed by the 'ID', which gives us an idea that it is a reference sequence (reference sequences are the curated sequence, there is a sub-section in NCBI called as *ref seq*, so they have reference sequences ; a kind of standard sequences to avoid any kind of redundancy. So, we can say these are the primary or the main sequences and we might have other alternative splice variants but references are the kind of true representative of the class). Followed by ref, we have another ID, which is the 'protein ID'. Then we have its brief description that it's a 'cellular tumor antigen' protein 'p53 isoform' and is also from '*Homo sapiens'*. Finally, we have the actual sequence of this protein and in the end we have dashes that represents it is an incomplete sequence and steric (*) is shown (sometimes the steric (*) is found to be seen in *fasta*files but sometime it don't, so the software must know what does this specific steric (*) stands for).

GeneBank Sequence Format:

GeneBank sequence format is found to be in the GeneBank Database which is a kind of standard format and other formats are pretty similar to it.
A sequence file in Gene Bank format can contain several sequences. One sequence starts with a line containing the word LOCUS and a number of annotation lines. The start of the sequence is marked by a line containing "ORIGIN" and the end of the sequence is marked by two slashes ("//").

MUHAMMAD IMRAN

Example:

```
LOCUS       AAU03518     237 bpDNA  PRI       04-FEB-1995
DEFINITION  Aspergillusawamori internal transcribed spacer 1 (ITS1) and
18S rRNA and 5.8S rRNA genes,  partial sequence.
ACCESSION U03518
BASE COUNT 41 a   77 c    67 g 52 t
ORIGIN
1 aacctgcggaaggatcattaccgagtgcgggtcctttgggcccaacctcccatccgtgtc
61 tattgtaccctgttgcttcggcgggcccgccgcttgtcggccgccgggggggcgcctctg
121 ccccccgggcccgtgcccgccggagaccccaacacgaacactgtctgaaagcgtgcagtc
181 tgagttgattgaatgcaatcagttaaaactttcaacaatggatctcttggttccggc
//
```

Here is the GeneBank format, which starts with the word '*LOCUS*', then we have it's 'ID', it is 237 base pairs long, we have some short description that it is a '*DNA*', '*PRI*' – primary sequence, submitted on '*04-FEB-1995*'.

Then we have a 'DEFINITION' line where we can have some description/explanation about this gene. Then again we have an 'ACESSION number'. It also provides us with the 'BASE COUNT' (i.e. how many A's (Adenines), G's (Guanines), C's (Cytocines), T's (Thymines) are there).

Then finally the word 'ORIGIN' tells us that the actual sequence is right here, we have these lines (60 bases on each line) that are separated into chunks of 10 bases and is a kind of standard practice. The sequences ends with the those slashes (//).

EMBL Format:
This format is similar to that of GeneBank Format. An example sequence in EMBL format is:

```
ID   AA03518    standard; DNA; FUN; 237 BP.
XX
AC   U03518;
XX
DE   Aspergillusawamori internal transcribed spacer 1 (ITS1) and 18S
DE   rRNA and 5.8S rRNA genes, partial sequence.
XX
SQ   Sequence 237 BP; 41 A; 77 C; 67 G; 52 T; 0 other;
aacctgcggaaggatcattaccgagtgcgggtcctttgggcccaacctcccatccgtgtc        60
tattgtaccctgttgcttcggcgggcccgccgcttgtcggccgccgggggggcgcctctg      120
ccccccgggcccgtgcccgccggagaccccaacacgaacactgtctgaaagcgtgcagtc  180
tgagttgattgaatgcaatcagttaaaactttcaacaatggatctcttggttccggc      237
//
```

Here, we have ID, accession number (AC), descriptions (DE), and the sequence actually starts from where the word 'SQ' is there, and we can observe that we have pretty similar lines as seen in the previous example. Finally, the sequence ends with doubles slashes same as in GeneBank format.

SwissProt Format:

SwissProt protein sequence format is similar to EMBL format but there is considerably more information about physical and biochemical properties of a protein (as you can see below there is more description).

```
ID    - Identification.
AC    - Accession number(s).
DT    - Date.
DE    - Description.
GN    - Gene name(s).
OS    - Organism species.
OG    - Organelle.
OC    - Organism classification.
RN    - Reference number.
RP    - Reference position.
```

MUHAMMAD IMRAN

RC - Reference comments.
RX - Reference cross-references.
RA - Reference authors.
RL - Reference location.
CC - Comments or notes.
DR - Database cross-references.
KW - Keywords.
FT - Feature table data.
SQ - Sequence header.
// - Termination line.

XML Format:

It is a modern practice in which we try to put those sequences in kind of a machine language. So, XML stands for Extensible Markup Language. The format is similar to HTML (language for Web programming).
The good part is that this language is in between machine and man readable so it's kind of easy to code over this.
And it is becoming standard data format for transferring genome data.

Example:

```
<xsd:annotation>
<xsd:documentation>
        XML Schema for SBOL core data model compatible with RDF/XML serialization.
<dc:date>2012-01-19</dc:date>
<dc:creator>EvrenSirin</dc:creator>
<dc:contributor>Michal Galdzicki</dc:contributor>
</xsd:documentation>
</xsd:annotation>
```

This format seems pretty weird but not for the people with computer science background.**NBRF Format:**

```
>DL;seq1
seq1, 16 bases, 2688 checksum.
agctagctagctagct*

>DL;seq2 seq2,
16 bases, 25C8 checksum.
aactaactaactaact*
```

The format is pretty similar to fasta but in addition to that it gives us the checksum value (checksum- we take those nucleotides and since we know that in computers every digit is related to some 'ascii' value, we can take those values and add them up together and then we can come up with this number known as checksum. So, it's a good thing to have this number as when somebody is downloading the sequence, he can again check on his computer and find the checksum, if they are equivalent to one another, the sequences are correctly downloaded otherwise there must be some issues with the downloading)

GCG FORMAT:

GCG stands for Genetics Computer Group (basically it was a group of scientists who were helping the biological community to develop different software and training programs to help with the biological sequence analysis problems, so they also came up with the sequence formats). This format is kind of similar to the NBRF format (we have checksum but we don't have greater than (>) sign as in fasta, we have length of the sequence). There can be multiple sequences in one file.

Example:

```
seq1 seq1 Length: 16 Check: 9864 ..  1 agctagctagctagct   seq2 seq2 Length: 16 Check: 9672 ..
1 aactaactaactaact
```

MUHAMMAD IMRAN

Sequence converters:

Sometimes, we need to convert between sequences so you can come up with your own script or you can come up with your own codes and there are also some programs meant for this purpose alone such as READSEQ is a useful sequence converter (developed by D.G.Gilbert at Indiana University, USA) basically it recognizes DNA or Protein sequence file and interconvert them between different formats.

Conclusions:

What we conclude in the end of this lecture is the following:

- Databases store sequences in specified formats
- Genebank, DDBJ and EMBL has similar formats
- Different software need sequences in different formats

We might convert the sequences into other formats on our own or we can also simply use one of the programs available for converting like READSEQ

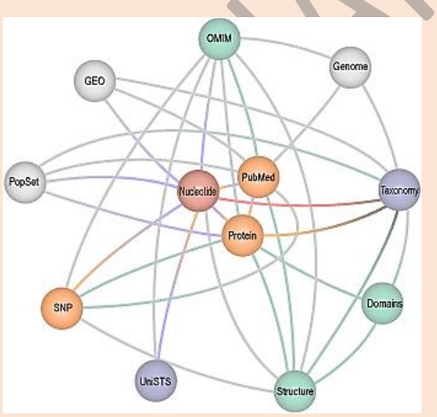
Topic -12 Data Retrieval

Data Retrieval:

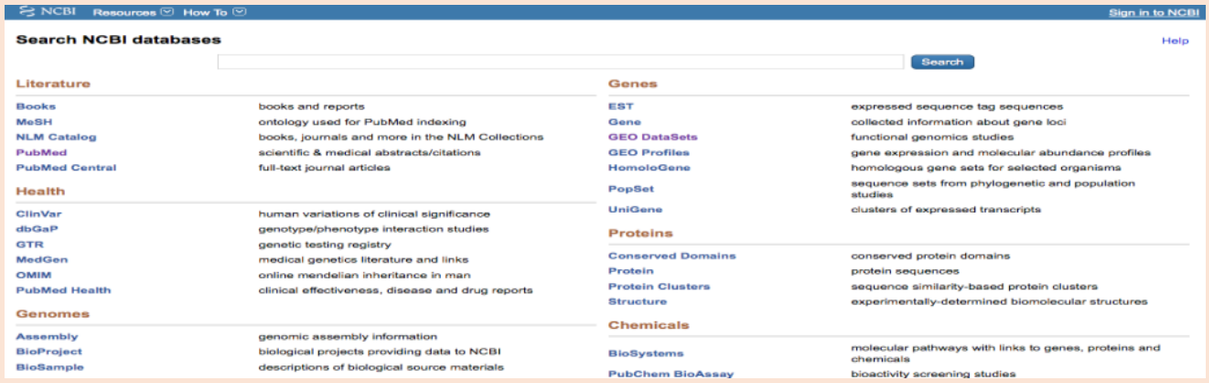
Nearly all biological databases are available for download as simple text (flat) files. Sometimes we are interested to download the database and do the analysis locally in our own machines which might save our time as the local version of the database allows one greater freedom in processing the data.

ENTREZ:

It is an integrated search engine that works behind NCBI, so you can do lot of researches and can look for variety of data using it (It can be accessed from the site www.ncbi.nlm.nih.gov/Entrez/). It integrates PubMed and 39 other scientific literatures, nucleotide and protein databases. For example, it can be **protein domain data**, **population studies**, **expression data**, **pathways**, **genome details** and **taxonomic information.**



Here, we can see it integrates between GEO (gene expression sets), OMIM (Online Mendelian Inheritance in Man), Genome Databases, taxonomy Databases, etc. And we can see that in the middle we have Nucleotide, PubMed and Protein. So it is an integrated system which operates between different databases, so you can simply search for whatever you are looking after and ENTREZ will search it for you.



Here, is the page of ENTREZ that allows you to search anything by the help of a search bar at the top. It has different connections like we have Literature resources, we have Health Databases, Genomes, different Genes Databases, Proteins and Chemicals.

Bulk Data Retrieval:

Sometimes, we need to obtain data in bulk amount and for this purpose normally we use Linux but for Windows users, there are some packages or programs available and are known as FTP clients so the best option is to use FTP (File transfer protocol). The File Transfer Protocol (**FTP**) is a standard network protocol used to transfer files Via command line or application programs like FTP clients (we'll be using it).

Once, we get the data which is mostly not in a proper format and every other software require different specific formats so we might want to use some programming languages to help convert the data into the required format. The programming languages like PERL and Python are good for processing Biological data in Bioinformatics.

Conclusions:

We have learned that :

- Data is transferred over the internet.
- Data needs to be transformed or processed before handing it over to any software.

Topic # 13 Genome Informatics

Now we talk about the main subject which we are seeking in this course that is the ***Genome Informatics*** and let's look into it.

Genome Informatics:

It is about the Genome sequencing that provides the sequences of all the genes of an organism. The major application of Bioinformatics is the analysis of full genomes that have been sequenced. Whereas the challenge is to identify those particular genes that are predicted to have a specific biological function.

Genomics definition:

NHGRI (**N**ational **H**uman **G**enome **R**esearch **I**nstitute) defines Genomics as:

"Study of all of a person's genes (the Genome), including interactions of those genes with each other and with the person's environment."

So, ***Genome Informatics*** can be defined as:

It is the field in which computational and statistical techniques are applied to derive biological information from genome sequences.

MUHAMMAD IMRAN

"Genome informatics includes methods to analyze DNA sequence information and to predict protein sequence and structure."
(Iossifov, et al. 2014)

Genome Sequences:

So, after we have the genome sequences, we do the analysis of those sequences and it includes the following:

- The discovery and utilization of sequence polymorphisms (different sequence vary from one another, so we can identify those polymorphisms).
- Opportunity to explore genetic variability both between and within the organisms (we can help identify different traits of those organisms by using those polymorphisms).

Genome Analysis:

In Genome Analysis we mainly perform the following tasks;

Sequencing

- ***Assembly*** (since the sequencing is done in a way that whole genome is broken down into short fragments and once those fragments are sequenced, we need to put them together, this step in genome analysis is known as Assembly).
- ***Repeat identification and masking out*** (once we assemble that genome, we try to find out the regions in which we have large number of repeats because assemblies jumble up where we have those repeats so we need to find those regions and it is one of the important task to go and look into those assemblies while keeping in mind those regions in which we have those repeats).
- ***Gene prediction*** (after we have assembled a finished genome, now we can go for the prediction of the genes where we can find the genes by using different patterns or features of those genes).
- ***Looking for EST (Expressed Sequenced Tags) and cDNA (complementary DNA) sequences.***
  (EST and cDNA are basically originated from the DNA where the genes that are expressed are transcribed into mRNA which is then reversed transcribed back into cDNA. So by the help of cDNA, we can look into where those expressing regions are present in the genes that will give us the idea of the gene expression or the regions from where the mRNAs are made).
- ***Genome annotation*** (in which we can find out similar functions performed by different genes)
- ***Expression analysis*** (once we have the idea about the regions of the gene in which we can have the gene expression then we can explore the quantification i.e. how much those genes are being expressed).
- ***Metabolic pathways and regulation studies*** (once those genes are expressed, their products interact with each-other and then they perform different metabolic roles in the shape of different metabolic pathways and networks).
- ***Functional genomics*** (where we are actually looking into the different functions performed by different regions of the genome that are under the control of different genes and what exactly would be the effect of changes in those genes specifically if we want to study about the genes related to diseases).
- ***Gene location/gene map identification*** (map the location of those genes on the chromosomes).
- ***Comparative genomics*** (in which we can take one genome and compare it with another genome, where we can find the comparative features; what is present in the first genome and not in the second one and the intersections between them, etc.).

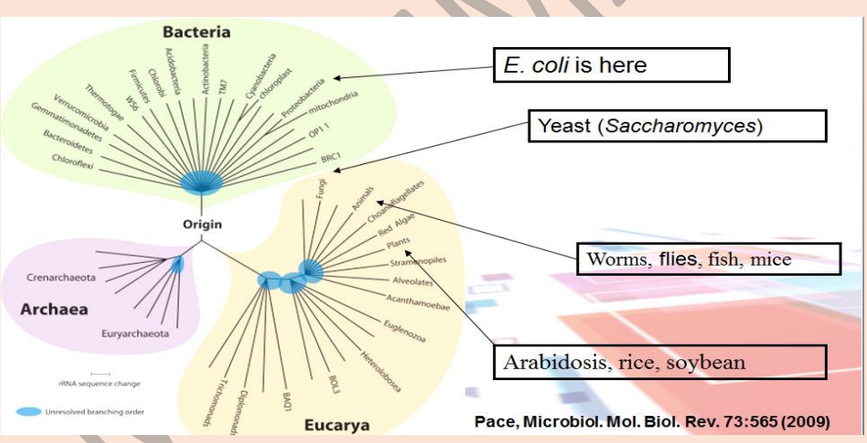
MUHAMMAD IMRAN

- ***Identify clusters of functionally related genes*** (those genes they might be having similar structures, sequences and also performs similar functions, which can give us the idea about the evolution).
- ***Evolutionary modeling*** (so the identification of the clusters of functionally related gene can help us in making an evolutionary model).
- ***Self-comparison of proteome*** (sometimes we are interested in finding genes which are kind of duplicated within the same organism, so in order to do that this self-comparison of proteome is made, where proteome is the collection of the proteins which are derived from those genomes. Therefore, the whole collection of one organism's proteins can be termed as proteome and we can compare it with itself and can find about those sequences which are being duplicated in it).

- **Model organisms:**

Most of the times, while we are doing those genome sequencing projects, our objective is to find the cure of some disease, or improving some variety of the crop for enhancing its production, or looking into some drugs against different organisms so it's a good idea to have some model organisms that can be used for studying various processes in labs and there are is a range of model organisms which includes:

- ***E. coli – bacteria***
- ***S. cerevisiae – yeast***
- ***C. elegans – worm***
- ***D.melanogaster – fly***
- ***Daniorerio – zebrafish***
- ***Musmusculus - mouse***
- ***Homo sapiens – you and me***
- ***Arabidopsis - plant***



Here, in this diagram we see auniversal tree of life that has been made with the structures of small ribosomal RNA unit. It divides whole living organisms into three groups, we have *Bacteria* at the top, we have *Archaea*(they are special organisms that lives under hard conditions) and then we have Eucarya (which is obviously the biggest among all groups). We pick those model organisms from important branches of this tree of life so for example, E-coli is shown, yeast as an example of fungi, from animals we have worms, flies, fish, mice, and Arabidopsis, rice, soybean are the examples from plants. So, we try to get these organisms; best representatives from different classes from important branches on this tree of life.

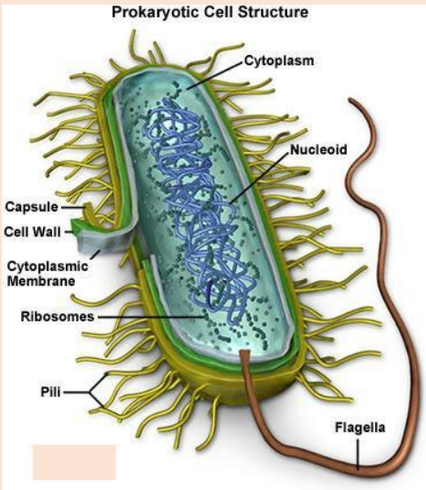
Conclusions:

We conclude the following:

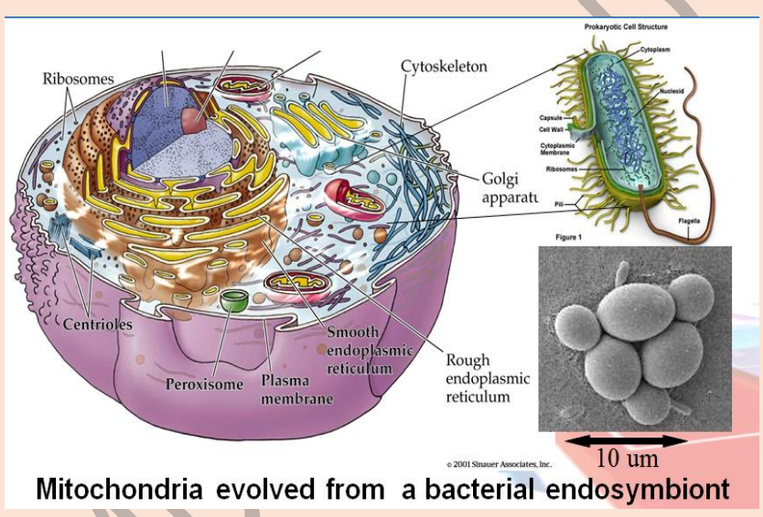
- Sequencing and analysis of full genomes paves the way for future discoveries
- Different model organisms are best source to explore our Genome and to interpolate the results towards the higher organisms.

Topic -14 Prokaryotic Genome

Now, we study the ***prokaryotic genome***, prokaryotes are the organisms whose Genetic material (DNA) is not enclosed in a nuclear membrane, so there is no nucleus in them. As there is no nucleus in prokaryotes, there is no justification to have other membrane bound organelles. These are relatively simple cells.



Here, in this diagram we see a prokaryotic cell which is a bacterium (here). We have a genome (DNA) in the shape of a big chromosome in the middle, and ribosomes (small structures important for protein synthesis that occurs in every other organism so ribosomes can also be seen here). It's relatively simple cell, having cell wall with different layers.



Here, in this diagram we see a comparison between a eukaryotic cell and a prokaryotic cell. We can clearly see the membrane bounded organelles in the eukaryotic cell, like mitochondria (involved with the respiration process; food is broken down into the energy. There is a hypothesis that mitochondria actually evolved from bacteria and is known as endosymbiont hypothesis). Here we can also see the difference in the size of both cells, so eukaryotic cells are complex and bigger than prokaryotes.

The first prokaryotic genome sequenced was that of *Hemophilus influenza* (we have seen in the previous section) and this organism was sequenced in a relatively moderate cost and with an efficient pace that paved the way for sequencing of many other organisms. So study of those prokaryotic organisms is important.

Selection Criteria:

There are different criteria for the selections of the organisms which are then send to the genome sequencing projects. Following are the criteria for selection:

- They had been subjected to a detailed biological analysis/ extensive studies and thus were model organisms.
- They might be important human pathogens (so it's important for us to study its genome).
- They were of phylogenetic interest.
- Sequences were annotated as they were sequenced.

Table 10.2. *Features of representative prokaryotic genomes*

Organism (reference)	Phylogenetic group	Genome size (Mbp) (no. protein-encoding genes)	Novel functions
Escherichia coli (Blattner et al. 1997)	Bacteria	4.6 (4288)	model organism
Methanococcus jannaschii (Bult et al. 1996)	Archaea	1.66 (1682)[a]	grows at high temperature and pressure and produces methane
Hemophilus influenzae (Fleischmann et al. 1995)	Bacteria	1.83 (1743)	human pathogen
Mycoplasma pneumoniae (Himmelreich et al. 1996)	Bacteria	0.82 (676)	human pathogen that grows inside cells; metabolically weak
Bacillus subtilis (Kunst et al. 1997)	Bacteria	4.2 (4098)	model organism
Aquifex aeolicus (Deckert et al. 1998)	Bacteria	1.55 (1512)[b]	ancient species, grows at high temperature and can grow in a hydrogen, oxygen, carbon dioxide atmosphere in the presence of only mineral salts
Synechocystis sp. (Kaneko et al. 1996a,b)	Bacteria	3.57 (3168)	ancient organism that produces oxygen by light-harvesting; may have oxygenated atmosphere

Mount

Here, in this table, we see different prokaryotic organisms; representative prokaryotic organisms and their genomes.

E-coli that was sequenced by Blattner et al, the phylogenetic group is bacteria, genome size is 4.6 Mbp (4288 protein encoding genes), and the Novel functions or description of E-coli is that it is a model organism.

In phylogenetic section, we just have one Archaea whereas rest of them are bacteria.

Methanococcus is an archaea with genome size of 1.66 Mbp (1682 protein encoding genes) and it grows at high temperature and pressure and produces methane (maybe a good source to have natural gas from it). *Hemophilus*is a bacterium with genome size of 1.83 Mbp (1743 protein encoding genes) and is a human pathogen. *Mycoplasma* is another bacterium with genome size of 0.82 Mbp (676 protein encoding genes) and is also a human pathogen that grown inside cells; metabolically weak. In the end, we have *Synechocystis*which is again a bacterium with 3.57 Mbp (3168 protein encoding genes) genome size and is an ancient organism that produces oxygen by light-harvesting; may have oxygenated atmosphere.

Conclusions:

We conclude that:

- Prokaryotes are simple Genomes.
- They are easy models to study Biochemistry, physiology and Molecular biology of life processes.
- Sequencing is done on economically important organisms (i.e. first it's implemented on simpler genome which is then used to explore complex genomes).

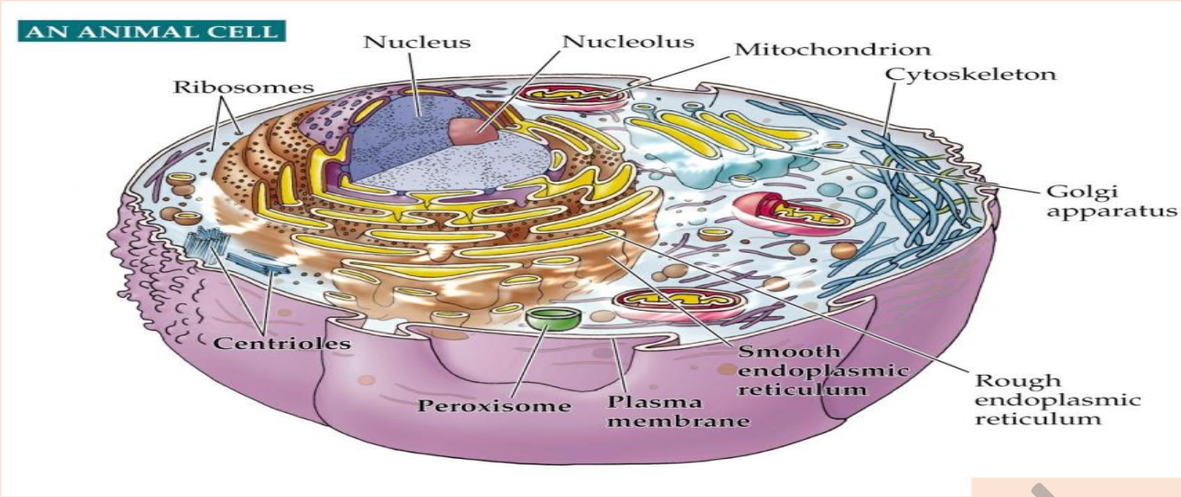
Topic -15 Eukaryotic Genome

Introduction:

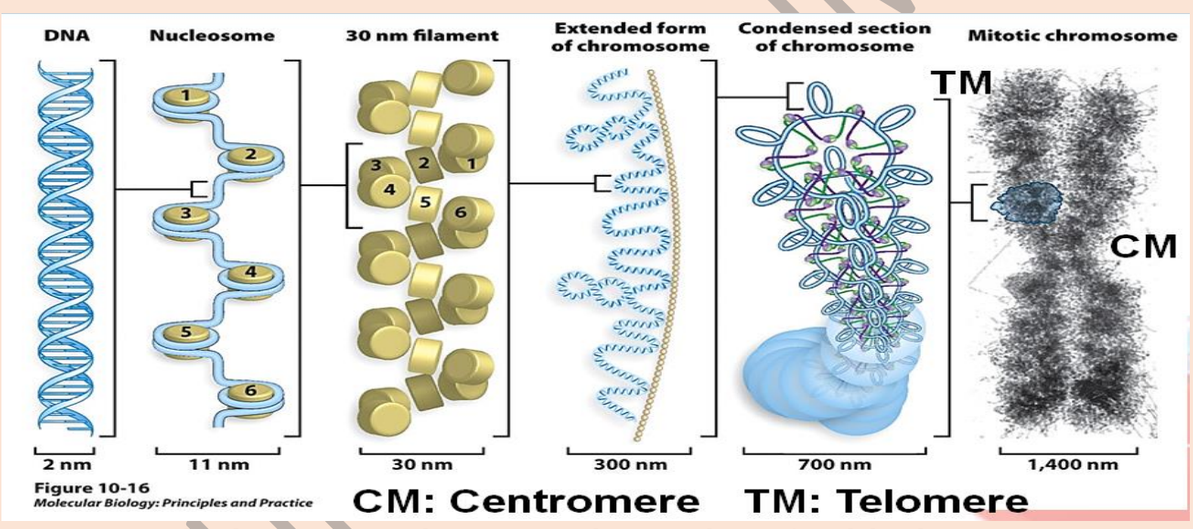
We have seen that prokaryotes are simple genomes in comparison to eukaryotes which are relatively complicated. So distinct properties of eukaryotes are mentioned below:

- Eukaryotes have larger genomes
- Have tandem repeats
- Have introns in their protein-coding genes (introns are between the exons which are the protein coding regions within the genes).
- Heterochromatin and euchromatin region (eukaryotes have complicated genome, so the chromosome is grouped as heterochromatin; densely packed region and euchromatin; lightly packed region).

MUHAMMAD IMRAN



Here, in this diagram we can see a typical eukaryotic cell which is pretty stuffed as compare to prokaryotic one. We have nucleus in the middle, channels coming out of the nucleus known as *endoplasmic reticulum* (helps in transportation), ribosomes (for protein synthesis), mitochondria (energy synthesis), we can also see the cytoskeleton that makes the structure of this cell intact and Golgi apparatus (are concerned with the secretion). So complicated membrane bounded organelles are present in the eukaryotes



Here, in this diagram we see the connection between the DNA and the chromosomes.

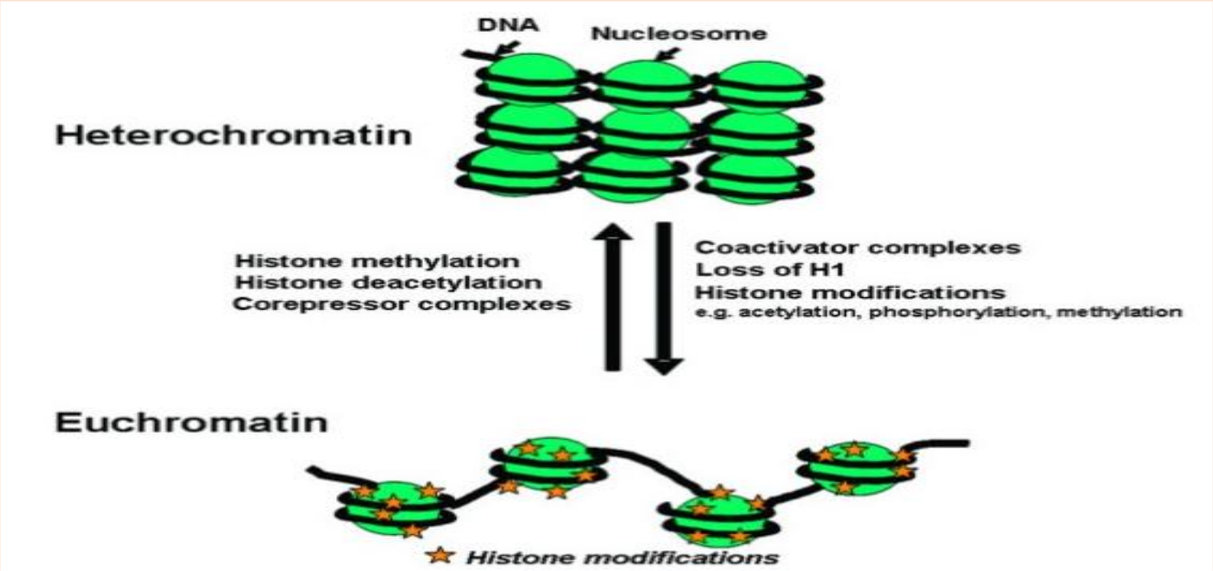
On the left-hand side, we see a DNA strand that is a 2nm wide strand. So, the DNA wraps over the protein complex molecules (histones - labelled as 1, 2, 3...), and this structure is known as *nucleosome*. Then these histones, turn around and makes a wider structure and it makes a 3nm filament (in third section). Then these nucleosome structures supercoil on their selves to make those further bigger fibres and until they reach the chromosome the width is 1,400 nm. So, if we look into the chromosome, we can recognize that there are different arms in it, which are known as sister *chromatids* (remember this is just one chromosome but we have two chromatids), somewhere in the middle we see a constricted part known as *centromere*, whereas the terminals are known as *telomeres*, (remember these nomenclature while we are discussing the heterochromatin and euchromatin parts).

Staining with dyes:

So chromosomes if stained with the dyes, they give different coloring patterns, we can come up with the following:

- Dense heterochromatin (dense regions obviously take more color)
- Light Euchromatin (light regions take less color)

If we look into the gene expression, the *heterochromatic regions* are packed so the enzymatic machinery cannot reach there; hence these regions are poorly transcribed (expressed). Whereas, the *euchromatic regions* are highly expressed because they are loosely packed and the enzymatic machinery can easily reach to them.



Here, is the diagram in which we can see the relationship between heterochromatin and euchromatin. You can look into these nucleosomes (combination of DNA and histones), which are quite jammed packed with one another, so definetly the enzymes cannot access the DNA which is embedded inbetween. There are different modifications on the DNA or histones that bring about those structures (we can see on the top), so for example there are histone methylations (in which methyl groups are added to those histones) in that case the system moves towards downside (as shown in the diagram) so it becomes *euchromatin* and similarly, we see that there are some other methylations on some other aminoacids that can move back into the opposite direction also. So, histone methylation, histone deacetylation and there are some other complex proteins which gets attached and give us this *heterochromatin region* and in the reverse process, we get the *euchromatin region*. In the Euchromatin region, the histones are quite spaced and DNA can be acessible. So, this is the reason why the euchromatin region is expressed more as compared to the heterochromatin region.

Conclusions:

We conclude that:

- Eukaryotes are distinguished by the presence of prominent nuclei
- Eukaryotes have larger genomes, tandem repeats and introns in their protein-coding genes (i.e. they are complicated).

Topic # 16 Epichromosomal Elements (EEs)

Introduction:

Genome is the total collection of genetic material and is made up of

- Chromosomes
- Epichromosomal Elements

Prokaryotic EEs:

1. Plasmids
2. Self-replicating
3. Additional rings
4. Bacteriophages
5. Host colonization

6. Transposons
7. Parasitic DNA elements

Eukaryotic EEs:

Eukaryotes have extra organelles that contain the genome (DNA) which we call it as ***Organellar DNA.***
Examples are:

Mitochondrial DNA (both in animals and plants), **Chloroplasts DNA** (in plants), these are membrane-bound organelles and they may be present in hundreds to thousands of copies (so there is also multiple copies of these genomes). Mitochondrion is the site for respiration whereas chloroplast is the site for photosynthesis. Their DNA's can be labeled as mtDNA or cpDNA respectively.

Plasmids, yeast, Transposons, Viral genomes and retroviruses are other examples of organellar DNA.

Endosymbiont hypothesis:

How do these organelles evolved?
So there is a hypothesis known as ***Endosymbiont hypothesis.*** According to this hypothesis, these organelles originated as separate prokaryotic organisms that were taken inside a primordial eukaryotic cell. Such symbiotic relationships in which two species are dependent upon one another to varying extents served as crucial elements of the evolutionary progression of eukaryotic cells.

This hypothesis was originally proposed in 1883 by ***Andreas Schimper***, but extended by ***Lynn Margulis*** in the 1980s.

So according to this theory, ***Mitochondria*** and ***Chloroplast***are derived from endosymbiotic bacteria (that got incorporated into the cells).

Organelle Genome:

Organelle genome (of mtDNA/cell or cpDNA/cell) features are as following:

- Circular
- Double stranded
- Supercoiled
- No histones
- Multiple copies

Genome	Size & Organization
Plant plastid	150 kb circle
Plant mitochondria	150 – 2000 kb multipartite
Human mitochondria	17 kb circle
Saccharomyces mitochondria	75 kb circle

Here, in this table we can see the size of these genomes. For example, *plant genome* is 150kb circular genome, *plant mitochondria* is 150-2000kb multipartite, *human mitochondria* is 17kb circular and *saccharomyces mitochondria* is 75kb circular.

Mostly the genomes are circular.

As far as the expression of these organelle genomes is concerned, it has been observed that their functions are actually dependent on nuclear genomes (they cannot make functions for themselves).

MUHAMMAD IMRAN

They encode only a subset of genes required to elaborate a functional organelle like rRNAs, tRNAs, ribosomal proteins, membrane-associated respiratory or photosynthetic components.

Other components which are encoded by nuclear genome are translated in the cytosol of the cell and are imported into the organelle. It has been observed that 10% of nuclear genes are devoted to mitochondrial function whereas 15% to plastid function.

Conclusions:

We conclude the following:

- Organelle genome is similar to prokaryotes.
- It is in high copy number.
- The mtDNA and cpDNA depends on Nuclear DNA (genome) for their function.

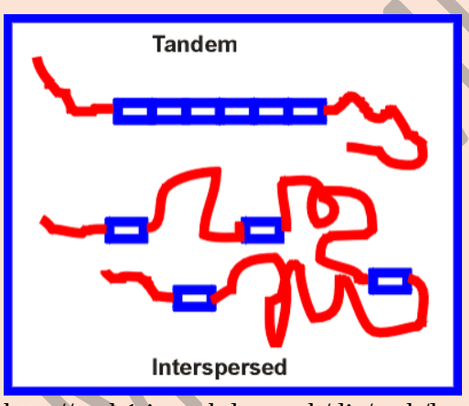
Topic # 17 Genome Repeats

We have seen that the major proportion of genomes in the eukaryotes is made up of repeats (they are also present in prokaryotes). So let's look into what these repeats actually are.

Sequence Repeats:

These repeats *skew the base composition* (normally the A's, T's, G's and C's relative proportion is similar to one another but if repeats are present and these repeats are of same types, for example if we have runs of GC's, then obviously they'll change the proportion of different bases) which can contribute to having differences in there buoyant densities (so those fragments can then be separated on the basis of those differential densities).

The repeat containing DNA can be separated as **satellite DNA** on the bases of these densities.



Here, is an example in which we can see the repeats, they can be *Tandem repeats* (array of repeats together), *interspersed repeats* (those repeats which are separated by normal genome followed by repeats i.e. normal genome is between those repeats).

The link to this figure is:

http://mcb1.ims.abdn.ac.uk/djs/web/lectures/repeats1.html#anchor10305

Satellite DNA:

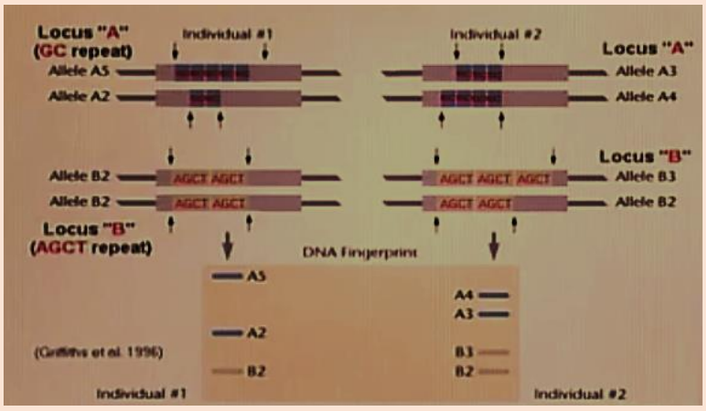
Satellite DNA has following features:

- It may be one to several thousand bp long and it can also be present as *Tandem;* array of 100 million bases long.
- They are present near centromere and telomere and
- They can be classified as *Mini-satellite* and *Micro-satellite.*

Mini-satellite:

Mini-satellite features are as follows:

- They are 15 bases long in array of several hundred to thousands kb.
- They are typically present in euchromatin region.
- Example is VNTR and is used to identify human individuals in forensics.



Here, in this figure we can have different repeat patterns. In the individual #1, we have those GC repeats (runs of GC's together), and obviously the organisms are diploid i.e. they are having two chromosomes, from their parents. So, the same *Allele A2* has two of those repeats, if we look into another *locus* (gene position), we can see there is *Allele B2* that has AGCT repeat (2 copies in both allele) and in the second individual we have 3 copies in 1st allele and 2 copies in second allele. So, when we digest them with restriction enzyme (restriction enzyme cut them on the repeat regions), we can have *differential banding patterns* when we run them on gel. In this way, we can recognize those individuals. In the figure (below), we can see the bands, on the left-hand side are the bands of the individual#1 and on the right-hand side are the bands of the individual#2. In this way, we are getting those repeat regions and by running them on the gel, we can detect which DNA belongs to which organism.

Microsatellite:

Micro-satellite features are as follows:

1. They are 2-6 bases long and can be in arrays of 10-100 bases.
2. They are inherited to offspring.
3. Mainly they are useful markers for genetic analysis and evolutionary studies (just like in previous example)
4. They are found in telomeres TTAGGG (one example of six nucleotides segment).
5. SSRs and STRs (**S**imple **S**egment **R**epeats and **S**hort **T**andem **R**epeats are typical examples).

Transposable Elements (TEs):

These are also important kind of repeats and their features are as follows:

1. They are making up the larger proportion of the eukaryotic genome.
2. They thought to play an important role in the evolution of these genomes.
3. They move (Jump) from one location to another even faster than chromosome replicate (and are known as jumping genes).
4. They have a potential to increase in number,
5. So they make up a large proportion of eukaryotic genome.
6. They are detectable but sometimes they blend into the genome due to mutations and cannot be detected.

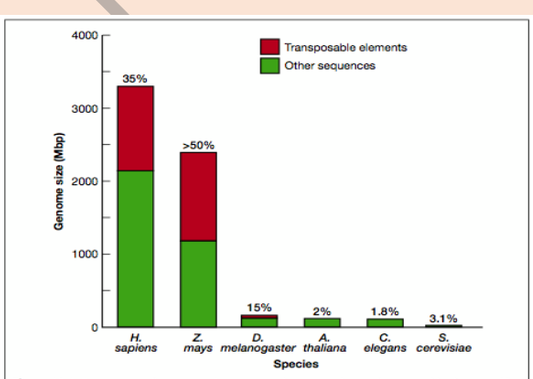


Figure 10.1. Percentages of representative genomes that are made up of transposable elements. The genomes include those of humans, maize, the fruit fly *Drosophila*, the model plant *Arabidopsis*, the nematode *C. elegans*, and budding yeast *S. cerevisiae*, respectively. (Adapted, with permission, from Kidwell and Lisch 2000 [copyright Elsevier Science].)

Here, in this diagram, we can see the proportion of these transposable elements (red), whereas the green ones are the other sequences. We have different organisms like *Homo sapiens, Z. mays, Drosophila melanogaster, Arabidopsis, C. elegans and S. cerevisiae (yeast).*

We can see that in human genome, these Transposable elements make up 35% (huge number), in *Z. mays* they are even more than that i.e. 50%, in *Drosophila melanogaster*, they are comparatively less i.e. 15% genome (y- axis represents the genome size), and in rest of the organisms, they have less proportions of those repeats.

MUHAMMAD IMRAN

So, mainly in humans and *Z. mays*, they are present as a major proportion.

Conclusions:

We conclude that:

- Large proportion of eukaryotic genome is composed of repeats
- Different repeats act as markers to detect genetic variation (of organisms) and are also used to study evolution of those organisms

Topic . 18 Transposable Elements (TEs)

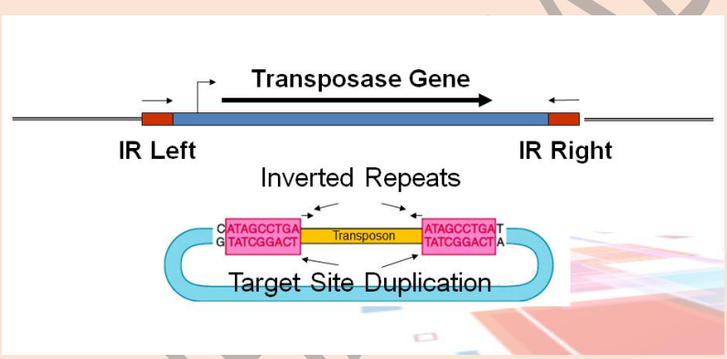
Introduction

Transposable Elements are the elements which can transpose i.e. they can move from one place to another in the genome and then they can cause the repetitive elements (repetitive DNA within that genome).

Insertion Sequences (IS elements)

These are the simplest transposable elements and their features are as follows:

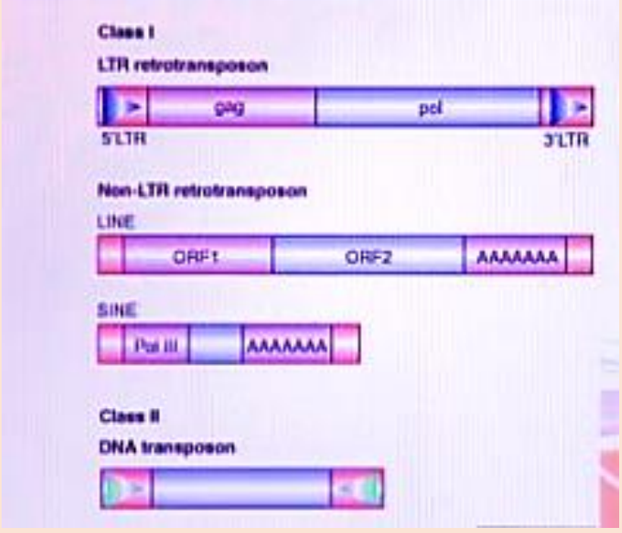
- They code only for the ability to transpose and are found in prokaryotes.
- They are usually very small (< 1 Kb to 2 Kb)
- They are flanked by inverted repeat sequences (IRs).
- They encode at least one gene that provides their own transposition functions.
- They do not code for noticeable (phenotypic) traits.
- They can cause mutations by transposition into genes.



Here, we see a structure of *Insertion Sequences* (IS elements). We have *Invert Repeats* (IR) which are anti-parallel or facing each other (repetitive elements; these are the distinctions in them). We have a transposase gene that gives the transposition properties to transposable elements. Another distinction in them is that when they get into the host genomes, they cause *target site duplication*- normally they create the sticky ends in the genomes of the host and later on when the complementary nucleotides are formed, they become those duplications (in pink).

Transposons:

Transposons are more complex transposable elements as compared to the simple IS (*Insertion Sequence*) elements and they code for additional characters in addition to the gene responsible for their transposition.



Here, in this diagram, we can see that there are two major classes termed as Class I and Class II.

Within Class I, we have **LTR** (**L**ong **T**erminal **R**epeats) retrotransposon (they have the ability of reverse transcription, so they have RNA's which gets converted into cDNA's; this is how they replicate). They have LTR at both ends (they are not invert but they are kind of forward repeats and are in the same direction).

In the same Class I, we have another category known as **Non-LTR** (**L**ong **T**erminal **R**epeats) retrotransposon, which doesn't have those LTR and are of two types namely **LINE** (**L**ong **I**nterspersed **N**uclear **E**lements) and are larger in size and **SINE** (**S**mall **I**nterspersed **N**uclear **E**lements) and are smaller in size. In Class II, we have DNA transposon.

Eukaryotic TEs:

As we have seen earlier, eukaryotic Transposable Elements has two classes in them; Class I and Class II.

In Class I, they have reverse transcriptase in them so they use RNA-mediated mechanisms of transcription. This Class I can be categorized into***LTR (Long Terminal Repeats) retrotransposons***, simple ***Retrotransposons***and ***Retrovirus like Elements.***

As we have seen earlier, that **Non-LTR** (**L**ong **T**erminal **R**epeats) retrotransposon comprises of **SINES** and **LINES.SINES** are Short Interspersed Nuclear Elements and are **80-300 bp** long and the example is ***Alu*repeats** (found in humans). Whereas **LINES** (Long Interspersed Nuclear Elements) are **6-8 kb** long and are relatively longer than SINES.

The example of **Class II TEs** is *Ac-Ds* (which were studied in maize - the work of Barbara Mcclintock led to the discovery of these transposons where he gave different characteristics to the kernels of the maize). Another example of **Class II TEs** is *P elements in Drosophila*.

Additional elements are known as **MITES** (**M**iniature **I**nverted **R**epeats TEs) that has the features of both Class I and II and are 400 bp long.

MUHAMMAD IMRAN

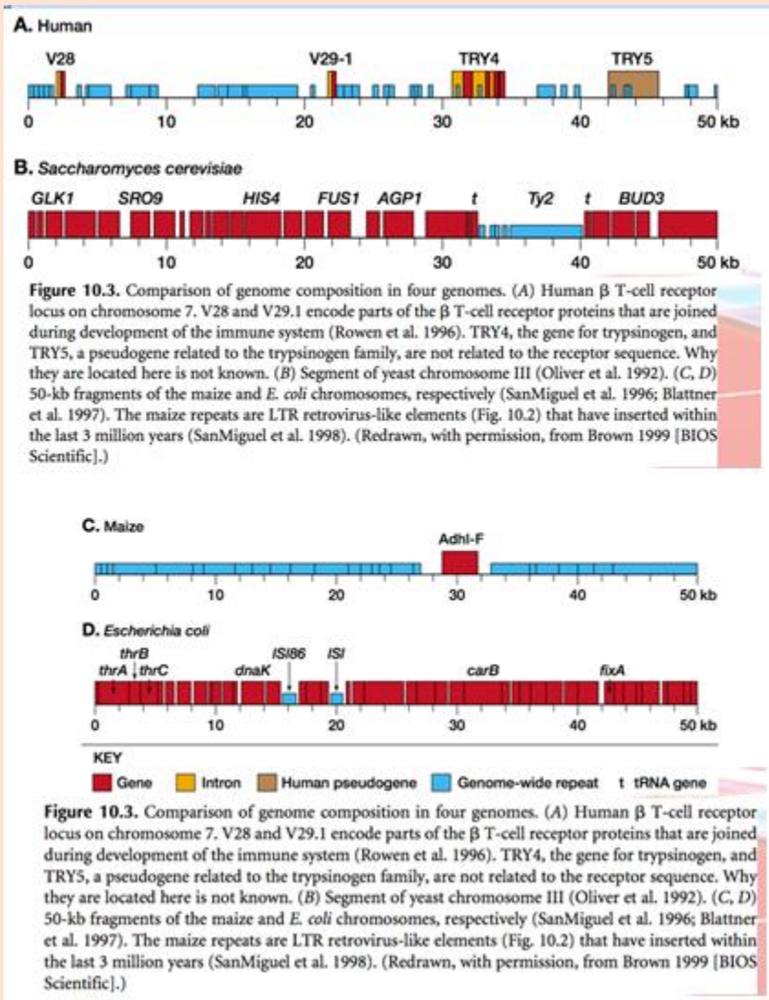


Figure 10.3. Comparison of genome composition in four genomes. (*A*) Human β T-cell receptor locus on chromosome 7. V28 and V29.1 encode parts of the β T-cell receptor proteins that are joined during development of the immune system (Rowen et al. 1996). TRY4, the gene for trypsinogen, and TRY5, a pseudogene related to the trypsinogen family, are not related to the receptor sequence. Why they are located here is not known. (*B*) Segment of yeast chromosome III (Oliver et al. 1992). (*C, D*) 50-kb fragments of the maize and *E. coli* chromosomes, respectively (SanMiguel et al. 1996; Blattner et al. 1997). The maize repeats are LTR retrovirus-like elements (Fig. 10.2) that have inserted within the last 3 million years (SanMiguel et al. 1998). (Redrawn, with permission, from Brown 1999 [BIOS Scientific].)

Figure 10.3. Comparison of genome composition in four genomes. (*A*) Human β T-cell receptor locus on chromosome 7. V28 and V29.1 encode parts of the β T-cell receptor proteins that are joined during development of the immune system (Rowen et al. 1996). TRY4, the gene for trypsinogen, and TRY5, a pseudogene related to the trypsinogen family, are not related to the receptor sequence. Why they are located here is not known. (*B*) Segment of yeast chromosome III (Oliver et al. 1992). (*C, D*) 50-kb fragments of the maize and *E. coli* chromosomes, respectively (SanMiguel et al. 1996; Blattner et al. 1997). The maize repeats are LTR retrovirus-like elements (Fig. 10.2) that have inserted within the last 3 million years (SanMiguel et al. 1998). (Redrawn, with permission, from Brown 1999 [BIOS Scientific].)

Here, in this figure, we see the comparison of human, yeast, maize and E-coli.

In blue color, we see those repetitive regions (transposons), red are the exons, whereas orange are introns and brown colored are the pseudo-genes.

We are comparing the 50kb region of the genome where in human, we can see the blue color prevails i.e. many transposable elements can be seen. In yeast the transposons are less in number. Whereas in maize, they are found to be in huge proportion and in E-coli we can see the IS (Insertion Sequences) elements which are extremely low in number.

Conclusions:

We conclude the following about Transposable elements:

- They make up a significant part of organisms' genome especially in that of the eukaryotic genome.
- They move within and across genomes and
- Causes genome expansion.

Topic # 19 Eukaryotic Gene Structure

Eukaryotic genes:

Eukaryotic genes are relatively complicated and are not simple as prokaryotic genes. They possess exons and introns.

Exons:

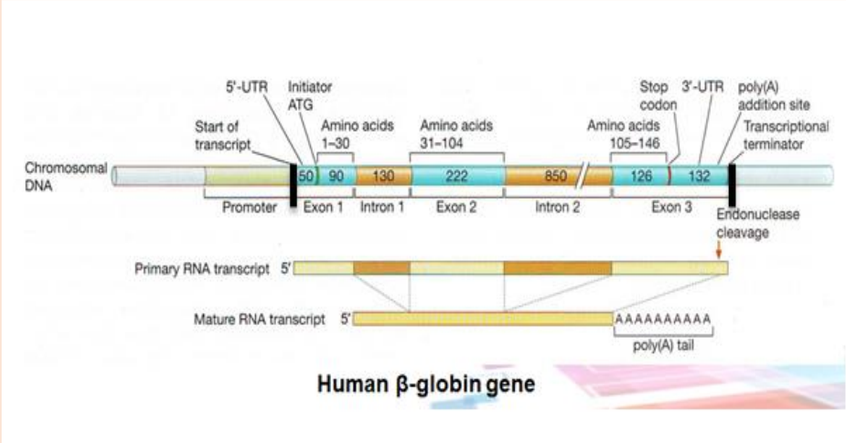
Exons are protein coding regions and are interrupted with introns (i.e. in between exons are introns). During the gene expression, both exons and introns are first transcribed into mRNA and then the introns are removed out, so the remaining structure is known as ORF (Open Reading Frames) which only consists of the exons.

Introns:

They contribute a very small proportion in yeast i.e. only 239 introns in its genome whereas in human the introns makes up 95% of its genome.

The introns stay on the same location and they might also have embedded genes in them.

They can be distinguished by the presence of GT at the 5' ends and AG towards the 3' end (**GT----------AG)** and this trend is highly preserved all over the genome.



Here, in this diagram is the structure of a typical eukaryotic gene.

We see the chromosome and gene is the region which has specific patterns so we can observe the promoter region (in the beginning of a gene), the blue ones are the exons and those orange ones are introns.

There is start of transcription (marked by black line) which ends at the exon3 (as shown here and marked by black line), this whole region is then transcribed into mRNA. We can see a 5'-UTR (Un-translated region) region, so this region is transcribed into mRNA but is not translated i.e. no protein is formed from this region, similarly we also have a 3'-UTR region.

When we see the ORF i.e. the region from start codon (initiator) to stop codon, and in between them we can see there are number of amino-acids, so this is the region from where translation takes place and we get a protein.

After transcription, the transcript is known as *Primary RNA transcript* and we can see that it also contains those introns. Which are later on removed through a process called as splicing and then we get a *Mature RNA transcript*, so that transcript is then translated into the proteins. This mature RNA transcript is also recognized by the presence of a poly-A tail (long runs of A's)

Intron origin

So, about the origin of introns, there are two theories which are as follows:

- **Intron-early**- According to this, they used to assemble the genes from already existing exons (so they brought the exons together and then these structures became the genes).
- **Intron-late**- According to this, the exons were already present with one another, then introns got into them (i.e. they Broke up previously continuous genes by inserting into them).

Number of Genes:

Now we talk about the degree of compactness, so the compact genomes whose size is small and the relative proportion of gene is higher which contributes to the variation in gene density. In short, we can say that compact genomes have higher genome density.

MUHAMMAD IMRAN

Organism	Genome size (haploid MB)	Predicted genes
A. thaliana (plant)	130	~25,000
C. elagans (worm)	100	18,424
Drosphila melangaster	180	13,601
Escherichia coli	4.7	4,288
Homo sapiens (human)	3000	45,000 – 120,000
S. cerevisiae (yeast)	13.5	6,241

Mount: Table 10.3, 11.3 2nd edn

Here, in this table, we have different genomes (mostly eukaryotes).

We can see that the size of genome in Arabidopsis is 130MB and number of genes is approximately around 25,000.

E-coli (prokaryote; bacteria) can be seen here, with genome size of 4.7MB and there are over 4,288 genes.

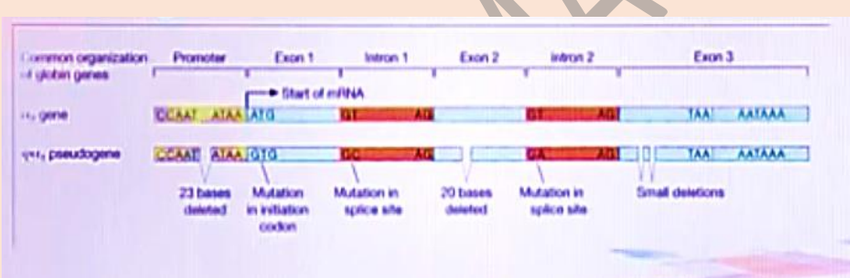
In humans, the genome size is 3000 MB (it's not megabytes, it's mega base pairs), and 45,000 to 120,000 genes (slightly around 30,000 have been identified).

So, if we look into those smaller organisms, like those having smaller genomes (E-coli), and if we take the number of genes and divide that to the genome size, we'll see that they have more densities as compared to the larger genomes.

Pseudogenes:

These are non-functional genes (sometimes there are mutations in the genes and if those mutations are present in some important regions then the genes' functions gets knocked out – known as psudogenes).

There is one category of them and is known as processed pseudogenes that lack introns and promoters.



Here, is the diagram in which we see the normal gene and a psuedogene.

In psudogene, we see that the bases are deleted from the promoter region (promoter must have specific pattern of bases in them), so in this case, this deletion is lethal. Similarly, in start codon (initiation code) we can see GTG instead of ATG, and we can also see the mutations in the splice sites (its GC rather than GT and GA rather than GT respectively). And in exon2, there are 20 bases which are also been deleted. Hence, this is how a normal gene has now become a pseudogene.

Gene families:

Sometimes, the set of genes have similar sequences as well as similar functions. And if we try to find out similar genes within an organism or between different organisms, performing similar functions can be categorized as gene families. Gene families arise from gene duplication and subsequent divergence events.

MUHAMMAD IMRAN

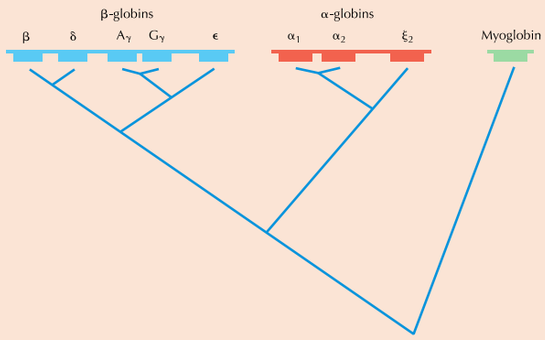


FIGURE 27.29. Gene duplications during the evolution of the human globin gene families. The initial split gave rise to two lineages, one leading to the modern gene for myoglobin and the other to the globin genes. Subsequently, the proto-α-globin and proto-β-globin lineages split following a duplication. Other duplications took place within the α and β lineages. (Modified from Strachan T. and Read A.P. *Human Molecular Genetics 2*, Fig. 14.16, © 1999 Garland Science.)

Evolution © 2008 Cold Spring Harbor Laboratory Press

Here, is the diagram showing the example of gene duplication, we have human globin gene.

So, first duplications give rise to two types of globins (one is myoglobin and other globins).

Within the second type (globins), we have alpha globin and and beta globin.

So this process is called as *gene duplication*.

Conclusions:

In the end, we conclude the following:

- Eukaryotic genes have exons and introns.
- Introns make up a significant portion of higher organisms' genome (Human genome).
- Pseudo genes are non-functional genes.
- The genes which are similar in function, they make up the gene families

Topic # 20 Comparative Genomics

Introduction:

Comparative genomics is where we compare the genomes of different organisms like we do comparison of gene number, gene content and gene location in both prokaryotic and eukaryotic groups of organisms.

The availability of genome makes it possible to have a comparison of all the proteins; proteome (so we can have the genome and translate them into the proteins or we can get the actual proteins and can do those protein comparisons – known as comparative proteomics).

Orthologs:

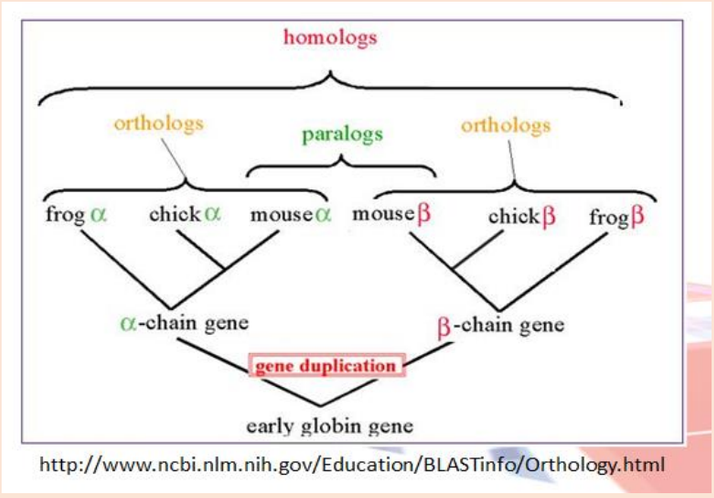
Orthologs are the genes that are present in two different organisms and are so similar that they must have the same function and the evolutionary history *(Fitch, 1970).*

Paralogs:

Paralogs are the gene families that originate from gene duplication events (they maybe within the same organism) over the evolutionary time.

MUHAMMAD IMRAN

(Unlike pseudogenes, 2nd copy of gene remains functional)



Here, in this diagram we can see that the similar genes are called as *homologs*, which is further divided into *orthologs*and *paralogs*categories.

The *Orthologs*, for example if we start from the bottom, we can see some globin gene (early) and then there is a gene duplication (as seen earlier), so it becomes a beta chain gene and an alpha chain gene.

Then those alpha chain genes can be seen in three different organisms (frog, chick and mouse) and are similar so that's why they are known as orthologs. Similarly, those beta chain genes can also be seen in three different organisms.

Now, if we look into those genes that are similar but gets diverted (and are in the same organism) as in mouse, chick and frog (we have alpha globins and beta globins) so these organisms maybe classified as paralogs.

***Drosophila* and yeast:**

When *Drosophila* is compared with yeast, we see that *Drosophila*has core proteome only twice the size of that of the yeast and *Drosophila* proteome is comparatively more similar to mammalian proteomes than worm or yeast.

Drosophila* and *C. elegans:

Now if we compare the fly *Drosophila* with the worm*C. elegans*, we can see that despite the large differences between them, the core proteome is of the similar size in both.

And nearly 30% of the fly genes have putative orthologs in the worm.

***Drosophila* and Human:**

When we compare fly *Drosophila* with Human, interestingly some human disease genes are absent in *Drosophila* but we can see a number of previously unknown counterparts to human cancer and neurological disorders genes are present in *Drosophila*, so it can be used as a good model in cancer studies.

Conclusions:

We conclude the following:

- Comparative genomics reveals the relationship among different organisms.
- Fruit fly has more similarities with mammals, so we can utilize it especially as a model for cancer studies

Topic #21 Comparative Proteomics Introduction

Introduction:

Since, genes encode proteins and these proteins perform actual life functions, we can have the genomes and get the translated protein from it and by comparing those translated proteins with one another, we can say it's a Comparative Proteomics study.

The collection of the protein sequences that are encoded by the genome makes up the proteome of that individual.

MUHAMMAD IMRAN

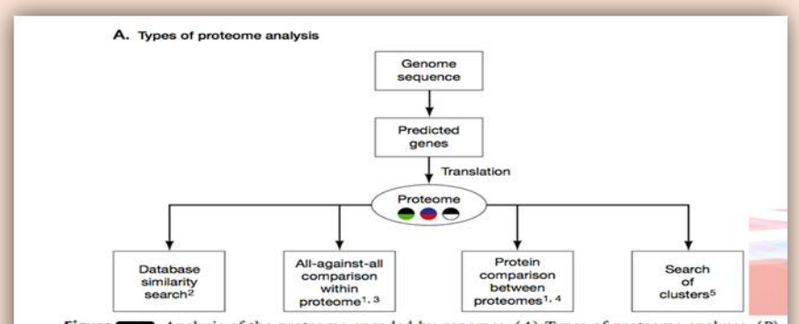


Figure [redacted]. Analysis of the proteome encoded by genomes. (*A*) Types of proteome analyses. (*B*) Examples of database hits resulting from domain structure of proteins. (*C*) Cluster analysis of similar sequences. (*D*) Domain identification.

Here, is the figure of how comparative proteomics can be done.

So from the genome sequence, we can get the predicted genes out of it, then we do translations which gives us the proteome of that individual. We can use those proteins and get some Database searches (we can find the homologous proteins, whereby we can predict their functions and roles).

We can also do the comparison of proteome by themselves so that will help us finding the paralogs (studied earlier) within that individual.

We can also do the comparison of proteins between different organisms and we can also find the clusters of co-related proteins in terms of their sequences and functions.

All against all, self-comparison:

Firstly we talk about the all against all; self-comparison, where we can do the following:

- Comparison of all proteins with each other within the individuals' proteome.
- Identify unique proteins from the ones having paralogs.
- Identify Gene families
- If we have a good match between query sequence and some other sequence, we can suspect those two are the paralogs (because they are present within the same organism).

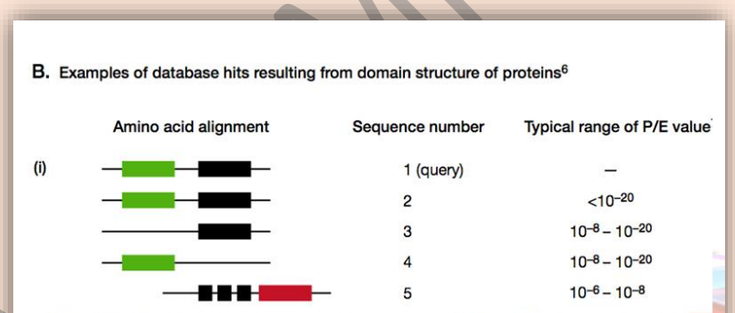


Figure [redacted] Analysis of the proteome encoded by genomes. (*A*) Types of proteome analyses. (*B*) Examples of database hits resulting from domain structure of proteins. (*C*) Cluster analysis of similar sequences. (*D*) Domain identification.

Here, in this figure we have the example of database searches.

We have some proteins which are aligned together.

Here, we can see some proteins which are similar to one another and as they are sharing the similar domains (domains are specific protein structure which is made up of specific amino-acids into particular structure).We do the sequence similarity searches by the software named as ***BLAST*** (will be discussed later).
We have some parameters (P/E), normally low P/E ratio is taken as a good score, for example if we have a good match we can have the range of P/E value less than 10^{-20} and it keeps on decreasing if we go down to other similar sequences (as shown).

Cluster Analysis:

In cluster analysis, we make the groups of proteins which are quite similar to one another. And reasons of doing it are as follows:

- To sort out the relationships of all the related proteins.

- Clustering classify the proteins based on some objective criteria e.g
  - ***E value cut off***
  - ***Distance in alignment*** (so the proteins which are more similar will be grouped together and distant proteins will be grouped from them so in this way we can have sub-groups or clusters in our data).

There are different clustering methods and are explained briefly in the below section.

Clustering by subgraph:

The way of clustering or grouping by the method of sub-graph is as follows:

- Each sequence is a vertex (vertex or vertices are the point or dots by which the edges (links) in a graph are connected. There can also be a vertex that's without any edge connecting to it, known as isolated vertex).
- Significant alignment score is an edge (on the vertices, we put our sequences and on the edges we put our alignment scores).
- Trimming by removing weak edges (if we have High P/E ratio, we will remove them).

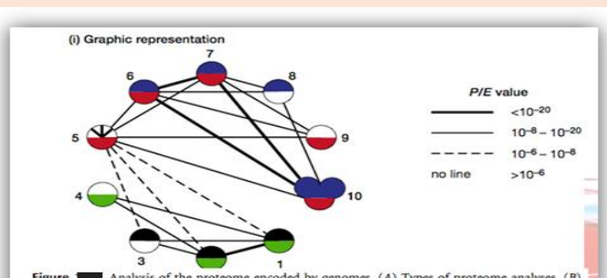


Figure Analysis of the proteome encoded by genomes. (*A*) Types of proteome analyses. (*B*) Examples of database hits resulting from domain structure of proteins. (*C*) Cluster analysis of similar sequences. (*D*) Domain identification.

Here, in this diagram we are representing the clustering, we have those vertices (shown as balls) which are the proteomes here, we have the edges (lines) which are the connection between those vertices and are the alignment scores (the thickness of edges relates with the small P/E value).

And if they are weakly co-related like the values are greater than 10^{-6} , so then they are not connected.

The dashed lines (or dotted lines) are where we have loose connections.

Clustering by Linkage :

There is another technique, where the clustering is done by linkage (almost similar to the previous ones except some changes). The method of doing it is as follows:

- Each sequence is a vertex.
- Significant alignment score is an edge.
- Trimming by removing weak edges (High P/E).
- Or remove > e^{-6} (we remove the ones which are not linked).
- Remaining sub-graph should share 2/3rd of edges.

Single Linkage:

Linkage is done by the following method:

- A group of sequences in all-against-all comparison is subjected to MSA (group those proteins which are co-related with multiple sequence alignment by first aligning them and then calculating their distances).
- Create distance matrix (by using those distance calculation just made).
- Neigbour joining is then used to do clustering (by distance matrices, we create those trees and the method used is *Neighbor joining*- will be discussed later).

MUHAMMAD IMRAN

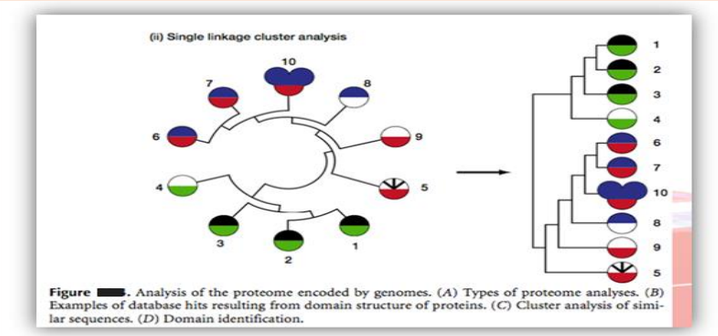


Figure ■. Analysis of the proteome encoded by genomes. (*A*) Types of proteome analyses. (*B*) Examples of database hits resulting from domain structure of proteins. (*C*) Cluster analysis of similar sequences. (*D*) Domain identification.

Here, in this figure we have the single linkage cluster analysis results, so we have those proteomes which are present in the end of those trees which we call them as leaves. The closely related ones are connected with one another.

We can see two types of arrangements, one is circular arrangement (on left) and we have typical binary tree arrangement (on right).

In binary tree, we can see that we have come up with two big clusters, and within those clusters we find those further groups or sub-clusters.

Core Proteome:

Core Proteome is when we do *all-against-all comparison*; which tells us about the proteins which are duplicated and it also gives us the information about those proteins which are uniquely present in those organisms- the core proteome.

Table 10.4. Numbers of gene families and duplicated genes in model organisms (Rubin et al. 2000)

Organism	Total number of genes	Number of gene families[a]	Number of duplicated genes[b]
Hemophilus influenzae (bacteria)	1709	1425[c]	284
Saccharomyces cerevisiae (yeast)	6241	4383	1858
Caenorhabditis elegans (worm)	18,424	9453	8971
Drosophila melanogaster (fly)	13, 600	8065	5536

[a] The number of clustered groups in the all-against-all analysis using the algorithm described in the text. This number represents the core proteome of the organism.
[b] Count of number of duplicated genes within the protein family clusters.
[c] 178 families have paralogs.

Here, in this table, we compare the core proteome with the total number of genes.

Here, for example in bacteria we have 17 hundred genes, 14 hundred are the gene families (represents the core proteome) and we have number of duplicated genes. In worm, we can see that we have 18 thousand genes where almost half of them are in the shape of gene families and rests of them are duplicated genes (so almost half of them are duplicated amongst them). In *Drosophila*, we have 13 thousand genes, 8 thousand gene families and 5536 are the duplicated genes.

Conclusions:

We conclude the following:

- Genome is translated into proteome.
- Self-comparison of proteome yields gene families and duplications

Topic # 22 Between-Proteome comparisons

Introduction:

In between-proteome comparisons, we compare the proteomes from different organisms with one another, so in this way we can find the genes which are similar between those organisms, and we call them as orthologous genes.

Here we take the proteome as a query and we do a database similarity search against another proteome or there can be a whole database where we have the set of proteomes.

If the proteome is not available, we can search the EST (Expressed Sequence Tag) Database as well.

Significance:

MUHAMMAD IMRAN

As mentioned earlier, this helps in finding the orthologs, gene families and the domains (between different organisms). There can be other significance of *between-proteome comparisons search* and are as follows:

- Proteins that have a highly significant alignment score can be suspected as the orthologs.
- Mostly the proteins that are related to core biological functions (basic functions of life) are likely to be orthologs.

Finding true orthologs:

How can we find true orthologs?
There is a technique which we call it as:

Method 1:

Where we have the **"Reciprocal Hits",** in this method, you take one organism at a time as a query and search against the other as a database and then you flip around and take first as database and second one as a query. So, if you are getting similar end results, it means that the same genes are co-related with one another (or highly similar to one another), and we can keep them as the best hits.

We can also apply some criteria on hit, for example here we can do a E-value cut off when we do BLAST (BLAST gives us the parameter called as E-value and a lower value is considered good and we'll talk about BLAST algorithm in coming up lectures but here the point is to let you know that we do some E-value cut off) and we can retain those gene pairs in those Reciprocal hits.

So, for example here we say that E < 0.01, we can retain them.

Similarly, while we are doing BLAST, since it is a local search tool, we can compare different regions amongst different genes or proteins, we look for the matches between different regions and sometimes both proteins are not greatly covered in the alignment, so we want to have at least some coverage criteria, so here we have like for example 60% coverage.

So we keep those matched pairs normally with a very conservative or low P value like 10^{-10} to 10^{-100} .

Clusters of Orthologous Group (COG):

In this way, while we are finding the true orthologs, we can group those organisms which are similar to one another and this method is known as Clusters of Orthologous Group (COG).

Orthologs are assumed to be derived from common ancestor, so they might also have paralogs (within the organisms, the orthologs might also go through the duplications).

Orthologs are clustered to form **COG** (can be studied as COGs).

Table 10.5. *Numbers of closely related yeast and worm sequences*

Cut-off *P* value	$< 10^{-10}$	$< 10^{-20}$	$< 10^{-50}$	$< 10^{-100}$
Total number of sequence groups	1171	984	552	236
Number of groups with more than two members	560	442	230	79
Number and percent of all yeast proteins (6217) represented in groups	2697 (40)	1848 (30)	888 (14)	330 (5)
Number and percent of all worm proteins represented in groups	3653 (19)	2497 (13)	1094 (6)	370 (2)

Adapted, with permission, from Chervitz et al. 1998 (copyright AAAS).

Here, in this table we have the example of the closely related organisms' sequences i.e. Orthologs (between yeast and worms).

We have different P cut-offs (like $<10^{-10}$, $<10^{-20}$, $<10^{-50}$, $<10^{-100}$).

We have the '*total number of sequence groups*' and at different cut offs if we go on a stringent criteria, we have less common orthologous groups and if keep the criteria less strict we can have more of these orthologous groups.

MUHAMMAD IMRAN

Then we have the '*number of groups with more than two members'* (as shown).
Lastly, we have the '*percentage of yeast*' and '*percentage of worm*' (i.e. how many amongst the total, they are present) and are presented in these two groups (the yeast and the worm), say we have 40 percent and 19 percent on $<10^{-10}$ cut-off P-value, and we have 5 percent and 2 percent on $<10^{-100}$ cut-off P-value (if our criteria is strict).

So, in this way we can group the similar proteins at different cut offs of P-values, and we can have the various results.

Proteomes to EST databases:

Sometimes, we take those proteomes and we match them or align them with Expressed Sequence Tags (EST) (which is cDNA copies of cell's mRNA sequences). We do this procedure for those organisms' genomes whose sequences are not available.

ESTs are single DNA reads and are mostly 3' biased (since we get them from mRNA and mRNA extraction protocols relies on getting those mRNA by using their 3-prime poly A-tail which is present on their 3-prime end, so that is why they are kind of more tiled or oriented towards 3-prime ends as they are mainly extracted from this site).

EST may be incomplete because it is wholly dependent upon the gene expression, so if we do not have genomes rather than we only have the ESTs, we might be biased towards only those genes which are expressed.

The softwareor the package in BLAST which is being frequently used for this purpose is **TBLASTN**.

Family and Domain Analysis:

Proteins are organized into domains that represent modules of structure or function (as domains are specific arrangements of amino-acids). And domain comparison sometimes is co-related with their biological functions.

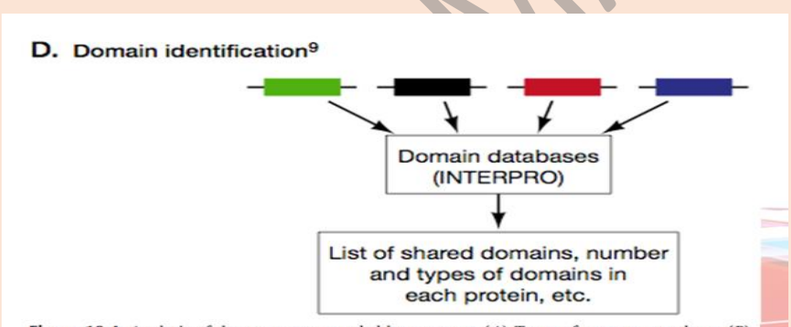


Figure 10.4. Analysis of the proteome encoded by genomes. (*A*) Types of proteome analyses. (*B*) Examples of database hits resulting from domain structure of proteins. (*C*) Cluster analysis of similar sequences. (*D*) Domain identification.

Here, in this figure for an example, we take the domains from different proteins and we put them into *Domain Databases*, and then in the end we can come up with the shared domains; the domains which are present in these different groups and lastly we can co-relate this information to have an idea about their functions.

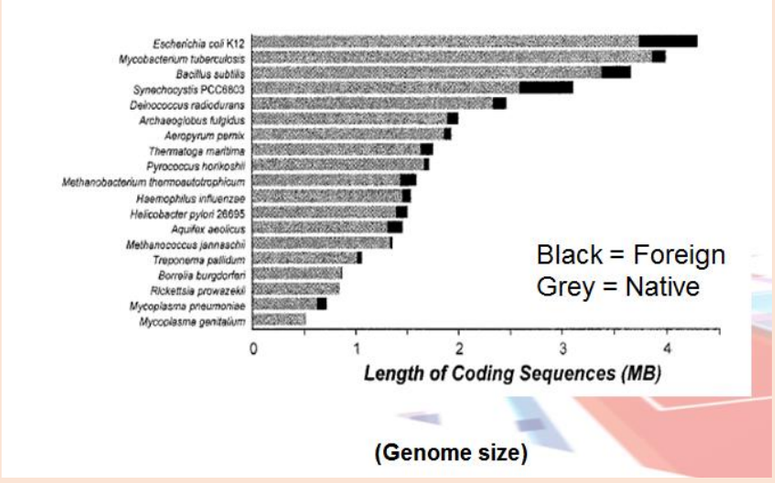
Conclusions:

- Proteome comparison helps finding orthologs, gene families and protein domains
- Domain comparison reveals their biological roles.

Topic #23 Horizontal Gene Transfer

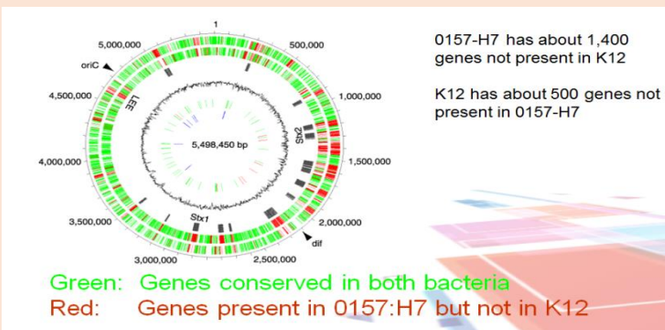
Introduction:

Horizontal Gene Transfer is where the genes are transferred at the same levels or horizontally (so two organisms they transfer their genes and it's not like transferring the genes from top to bottom like vertical transfers for example, from parent to the offspring) and is relatively slow process in evolution.



Here, in this chart we have the length of the coding sequences and the proportions which are coming from the foreign elements (in black). (Grey are the native ones).

So this is an indication of the horizontal gene transfer. We can see here that mostly in E-coli K12, there is the biggest portion of foreign gene and in Synechocystis, we can observe the huge part of foreign genes as well.



Here, is another case in which there is the comparative map of two genomes from E-coli 0157-h7 and K12.

We can see that here are about 1400 genes present in H7 which are not present in K12.

Similarly there are 500 genes found in K12 which are not seen to be in H7.

In this circular arrangement, we see the red which indicates the genes present in H7 but not in K12. Whereas green are the ones which are conserved in both the organisms.

Other Examples of Genes of Foreign Origin in Complex Eukaryotes

Eukaryote	Foreign Genes	Source
Various Plants	Hormone synthases	Bacterial
Aphids	Carotenoids	Fungal
Sturgeons	Various (15 genes)	Trematodes
Sea Slug	*psoB*, encodes a nuclear factor	Alga

Here, is the table of another example in which we observe the horizontal gene transfer in eukaryotes.

We have different plants, which has *hormone synthases gene* (a foreign gene) and is suspected to be coming from the bacteria.

The Aphids (insects) which has *carotenoids* as a foreign gene and it might be coming from fungal.

Sturgeons have some 15 genes which are foreign and are thought to be coming from Trematodes.
Sea Slug that has a *psoB*gene which encodes a nuclear factor and the source is some Alga.

Conclusion:

Horizontal Transfer of genes between different organisms is a relatively slow process that leads to acquisition of new traits.

Topic #24 Gene Order (Synteny)

Gene Mapping

- Gene mapping is determining the location of and relative distances between genes on a chromosome

Genetic vs physical Mapping

- Genetic map distances are based on the genetic linkage information measured in Centi-Morgans (CM)

Genetic vs physical Mapping

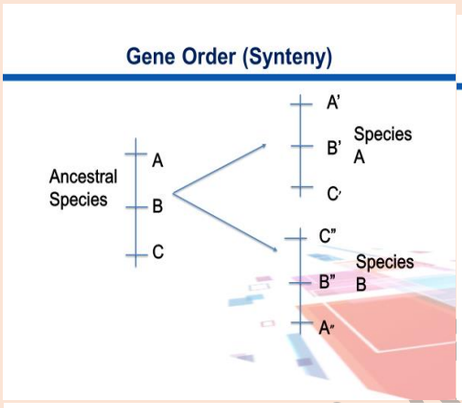
- Physical maps use actual physical distances usually measured in number of base pairs

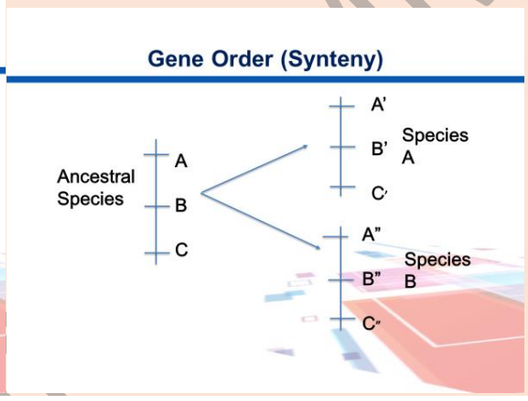
Synteny

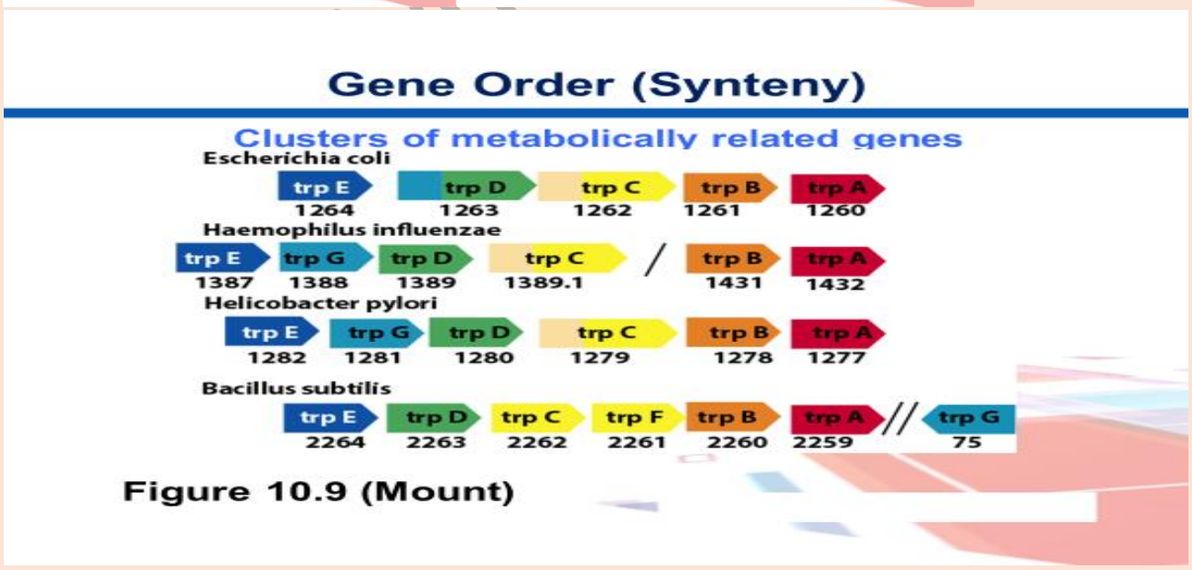
- Arrangement of genes on the chromosomes
- Comparing gene orders provides a good measure of similarities and differences among different organisms

Conserved Synteny

- Species diverged from common ancestor have similar chromosomes and gene order
- Gene duplication and rearrangements changes synteny
- As a results species diverge







Gene Order (Synteny)

Human Chromosomes

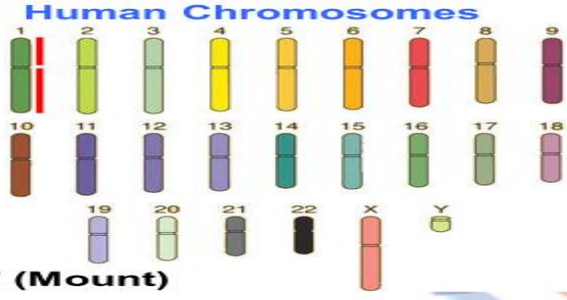


Figure 10.7 (Mount)

Gene Order (Synteny)

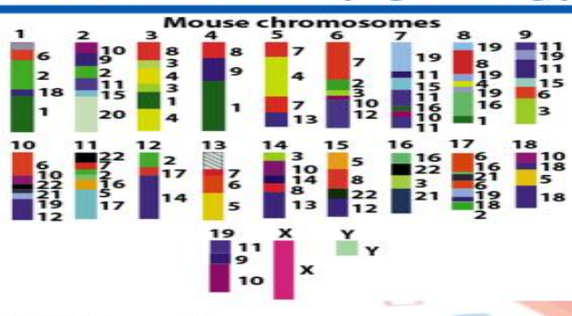


Figure 10.7 (Mount)

Gene Order (Synteny)

Analysis of Chromosomal Rearrangements

Sequence Alignment

Genome Alignment

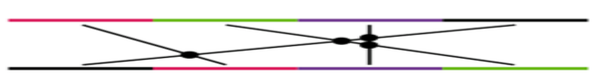


Figure 10.8: Mount

Gene Order (Synteny)

Alignment Reduction

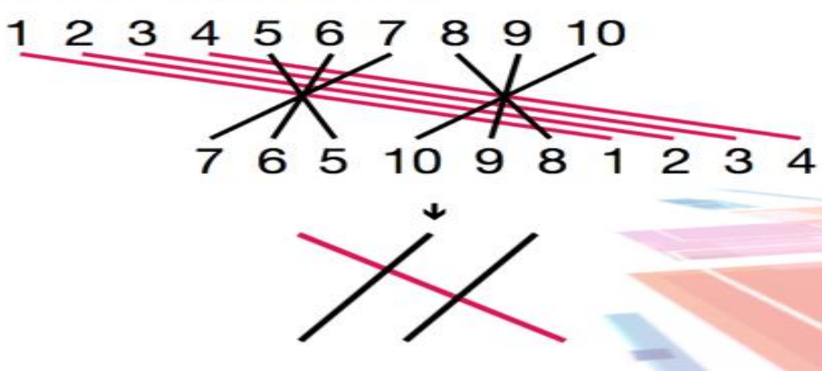
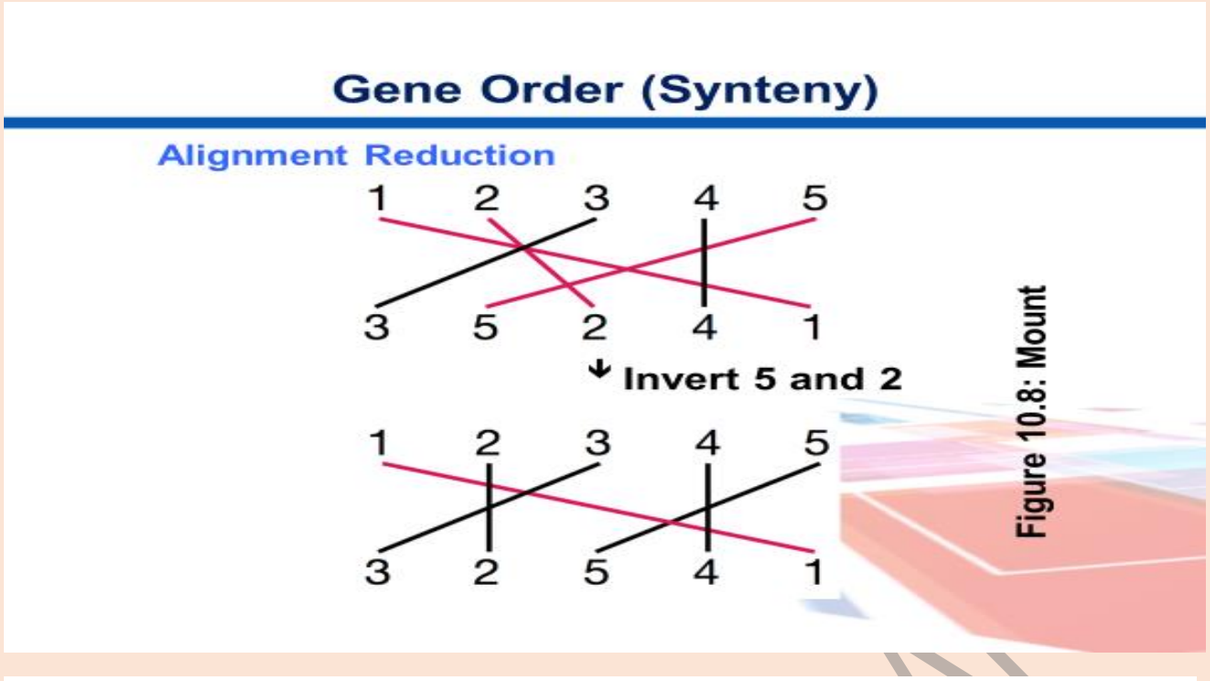
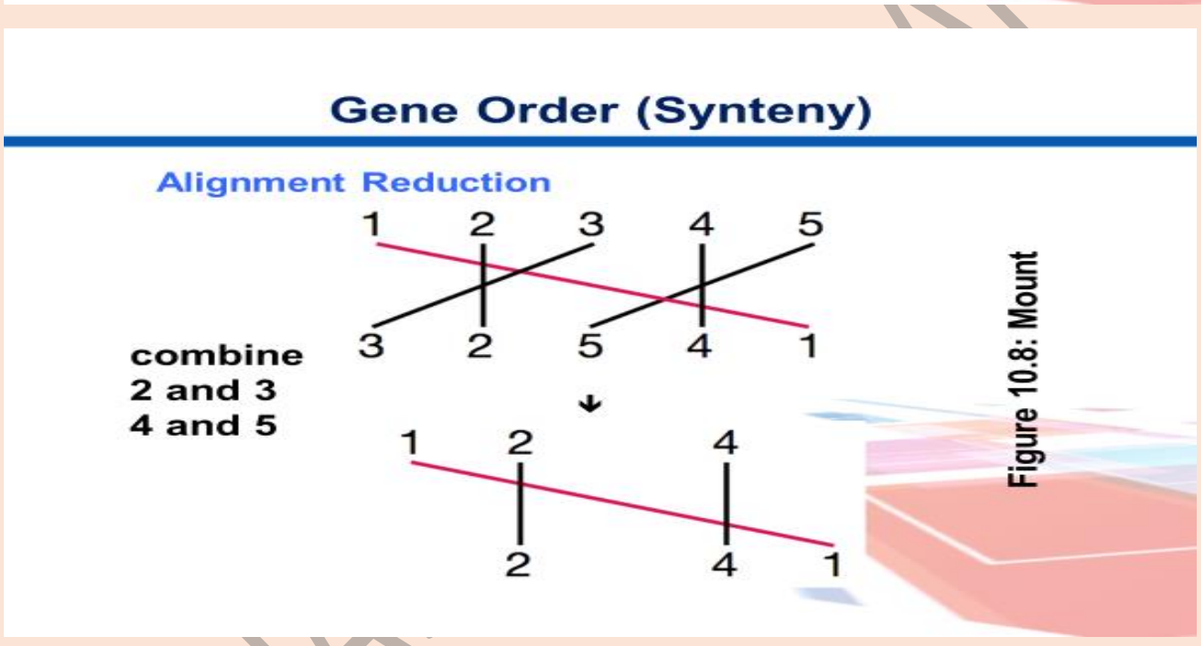


Figure 10.8: Mount





Conclusions

Species diverged from a common, expected to have similar synteny

Functionally related genes stay close as clusters

Topic # 25 Genome Annotations

Genome annotation:

After the genomes are assembled together, the genome annotation is a very important task whereby we could gain information of the genome. It has several procedures involved and which are as follows:

- We try to locate some important genes which are protein coding genes and also their products.
- We also try locating the RNA-encoding genes.
- We recognize the non-coding regions in a genome.
- We can also predict the function of the genes.
- The Biochemistry and structure of gene products can also be obtained.
- We can explore the Literature links.

- We can also explore the links to genetic maps where they are located on the chromosomes.
- We can look into the location of the repeats.
- We can also look into the location of STS (sequence tag sites).
- We can also look into the location of sequence polymorphisms.
- And we can find the significant alignment to some protein sequences of known function in databases (by comparison).

Annotations steps:

Annotations are divided into two types and are as follows:

- Structural annotation
- Functional annotation

Structural Annotation:

Structural Annotation is where we try to identify certain gene features like;

- Promoters
- Terminators
- Shine-Dalgarno sites; the ribosomal binding sites during the protein synthesis)
- DNA motifs (patterns of nucleotides within the genes)
- Co-transcription units
- Operons in microbes (in micro-organisms, lots of genes are transcribed together known as operons)

Annotations Tools:

There are two tools which are important worth mentioning here, one is MAGPIE and the other is GENEQUIZ and these are designed to assist with gene the genome annotations.

MAGPIE (Multipurpose Automated Genome Project Investigation Environment) - It's an automated genome analysis tool that is used for structural annotation.

GENEQUIZ- Focuses on deriving a predicted protein function based upon the available evidence; including evaluation of similarity to the closest homologue in the database (i.e. it is good tool for functional annotation).

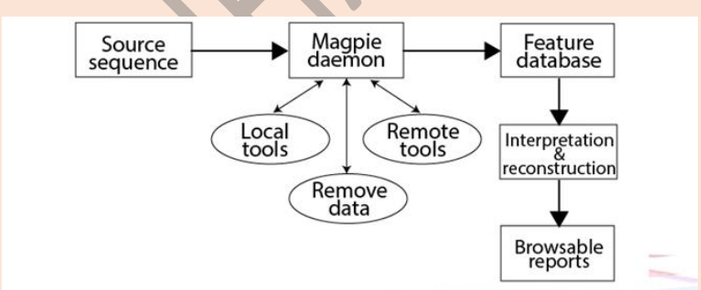


Figure 10.10 (Mount)

Here, in this figure, we have an outline and is a kind of a work flow of how exactly MAGPIE works.

We take some *source sequence,* and we give it to some software program known as *Magpie Daemon;* it takes the sequences (which are added to the database, so it automatically gets them) and sends that data to the *local tools*, *remote tools* (over the internet) and then it explores some specific features or annotation patterns and it puts them into the *Feature Database.*

Then later on those results are interpreted (*Interpretation and Reconstruction*) and then we get the *reports.*

In this way, we have a kind of automated annotation gathering tool

Functional Annotations:

The attributing biological information to the genes is called as functional Annotations and we can have it via;

- Biological function
- Biochemical function
- Gene expression (transcription of the gene is considered as gene expression)
- Regulation and interactions among different genes

8 Group Classifications:

There are different classification schemes, which are meant for functional classification, to classify the genes and their products into one of these groups;

- Enzymes
- Transporters
- Regulators
- Membranes
- Structural elements
- Protein factors
- Leader peptides (control transcription and translation)
- Carriers (transporters)

In this way, scientists have seen that 90% of the *E-coli* genes fit into these categories (so their annotations can be explained).

Enzyme Commission (EC) numbers:

It is another scheme which was put forward by the **E**nzyme **C**ommission (EC) that was working under the **IUBMB (International Union for Biochemistry and Molecular Biology).**

They say that the enzymes are classified on the basis of the reactions they catalyze and have a 4-digit scheme which is actually the enzyme commission number: **EC a.b.c.d**

'a' (first digit**)** informs that it is from one of the 6 classes of biochemical reactions (enzyme might be coming from one of these classes).
'b' (second digit**)** informs that is from the group of substrate (the thing on which the enzyme attacks)**.**
'c' (third digit**)** informs us that it is anaccepter molecule.
'd' (fourth digit**)** gives thedetails of biochemical reaction

For example,

tripeptideaminopeptidases

EC 3.4.11.4

Where **3** – tells us that it is a Hydrolase (use water to break substrate).

This **3.4-** tell us thatit is aHydrolase that acts on the peptide bonds.

The **3.4.11-** tells us that it is a Hydrolase that cleaves the amino terminal amino acids of polypeptide.

While putting everything together**, EC 3.4.11.4-** it tells us that it is a Hydrolase that cleaves the amino terminal amino acids of a tri-peptide.

With Enzyme Commission Scheme, they classified that 70% of E-coli genes shared a and b (first two classes), which means that they catalyzes the same biochemical reaction.

Three Groups Scheme:

This is another classification scheme known as a 'Three Groups Scheme', where we divide all those functions which are related to the following:

- Energy

MUHAMMAD IMRAN

- Information
- Communication

It was found that plants devotes half of their genome to the **energy metabolism** (they make food), whereas animals devotes half of their genome to **communication** (they talk a lot :D)

Conclusions:

We conclude the following:

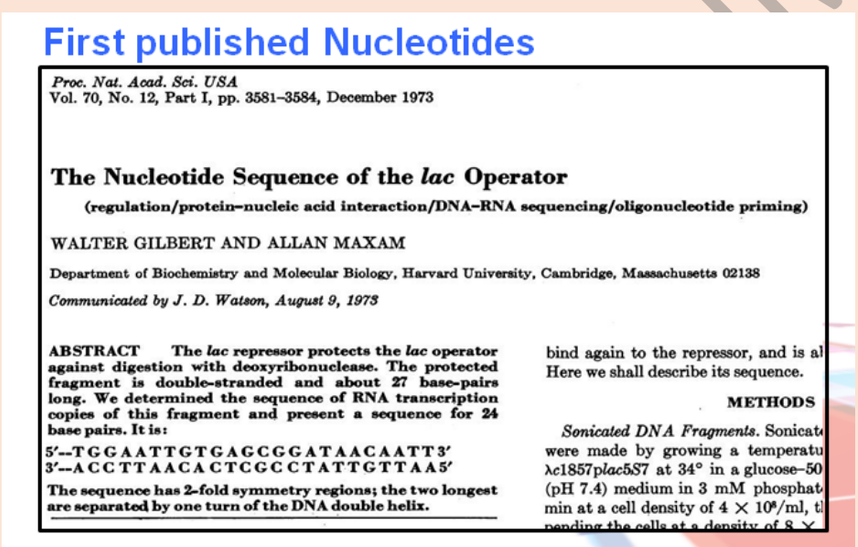
- Finding genes and their coding regions is an important task in Genome annotations.
- Functional annotations correlate the genes to different classes of functions

Topic # 26 Genome Sequencing

Introduction

Genome sequencing involves recognition of nucleotides in a Genome and determining their precise order of arrangement. And we have seen that the advances in sequencing technologies have revolutionized the pace of scientific discovery.

DNA sequencing began in 1970s with the development of Maxam-Gilbert's method and later got the pace with Sanger's method so most of the modern day sequencing employs the variants of Sanger's methods.



First published Nucleotides

Proc. Nat. Acad. Sci. USA
Vol. 70, No. 12, Part I, pp. 3581–3584, December 1973

The Nucleotide Sequence of the *lac* Operator

(regulation/protein–nucleic acid interaction/DNA–RNA sequencing/oligonucleotide priming)

WALTER GILBERT AND ALLAN MAXAM

Department of Biochemistry and Molecular Biology, Harvard University, Cambridge, Massachusetts 02138

Communicated by J. D. Watson, August 9, 1973

ABSTRACT The *lac* repressor protects the *lac* operator against digestion with deoxyribonuclease. The protected fragment is double-stranded and about 27 base-pairs long. We determined the sequence of RNA transcription copies of this fragment and present a sequence for 24 base pairs. It is:

5′--T G G A A T T G T G A G C G G A T A A C A A T T 3′
3′--A C C T T A A C A C T C G C C T A T T G T T A A 5′

The sequence has 2-fold symmetry regions; the two longest are separated by one turn of the DNA double helix.

bind again to the repressor, and is al
Here we shall describe its sequence.

METHODS

Sonicated DNA Fragments. Sonicat
were made by growing a temperatu
λc1857*plac*5S7 at 34° in a glucose–50
(pH 7.4) medium in 3 mM phosphat
min at a cell density of 4 × 10⁸/ml, t
pending the cells at a density of 8 ×

Here, we see the first published nucleotide, which was published by Gilbert and Maxam.

They presented the first 24 base pairs of the DNA (working on the *lac* Operator).

First genome sequenced:

So first genome was RNA of a virus (phage) **MS2** and was sequenced by **Walter Fiers** and colleagues at University of Ghent, Belgium (1972-76).We can say that RNA was first sequenced among nucleic acids (first complete genome sequence made was that of RNA).

Among the free living organisms, ***Haemophilusinfluenzae***was the first ever published genome by Fleischmann et al. in 1995. (Since we are in a discussion that viruses are living or non-living creatures, so if we talk about the living ones, it is the bacterium which was the first living organism whose published genome was made).

MS2:

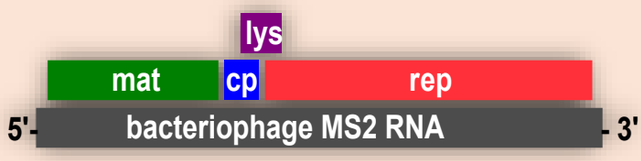
MS2 is a virus that infects *E-Coli* and *Enterobacteriaceae*. It is a single stranded sense RNA (in the host it gets duplicated and where anti-sense is formed and from that the RNA copies are again formed) and it encodes MS2 coat protein.

MS2 sequencing:

The entire nucleotide sequence was established by nuclease digestion and characterization of a fragment (it had the enzymatic digestion of the nucleotides and that's how it was sequenced).

MS2 genome

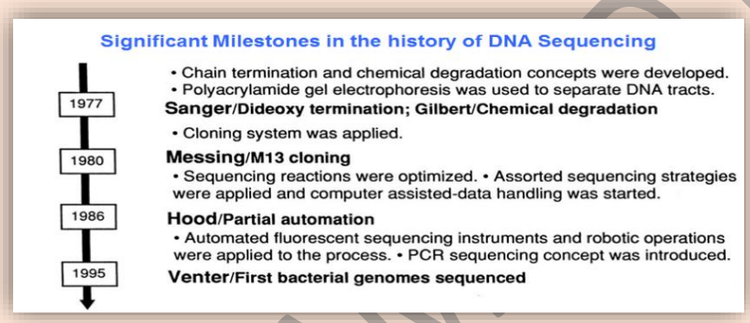
MS2 has 49 different codons in the genetic code that specify the sequence of the 129 amino-acids long coat polypeptide (virus has a coat which is made up of proteins on its outer side and it has a RNA; it's genome).



Here, we see the virus genome, which has the genes like *mat* (helps in assembly; putting those proteins together), *cp* (codes for coat protein), *rep* (codes for replicase protein) and *lys*(codes for lysis protein; that breaks the host cells). When we observe *cp*, *rep*, and*lys*, we can see the *lys*gene is embedded between these two genes, so we can have the genes within the genes (here).

Here, is the history of different milestones, which were done while we are moving towards this DNA sequencing.

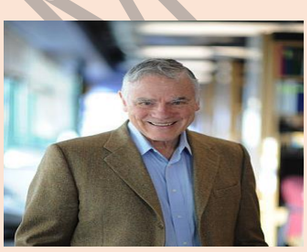
So we start way back in 1870 with the discovery of DNA, then in 1953, we see the major breakthrough is Watson and Crick Model



Here, in 1977, we can see that Sanger's method arose. And worth mentioning here is of Hood's name which gave us the automated sequencer that also changed the pace of technologies in this sequencing business.

Automated sequencing:

Leroy E. Hood's laboratory at the California Institute of Technology invented the first **semi-automated DNA sequencer**in 1986, which is the key technology used in **Human Genome Project.**



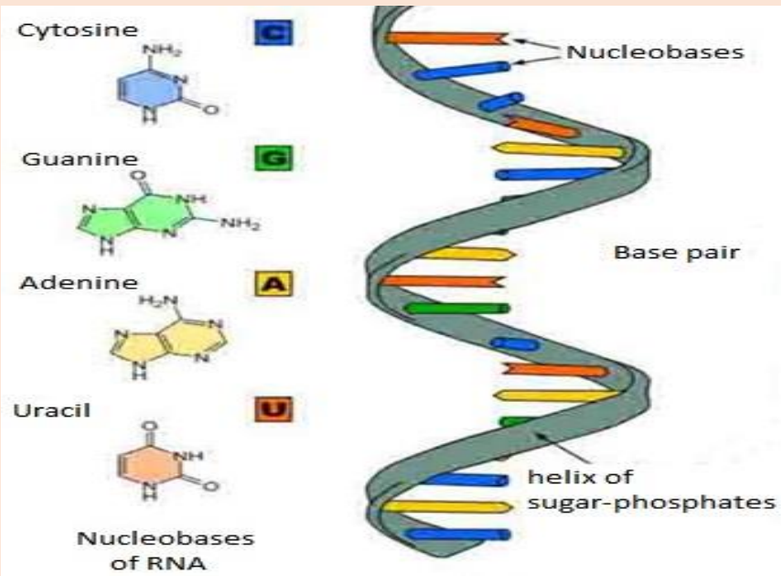
Leroy E. Hood
Institute of system biology
Seatle Washington
Conclusions:

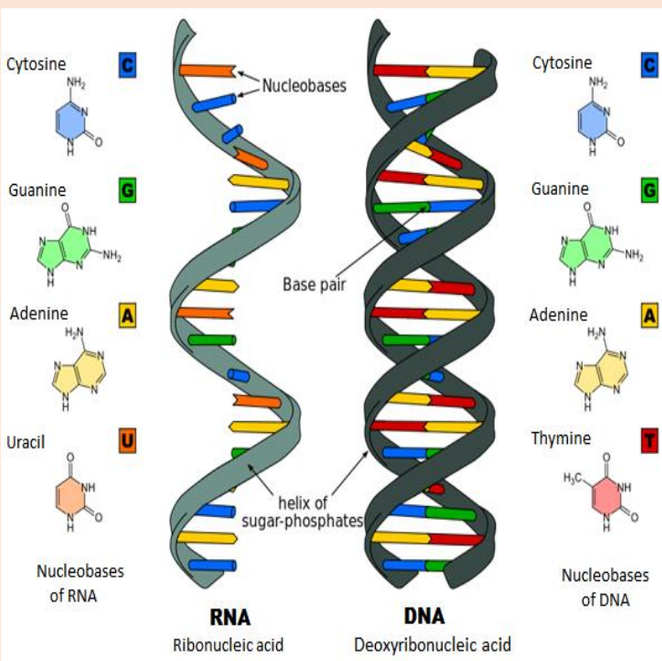
We conclude the following:

- Genome sequencing involves recognition and determining the precise order of nucleotides in a Genome.
- Advances in sequencing technologies have revolutionized the pace of scientific discover

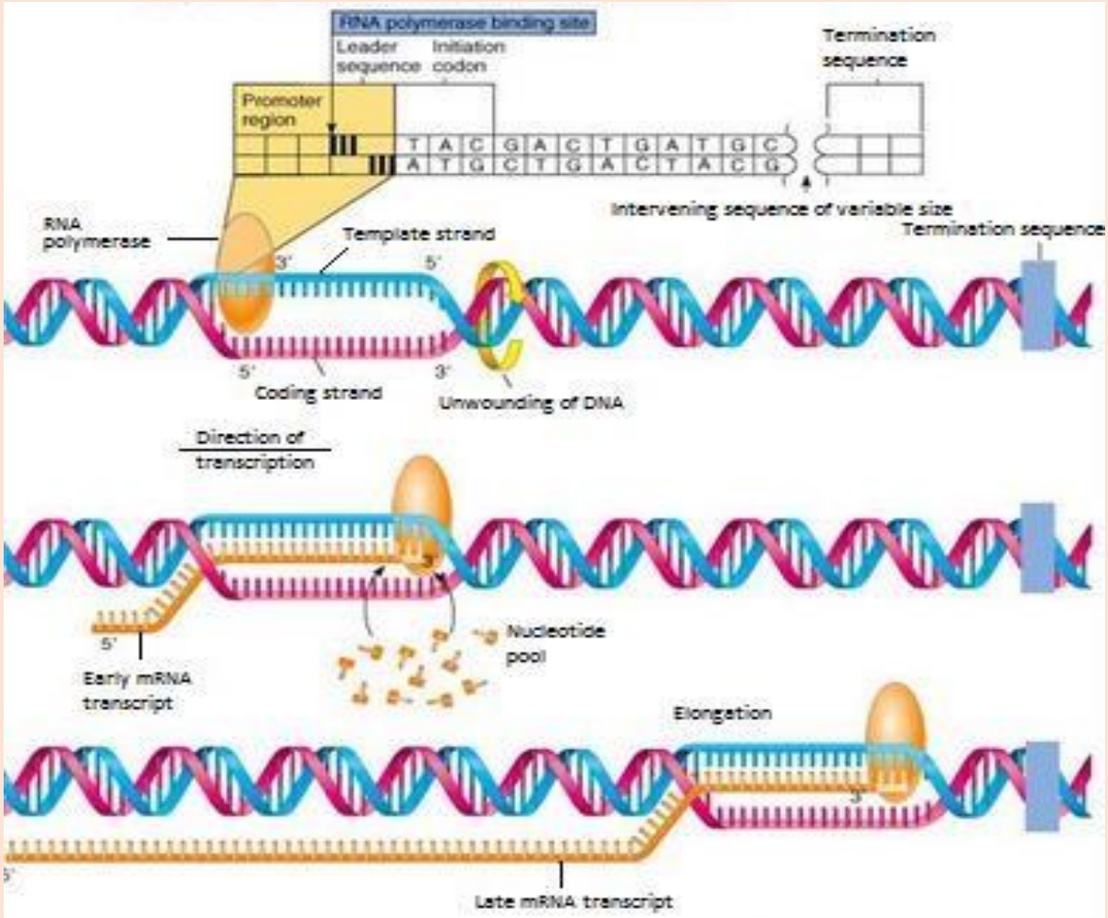
Topic # 27 Advanced Computing Approaches

Structure of RNA



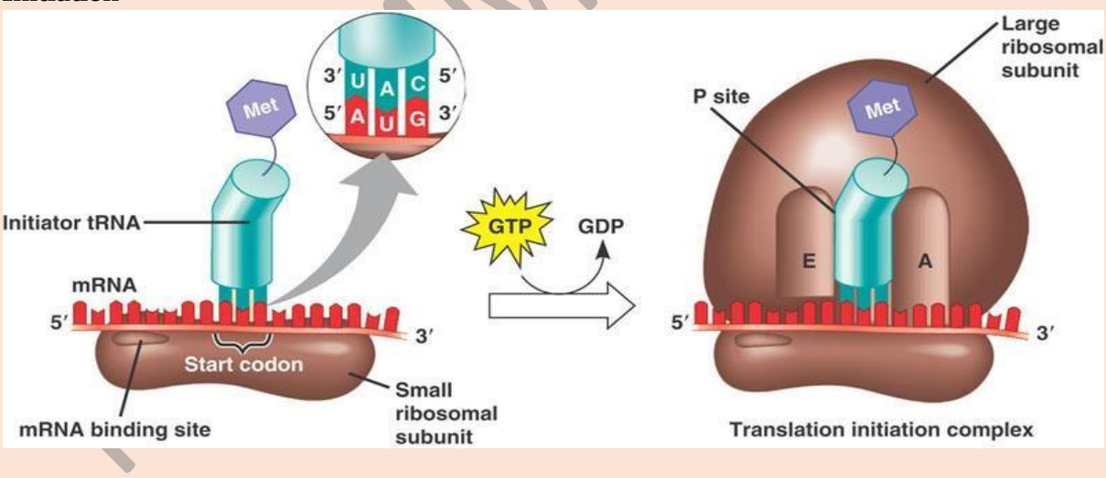


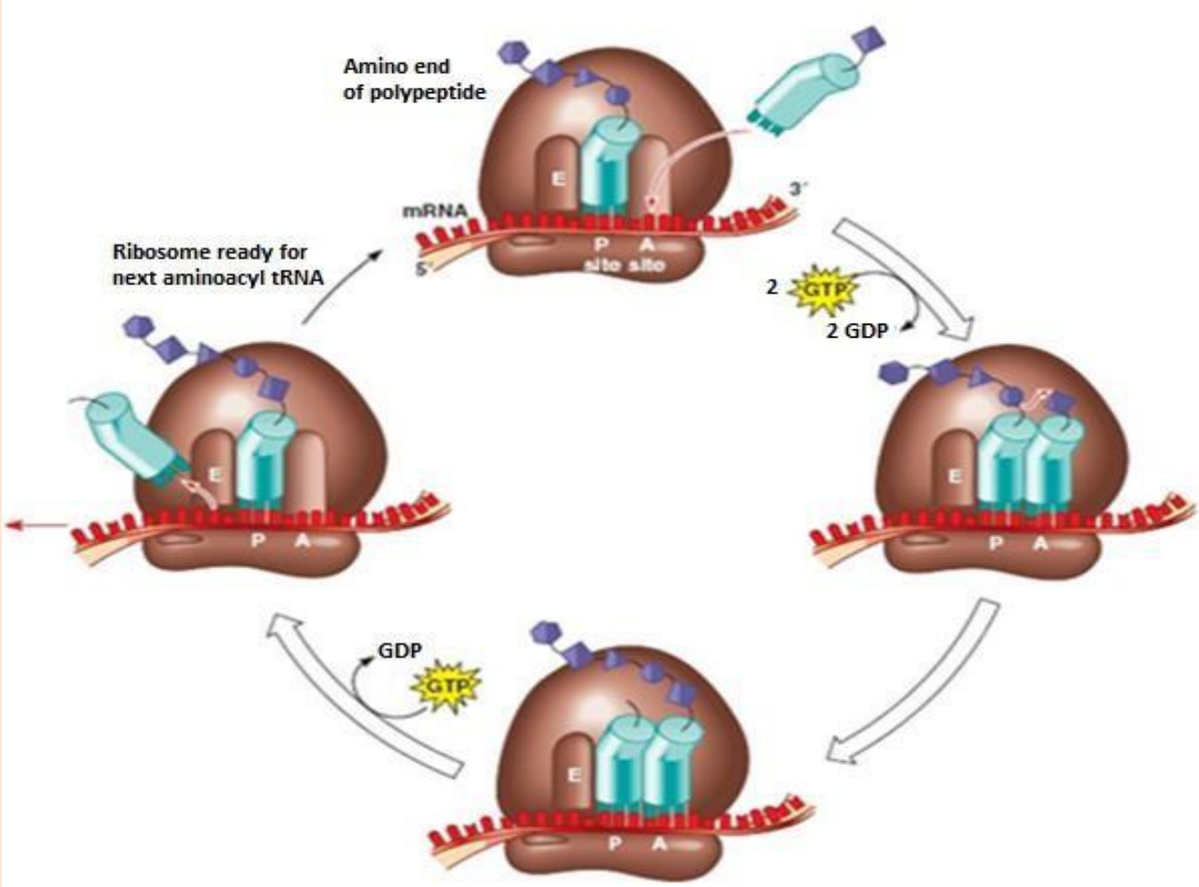
Topic # 28 DNA Transcription

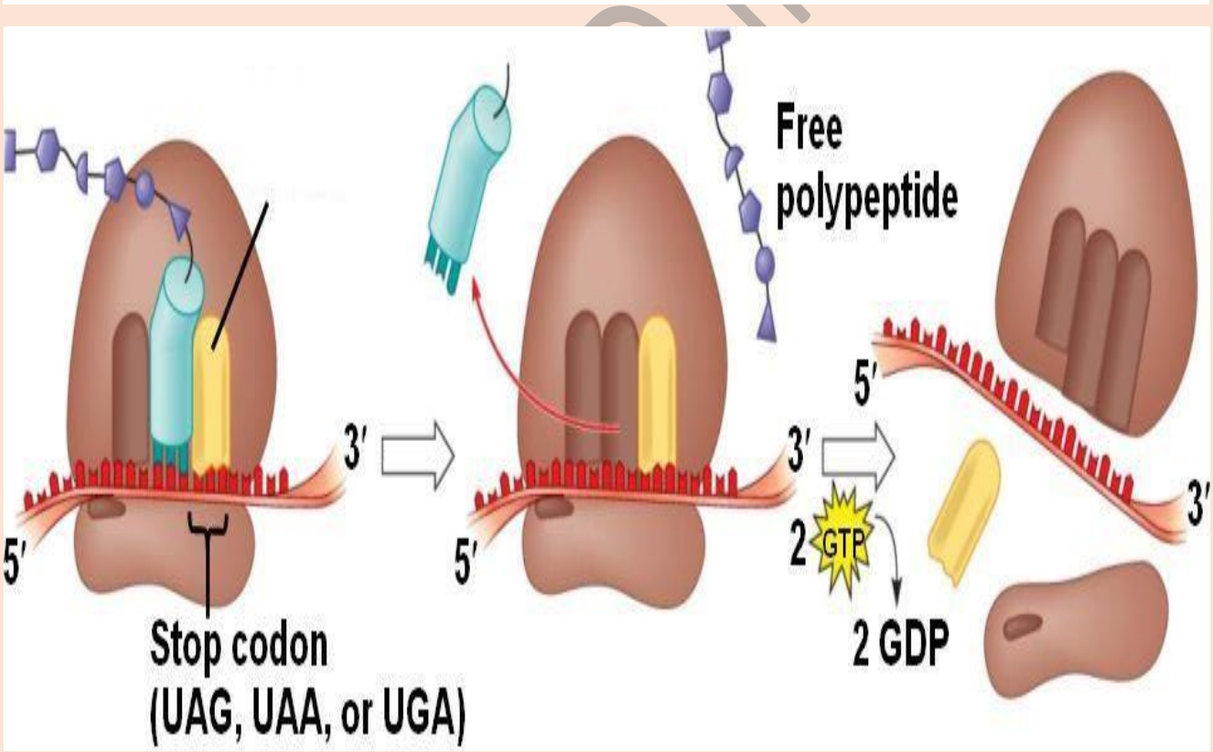


Topic # 29 Protein Translation

Initiation




Amino end
of polypeptide
mRNA
Ribosome ready for
next aminoacyl tRNA
E
P
A
site site
3'
5'
2
GTP
2 GDP
GDP
GTP


Free
polypeptide
3'
5'
Stop codon
(UAG, UAA, or UGA)
2
GTP
2 GDP

Second base

First base (5' terminus)	U	C	A	G	Third base (3' terminus)
U	UUU, UUC } Phe UUA, UUG } Leu	UCU, UCC, UCA, UCG } Ser	UAU, UAC } Tyr UAA Stop UAG Stop	UGU, UGC } Cys UGA Stop UGG Trp	U C A G
C	CUU, CUC, CUA, CUG } Leu	CCU, CCC, CCA, CCG } Pro	CAU, CAC } His CAA, CAG } Gln	CGU, CGC, CGA, CGG } Arg	U C A G
A	AUU, AUC, AUA } Ile AUG Met	ACU, ACC, ACA, ACG } Thr	AAU, AAC } Asn AAA, AAG } Lys	AGU, AGC } Ser AGA, AGG } Arg	U C A G
G	GUU, GUC, GUA, GUG } Val	GCU, GCC, GCA, GCG } Ala	GAU, GAC } Asp GAA, GAG } Glu	GGU, GGC, GGA, GGG } Gly	U C A G

© CSLS/The University of Tokyo

Topic # 30 31 Dynamic Programming

Block Game

m blocks

n blocks

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

Table "**R**"

3+4,
$R_{3,4} = W$
$R_{3,4} = L$
$R_{n,m}$

	0	1	2	3	4	5	6	7	8	9	10
0		W									
1	W	W									
2											
3											
4											
5											
6											
7											
8											
9											

10											

$R_{0,1}$ $R_{1,0}$ $R_{1,1}$

(1,0), (1,0), 1,1)
(2,0) → (1,0) = W
(0,2) → (0,1) = W
(2,1) → |
(1,1), (2,0), (1,0)
(2,0) = L
(1,2) = L
$R_{2,2}$ (2,2) →
(2,1), (1,2), (1,1)=W

Topic #31 Dynamic Programming

Block Game

(i-1, j), (i-1, j-1), (i, j-1)

	0	1	2	3	4	5	6	7	8	9	10
0	L	W	L	W	L	W	L	W	L	W	L
1	W	W	W	W	W	W	W	W	W	W	W
2	L	W	L	W	L	W	L	W	L	W	L
3	W	W	W	W	W	W	W	W	W	W	W
4	L	W	L	W	L	W	L	W	L	W	L
5	W	W	W	W	W	W	W	W	W	W	W
6	L	W	L	W	L	W	L	W	L	W	L
7	W	W	W	W	W	W	W	W	W	W	W
8	L	W	L	W	L	W	L	W	L	W	L
9	W	W	W	W	W	W	W	W	W	W	W
10	L	W	L	W	L	W	L	W	L	W	L

```
BLOCKS (n, m)
R0,0 = L
for i ← 1 to n
    if Ri-1,0 = W
        Ri,0 ← L
else
        Ri,0 ← W
for j  1 to m
    if R0,j-1 = W
        R0,j ← L
else
        R0,j ← W
for i ← 1 to n
    for j ← 1 to m
        if Ri-1,j-1 = W and Ri,j-1 = W and Ri-1,j = W
            Ri,j ← L
else
            Ri,j ← W
return Rn,m
```

FASTBLOCK(n, m)

1. if n and m are both even
2. return L
3. else
4. return W

Topic # 32 Algorithm for Dynamic Programming

Block Game Algorithm

```
BLOCKS (n, m)
R0,0 = L
for i ← 1 to n
    if Ri−1,0 = W
        Ri,0 ← L
    else
        Ri,0 ← W
    for j  1 to m
        if R0,j−1 = W
        R0,j ← L
    else
        R0,j ← W
```

$i = 2$

$R_{2-1,0} = R_{1,0} = W$

$j = 2$

$R_{0,2-1} = R_{0,1} = W$

```
BLOCKS (n, m)
for i ← 1 to n
for j ← 1 to m
if Ri−1,j−1 = W and Ri,j−1 = W and
    Ri−1,j = W
        Ri,j ← L
    else
        Ri,j ← W
return Rn,m
```

$i = 2,\ j = 2$

$R_{2-1,2-1}$, $R_{2,\ 2-1}$, $R_{2-1,2}$

$R_{1,1} = W$
$R_{2,1} = W$
$R_{1,2} = W$

FASTBLOCK(*n, m*)

1. if *n* and *m* are both even
2. return *L*
3. else
4. return *W*

$R_{n,m}$

$R_{2,2} = L$

$R_{4,4} = L$

$R_{4,5} = W$

Topic # 33 Restriction Mapping

***EcoR*I (*Escherichia coli*)**

GAATTC

ATGTTTGCATTACGATAGAATTCCGTCAAAGTGCTAG

TACAAACGTAATGCTATCTTAAGGCAGTTTCACGATC

GCCGTTATACGCTGGATTTAAATTGCTGTGAAATGGT

CGGCAATATGCGACCTAAATTTAACGACACTTTACCA

TACTGCCAAGACCGAATTCCTGCGAGTGCTGAAACG

ATGACGGTTCTGGCTTAAGGACGCTCACGACTTTGC

GCGATATTACGAATGTGCTTACAGCACCGAATTCATC

CGCTATAAAGCTTACACGAATGTCGTGGCTTAAGTAG

ATGTTTGCATTACGATAGAATTCCGTCAAAGTGCTAG

TACAAACGTAATGCTATCTTAAGGCAGTTTCACGATC

GCCGTTATACGCTGGATTTAAATTGCTGTGAAATGGT

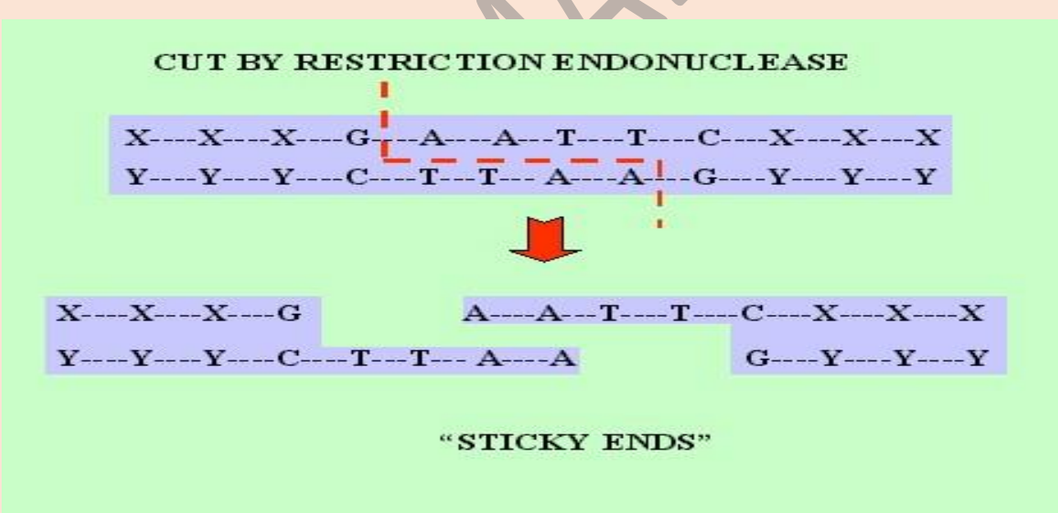
CGGCAATATGCGACCTAAATTTAACGACACTTTACCA

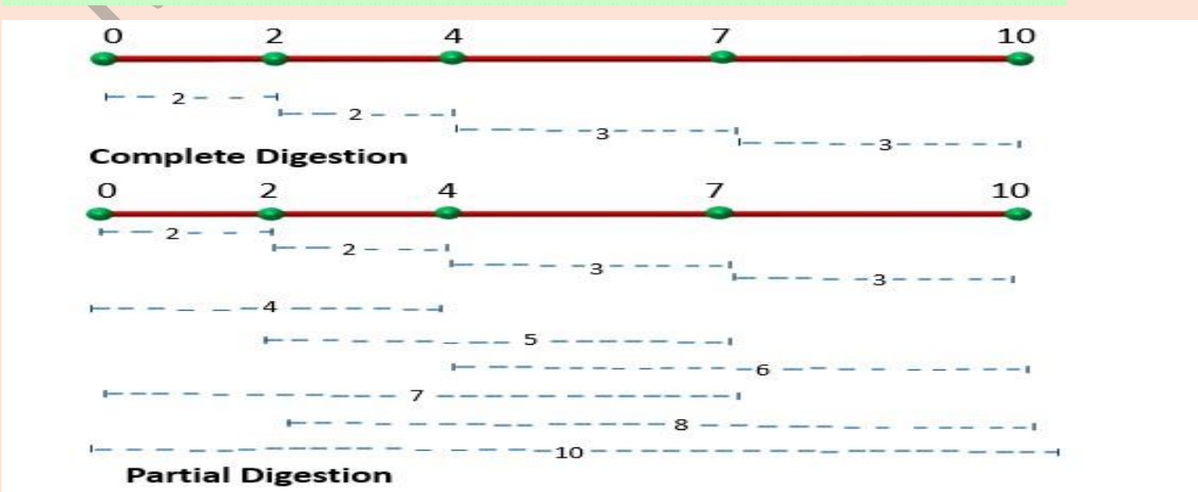
TACTGCCAAGACCGAATTCCTGCGAGTGCTGAAACG

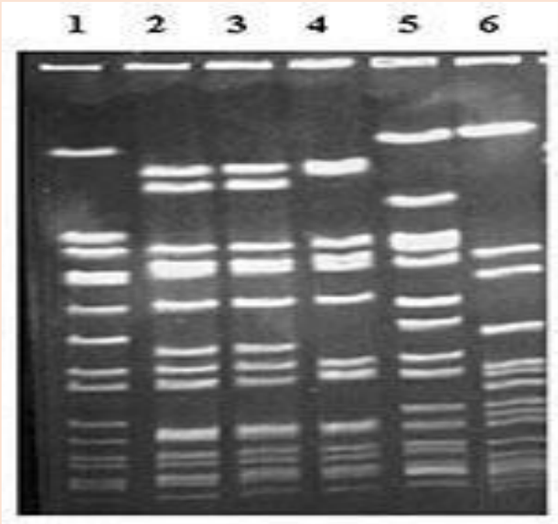
ATGACGGTTCTGGCTTAAGGACGCTCACGACTTTGC

GCGATATTACGAATGTGCTTACAGCACCGAATTCATC

CGCTATAAAGCTTACACGAATGTCGTGGCTTAAGTAG







{2, 2, 2, 3, 3, 4, 5}

If $X = \{x_1 = 0, x_2, \ldots, x_n\}$

$\Delta X = \{x_j - x_i : 1 \leq i < j \leq n\}$

X={0, 2, 4, 7, 10}, then ΔX={2, 2, 3, 3, 4, 5, 6, 7, 8, 10},

Representation of ΔX

	0	2	4	7	10
0		2	4	7	10
2			2	5	8
4				3	6
7					3
10					

The element at (i, j) in the table is the value $x_j - x_i$ for $1 \leq i < j \leq n$.

Topic # 34 Restriction Mapping

{2, 2, 2, 3, 3, 4, 5}

If $X = \{x_1 = 0, x_2, \ldots, x_n\}$

$\Delta X = \{x_j - x_i : 1 \leq i < j \leq n\}$

X={0, 2, 4, 7, 10}, then ΔX={2, 2, 3, 3, 4, 5, 6, 7, 8, 10},

Representation of ΔX

	0	2	4	7	10
0		**2**	**4**	**7**	**10**
2			**2**	**5**	**8**
4				**3**	**6**
7					**3**
10					

The element at (*i, j*) in the table is the value $x_j - x_i$ for $1 \le i < j \le n$.

Topic # 35 Partial Digest Problem

Partial Digest Problem:

Given all pairwise distances between points on a line, reconstruct the positions of those points. Input: The multiset of pairwise distances L, containing $\binom{n}{2}$ integers.

Output: A set *X*, of *n* integers, such that $\Delta X = L$

ΔA is equal to $\Delta(A \oplus \{v\})$, where $A \oplus \{v\}$ is defined to be $\{a + v : a \in A\}$,

Also $\Delta A = \Delta(-A)$, where $-A = \{-a : a \in A\}$

$A = \{0, 2, 4, 7, 10\}$, $\Delta(A \oplus \{100\}) = \{100, 102, 104, 107, 110\}$, and $-A = \{-10, -7, -4, -2, 0\}$

{0, 1, 3, 8, 9, 11, 12, 13, 15} and {0, 1, 3, 4, 5, 7, 12, 13, 15}

	0	1	3	4	5	7	12	13	15
0		**1**	**3**	**4**	**5**	**7**	**12**	**13**	**15**
1			**2**	**3**	**4**	**6**	**11**	**12**	**14**

MUHAMMAD IMRAN

	0	1	3	8	9	11	12	13	15
0		1	3	8	9	11	12	13	15
1			2	7	8	10	11	12	14
3				5	6	8	9	10	12
8					1	3	4	5	7
9						2	3	4	6
11							1	2	4
12								1	3
13									2
15									

3				1	2	4	9	10	12
4					1	3	8	9	11
5						2	7	8	10
7							5	6	8
12								1	3
13									2
15									

$\{1_4, 2_4, 3_4, 4_3, 5_2, 6_2, 7_2, 8_3, 9_2, 10_2, 11_2, 12_3, 13, 14, 15\}$

$U \oplus V = \{u + v : u \in U, v \in V\}$

$U \ominus V = \{u - v : u \in U, v \in V\}$

U = {6, 7, 9} and V = {−6, 2, 6}

U ⊕ V	-6	2	6
6	0	8	12
7	1	9	13
9	3	11	15

U ⊖ V	-6	2	6
6	12	4	0
7	13	5	1
9	15	7	3

BRUTEFORCEPDP(L, n)

1. M maximum element in L
2. for every set of n − 2 integers $0 < x_2 < \cdots < x_{n-1} < M$
3. X $\{0, x_2, \ldots, x_{n-1}, M\}$
4. Form ΔX from X
5. **if** $X = L$
6. **return** X
7. **output** "No Solution"

AMMAD IMRAN

ANOTHERBRUTEFORCEPDP(*L*, *n*)

1. M maximum element in L
2. **for** every set of n−2 integers $0 < x_2 < \cdots < x_{n-1} < M$ from L
3. X $\{0, x_2, \ldots, x_{n-1}, M\}$
4. Form ΔX from X
5. if $X = L$
6. **return** X
7. **output** "No Solution"

Topic # 36 Practical Restriction Mapping Algorithm

Brute Force Algorithm

Largest distance in "*L*"

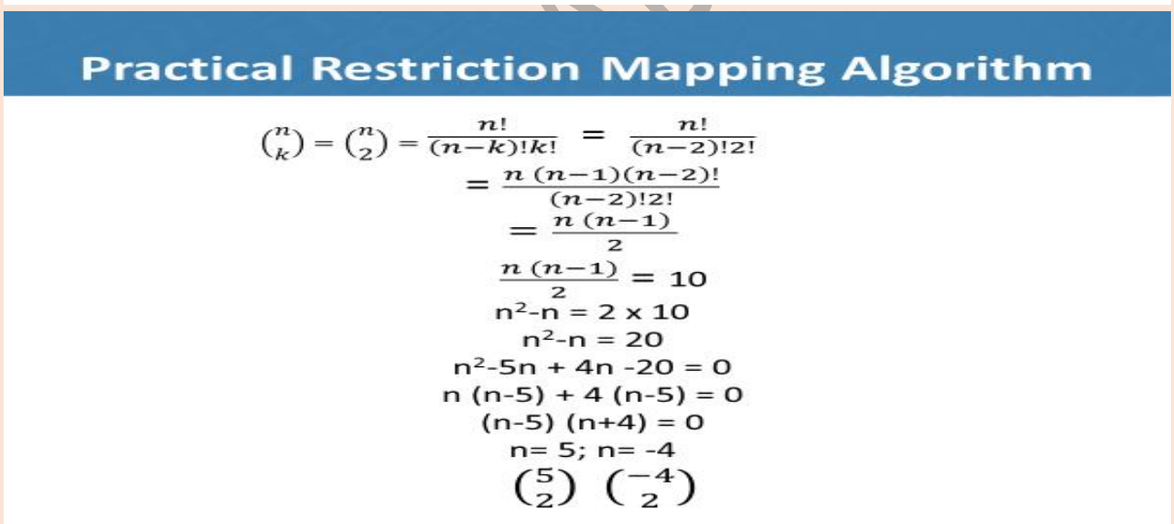
Outermost points of "*X*"

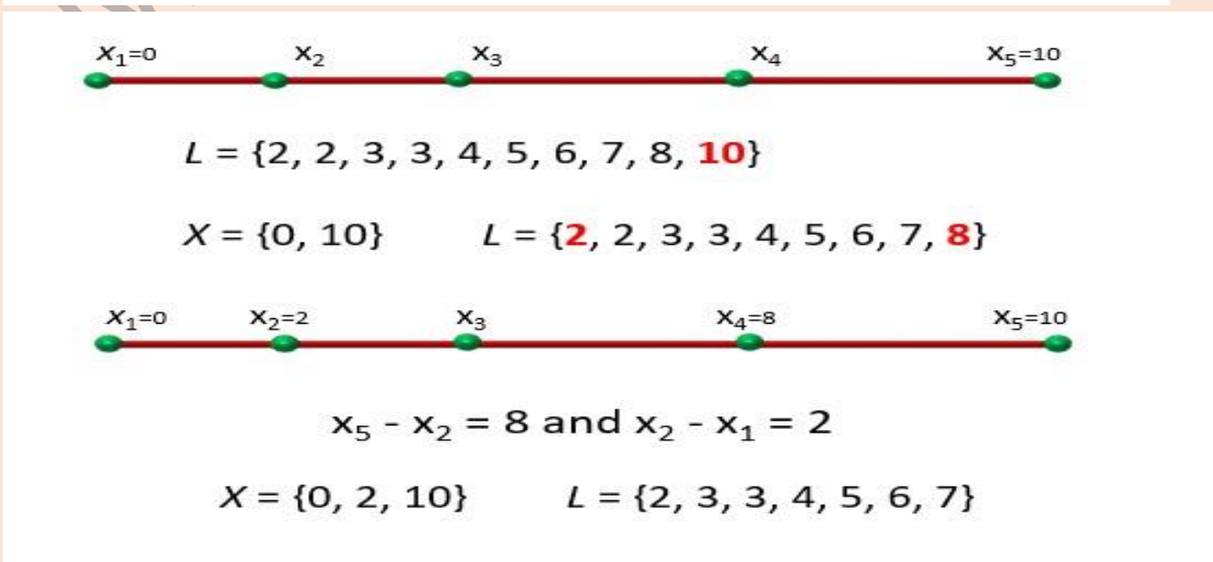
Remaining distance "δ"

$L = \{2, 2, 3, 3, 4, 5, 6, 7, 8, 10\}$

Size of L is $\binom{n}{2} = \frac{n(n-1)}{2} = 10$ where is "*n*" is number of points in the solutions.

Here "*n*" is 5 and positions of '*X*' as $x_1 = 0$, x_2, x_3, x_4 and x_5.



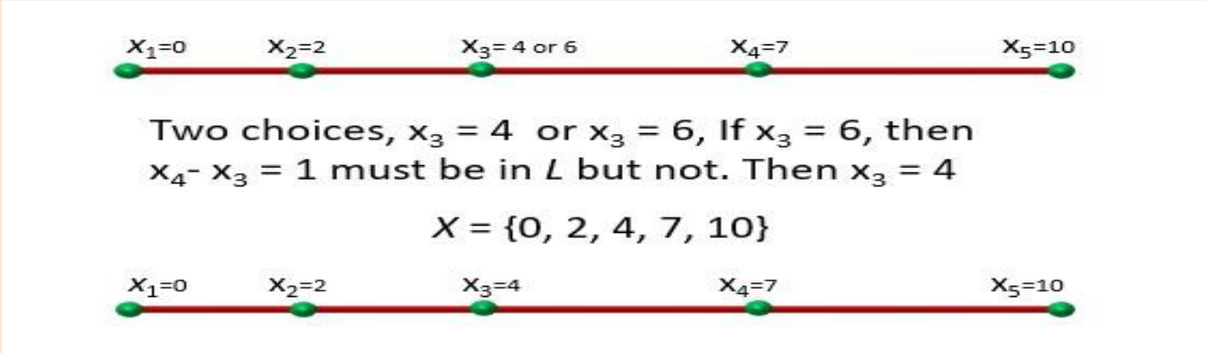


$X = \{0, 2, 10\}$ $L = \{2, \mathbf{3}, 3, 4,$

$5, 6, \mathbf{7}\}$ $x_4 = 7$ or $x_3 = 3$

If $x_3 = 3$ then $x_3 - x_2 = 1$ must be in L, but not, so x_4 must be 7, $x_5 - x_4 = 3$, $x_4 - x_2 = 5$, and $x_4 - x_1 = 7$ from L

$X = \{0, 2, 7, 10\}$ $L = \{2, 3, \mathbf{4}, \mathbf{6},\}$



Topic # 37 Partial Digest Algorithm

Brute Force Algorithm

List of pairwise distances, L, and uses the function DELETE(y, L)

Value y from L, notation (y,X) to denote the multiset of distances

For example,

$\Delta(2,\{1, 3, 4, 5\}) = \{1, 1, 2, 3\}$

PARTIALDIGEST (L)

1. *width* Maximum element in L
2. DELETE (*width*, L)
3. X $\{0, width\}$
4. PLACE (L,X)

PLACE (L,X)

1. if L is empty
2. output X
3. return
4. y Maximum element in L
5. if Δ (y,X) $\subseteq$ L
6. 6 Add y to X and remove lengths Δ (y,X) from L
7. PLACE (L,X)
8. Remove y from X and add lengths (y,X) to L
9. if ($width - y$,X) $\subseteq$ L
10. Add $width - y$ to X and remove lengths ($width - y$,X) from L
11. PLACE (L,X)
12. Remove $width - y$ from X and add lengths ($width - y$,X) to L
13. return

Topic # 38 Partial Digest Algorithm

PARTIALDIGEST (L)

```
PARTIALDIGEST (L)
1.  width ←— Max element in L        10
2.  DELETE (width, L)                10, L
3.  X ←— {0, width}                  0, 10
4.  PLACE (L,X)
```

***X* = {0, 10}**

```
PLACE (L,X)
1.  if L is empty
2.      output X
3.      return
4.  y ←— Maximum element in L
5.  if Δ (y,X) ⊆ L
6   Add y to X and remove
    lengths Δ (y,X) from L
7.      PLACE (L,X)
8.  Remove y from X and add
    lengths (y,X) to L
```

L = {2, 2, 3, 3, 4, 5, 6, 7, 8, **10**}
***L* = {2, 2, 3, 3, 4, 5, 6, 7, 8}**
***X* = {0, 10}**
8
Δ (*y*,*X*) = 8 (0, 10) = (8, 10)

```
9.  if (width − y,X) ⊆ L
10. Add width − y to X and
    remove lengths (width − y,X) from L
11.     PLACE (L,X)
12. Remove width − y from X and
    add lengths (width − y,X) to L
13.     return
```

***L* = {2, 2, 3, 3, 4, 5, 6, 7, 8}**
***X* = {0, 10}**
width − *y*,*X* = 10-8 (0,10)
= 2 (0, 10)
= (2, 8)

***L* = {2, 3, 3, 4, 5, 6, 7}**
***X* = {0, 2, 10}**

PLACE (L,X)

1. if L is empty
2. **output** X
3. **return**
4. y ⟵ Maximum element in L
5. **if** Δ (y,X) $\subseteq$ L
6. Add y to X and remove lengths Δ (y,X) from L
7. PLACE (L,X)
8. Remove y from X and add lengths (y,X) to L

L = {2, 3, 3, 4, 5, 6, 7}
X = {0, 2, 10}
7
Δ (y,X) = 7 (0, 2, 10) = (7, 5, 3)

L = {2, 3, 3, 4, 5, 6,}
X = {0, 2, 7, 10}

PLACE (L,X)

1. if L is empty
2. **output** X
3. **return**
4. y ⟵ Maximum element in L
5. **if** Δ (y,X) $\subseteq$ L
6. Add y to X and remove lengths Δ (y,X) from L
7. PLACE (L,X)
8. Remove y from X and add lengths (y,X) to L

L = {2, 3, 3, 4, 5, 6}
X = {0, 2, 7, 10}
6
Δ (y,X) = 6 (0, 2, 7, 10) = (6, 4, 1, 4)

```
PLACE (L,X)
1.  if L is empty
2.      output X
3.      return
4.  y ⟵ Maximum element in L
5.  if Δ (y,X) ⊆ L
6   Add y to X and remove
    lengths Δ (y,X) from L
7.      PLACE (L,X)
8.  Remove y from X and add
    lengths (y,X) to L
```

L = {2, 3, 3, 4, 5, 6}
X = {0, 2, 7, 10}
6
Δ (y,X) = 6 (0, 2, 7, 10) = (6, 4, 1, 4)

Topic # 39 Regulatory Motifs in DNA Sequences

Sequence Motifs

Biological function

Nucleases and transcription factors

Processes at RNA level

Specific sequence located upstream of genes TCGGGGATTTCC

NF-kB binding sites (nuclear factor kappa-light-chain-enhancer of activated B cells)

Regulatory motifs, transcription factors

Set of upstream regions in genes in the genome, each region containing one NF-kB

sites Suppose we do not know either location or sequence of NF-kB sites

"The Gold Bug" by Edgar Allan provided some clue of finding DNA motifs, one of the character find parchment written below

53++!305))6*;4826)4+.)4+);806*;48!8‘60))85;]8*:+*8!83(88)5*!;46(;88*96*?;8)
*+(;485);5 *!2:*+(;4956*2(5*-)8‘8*;4069285) ;)6!8)
4++;1(+9;48081;8:8+1;48!85;4)485!
528806*81(+9;48;(88;4(+?34;48)4+;161;:188;+?;

"**;48**" codes for "**THE**"
53++!305))6*THE26)H+.)H+)TE06*THE!E‘60))E5T]E*:+*E!E3(EE)5*!TH6(T
EE*96*?TE
)* +(THE5) T5*!2:*+(TH956*2(5*-H)E‘E*T
H0692E5)T)6!E)H++T1(+9THE0E1TE:E+1THE!E5TH)HE5!52EE06*E1(+9TH
ET(EETH(+?3HTHE) H+T161T:1EET+?T

";" for "T"

"4" for "H"

"8" for "E"

Topic # 40 Profiles 1

MUHAMMAD IMRAN

Conserved Pattern

32 nucleotide

7 sequences

1. CGGGGCTGGGTCGTCACATTCCCCTTTCGATA
2. TTTGAGGGTGCCCAATAACCAAAGCGGACAAA
3. GGGATGCCGTTTGACGACCTAAATCAACGGCC
4. AAGGCCAGGAGCGCCTTTGCTGGTTCTACCTG
5. AATTTTCTAAAAAGATTATAATGTCGGTCCTC
6. CTGCTGTACAACTGAGATCATGCTGCTTCAAC
7. TACATGATCTTTTGTGGATGAGGGAATGATGC

Figure 1

P = ATGCAACT

l = 8

1. CGGGGCTATGCAACTGGGTCGTCACATTCCCCTTTCGATA
2. TTTGAGGGTGCCCAATAAATGCAACTCCAAAGCGGACAAA
3. GGATGCAACTGATGCCGTTTGACGACCTAAATCAACGGCC
4. AAGGATGCAACTCCAGGAGCGCCTTTGCTGGTTCTACCTG
5. AATTTTCTAAAAAGATTATAATGTCGGTCCATGCAACTTC
6. CTGCTGTACAACTGAGATCATGCTGCATGCAACTTTCAAC
7. TACATGATCTTTTGATGCAACTTGGATGAGGGAATGATGC

Figure 2

P = ATGCAACT

l = 8

1. CGGGGCTATGCAACTGGGTCGTCACATTCCCCTTTCGATA
2. TTTGAGGGTGCCCAATAAATGCAACTCCAAAGCGGACAAA
3. GGATGCAACTGATGCCGTTTGACGACCTAAATCAACGGCC
4. AAGGATGCAACTCCAGGAGCGCCTTTGCTGGTTCTACCTG
5. AATTTTCTAAAAAGATTATAATGTCGGTCCATGCAACTTC
6. CTGCTGTACAACTGAGATCATGCTGCATGCAACTTTCAAC
7. TACATGATCTTTTGATGCAACTTGGATGAGGGAATGATGC

Figure 3

P = ATGCAACT

l = 8

7 x (32 + 8) = 280 nucleotides

Probability = $280/4^8$ = 0.004

1. CGGGGCTATcCAgCTGGGTCGTCACATTCCCCTTTCGATA
2. TTTGAGGGTGCCCAATAAggGCAACTCCAAAGCGGACAAA
3. GGATGgAtCTGATGCCGTTTGACGACCTAAATCAACGGCC
4. AAGGAaGCAACcCCAGGAGCGCCTTTGCTGGTTCTACCTG
5. AATTTTCTAAAAAGATTATAATGTCGGTCCtTGgAACTTC
6. CTGCTGTACAACTGAGATCATGCTGCATGCcAtTTTCAAC
7. TACATGATCTTTTGATGgcACTTGGATGAGGGAATGATGC

Figure 4

MUHAMMAD IMRAN

Topic # 41 Profiles 2

Conserved Pattern

18 sequences
NF-kB
TCGGGGATTTCC
***l* = 12**

Alignment

```
1-  TCGGGGATTTCA
2-  ACGGGGATTTTT
3-  TCGGTACTTTAC
4-  TTGGGGACTTTT
5-  CCGGTGATTCCC
6-  GCGGGGAATTTC
7-  TCGGGGATTCCT
8-  TCGGGGATTCCT
9-  TAGGGGAACTAC
10- TCGGGTATAAAC
11- TCGGGGGTTTTT
12- CCGGTGACTTAC
13- CCAGGGACTCCC
14- AAGGGGACTTCC
15- TTGGGGACTTTT
16- TTTGGGAGTCCC
17- TCGGTGATTTCC
18- TAGGGGAAGACC
```

Profile

```
A: 2 3 1 0  0 1 16 3 1 2 4 1
T: 12 3 1 0  4  1 0 9 15 11 5 6
G: 1 0 16 18 14 16 1 1 1  0  0 0
C: 3 12 0  0 0  0  1 5 1  5 9 11
```

Consensus

```
TCGGGGATTTCC
```

score = 12, 12, 16, 18, 14, 16, 16, 9, 15, 11, 9, 11= 159

1- position 8 - Sequence 1

2- position 19 - Sequence 2

3- position 3- Sequence 3

4- position 5- Sequence 4

5- position 31- Sequence 5

6- position 27- Sequence 6

7- position 15- Sequence 7

1- CGGGGCTATcCAgCTGGGTCGTCACATTCCCCTT

2- TTTGAGGGTGCCCAATAAggGCAACTCCAAAGCGGACAAA

3- GGATGgAtCTGATGCCGTTTGACGACCTA

4- AAGGAaGCAACcCCAGGAGCGCCTTTGCTGG

5- AATTTTCTAAAAAGATTATAATGTCGGTCCtTGgAACTTC

6- CTGCTGTACAACTGAGATCATGCTGCATGCcAtTTTCAAC

MUHAMMAD

7-
TACATGATCTTTTGATGgcACTTGGATGAGGGAATGATGC

Figure 6

Topic # 42 Profiles 3

Conserved Pattern

		1 2 3 4 5 6 7 8
Alignment		A T C C A G C T
		G G G C A A C T
		A T G G A T C T
		A A G C A A C C
		T T G G A A C T
		A T G C C A T T
		A T G G C A C T
Profile	A	5 1 0 0 5 5 0 0
	T	1 5 0 0 0 1 1 6
	G	1 1 6 3 0 1 0 0
	C	0 0 1 4 2 0 6 1
Consensus		A T G C A A C T

Figure 7

If s = (8, 19, 3, 5, 31, 27, 15) then

Score(s) = 5 + 5 + 6 + 4 + 5 + 5 + 6 + 6

= 42 Set of t DNA sequences

"n" nucleotides one position in each of

these "t" sequences s = (s_1, s_2, . . . , s_t),

with $1 \le s_i \le n - l + 1$ *l*-mers can be

compiled into $t \times l$ *alignment matrix*

whose (*i*, *j*)th element is the nucleotide in the $s_i + j - 1$th element in the *i*th sequence of figure 7

Topic # 43 Motif Finding Problem

Profile Matrix

If P(s) denotes profile matrix, starting from *s*

$M_{P(s)}(j)$-largest count in column *j* of P(s)

Alignment		A T C C A G C T
		G G G C A A C T
		A T G G A T C T
		A A G C A A C C
		T T G G A A C T
		A T G C C A T T
		A T G G C A C T
Profile	A	5 1 0 0 5 5 0 0
	T	1 5 0 0 0 1 1 6
	G	1 1 6 3 0 1 0 0
	C	0 0 1 4 2 0 6 1
Consensus		A T G C A A C T

$M_{P(s)}(1) = 5$
$M_{P(s)}(2) = 5$
$M_{P(s)}(3) = 6$
$M_{P(s)}(4) = 4$
$M_{P(s)}(5) = 5$
$M_{P(s)}(6) = 5$
$M_{P(s)}(7) = 6$
$M_{P(s)}(8) = 6$

MUHAMMAD IMRAN

For the starting positions in above figure, Score(*s*,*DNA*) =

5 + 5 + 6 + 4 + 5 + 5 + 6 + 6 = 42. Score(*s*,*DNA*) can be used to measure

The strength of a profile corresponding to the starting positions s. consensus score of $l \cdot t$ corresponds to the best possible alignment *Score(s,DNA)* consensus score is defined to be *Score(s,DNA)* = $\sum_{j=1}^{l} MP_{(s)}(j)$ Score(s,DNA) = 5 + 5 + 6 + 4 + 5 + 5 + 6 + 6 = 42 consensus score of $l \cdot t$ corresponds to the best possible alignment

A consensus score of $l.t/4$, is worst possible alignment

Motif Finding Problem:

Given a set of DNA sequences, find a set of l-mers, one from each sequence, that maximizes the consensus score.

Input: A $t \times n$ matrix of *DNA*, and, the length of the pattern to find. Output: An array of t starting positions s =l (s_1, s_2, . . . , s_t) maximizing Score(*s*,*DNA*).

Topic # 44 Motif Finding Problem 1

Median String

Median string

Given two *l*-mers *v* and *w*, can compute the *Hamming distance* between them, $d_H(v,w)$, as the number of positions that differ in the two strings

A T T G T C

: x : x : :

A C T C T C

s = (s_1, s_1, . . . , s_t)

v is some *l*-mer $d_H(v, s)$ to denote the total Hamming distance between *v* and the *l*-mers starting at positions s: $d_H(v, s) = \sum_{i=1}^{t} d_H(v, si)$ where $d_H(v, s_i)$ is the Hamming distance between *v* and the *l*-mer that starts at s_i in the *i*th DNA sequence TotalDistance(*v*,*DNA*) = $\min_s(d_H(v, s)$

Finding Total Distance(*v*,*DNA*) is a simple problem:

find the best match for *v* in the first DNA sequence (i.e., a position minimizing $d_H(v, s_1)$ for $1 \le s_1 \le n-l+1$), then the best match in the second sequence and so on

Median string for DNA as the string v that minimizes TotalDistance(*v*,*DNA*); this minimization is performed over all 4^l strings *v* of length *l*.

Median String Problem:

Given a set of DNA sequences, find a median string. **Input**: A $t \times n$ matrix DNA, and *l*, the length of the pattern to find

Output: A string v of *l* nucleotides that minimizes TotalDistance(*v*,*DNA*) over all strings of that length

Double minimization: finding a string *v* that minimizes TotalDistance(*v*,*DNA*), which is in turn the smallest distance among all choices of starting points *s* in the DNA sequences.

Min min

d_H(v, s) all choices of

all choices *l*-mers *v*

starting positions *s*

The Median String problem- Minimization problem

The Motif Finding problem- Maximization problem

Computationally equal

Let *s* be a set of starting positions with consensus score Score(*s*,*DNA*), and let *w* be the consensus string of the corresponding profile. Then $d_H(w, s) = lt - Score(s,DNA)$

Hamming distance between the consensus

string *w* and each of the seven implanted patterns is 2, and $d_H(w, s)$ =2 x 7 = **14**

= 7 x 8 − 42 = **14**

Topic # 45 Motif Finding Problem 2

Median String

Median string

Given two *l*-mers *v* and *w*, can compute the *Hamming distance* between them, $d_H(v,w)$, as the number of positions that differ in the two strings

A T T G T C

: x : x : :

A C T C T C

s = (s_1, s_1, . . . , s_t) *v* is some *l*-mer $d_H(v, s)$ to denote the total Hamming distance between *v* and the *l*-mers starting at positions s: $d_H(v, s) = \sum_{i=1}^{t} d_H(v, si)$ where $d_H(v, s_i)$ is the Hamming distance between *v* and the *l*-mer that starts at s_i in the *i*th DNA sequence

TotalDistance(*v*,*DNA*) = $\min_s(d_H(v, s)$

Finding Total Distance(*v*,*DNA*) is a simple problem:

find the best match for *v* in the first DNA sequence (i.e., a position minimizing $d_H(v, s_1)$ for 1
$\leq s_1 \leq n-l+1$), then the best match in the second sequence and so on

Median string for DNA as the string v that minimizes TotalDistance(*v*,*DNA*); this minimization is performed over all 4^l strings *v* of length *l*.

Median String Problem:

Given a set of DNA sequences, find a median string. **Input**: A $t \times n$ matrix DNA, and *l*, the length of the pattern to find

Output: A string v of *l* nucleotides that minimizes TotalDistance(*v*,*DNA*) over all strings of that length

Double minimization: finding a string *v* that minimizes TotalDistance(*v*,*DNA*), which is in turn the smallest distance among all choices of starting points *s* in the DNA sequences.

Min min $d_H(v, s)$

all choices of all

choices *l*-mers *v*

starting positions *s*

The Median String problem- Minimization problem

The Motif Finding problem- Maximization problem

Computationally equal

Let *s* be a set of starting positions with consensus score Score(*s*,*DNA*), and let *w* be the consensus string of the corresponding profile. Then $d_H(w, s) = lt - Score(s,DNA)$

Hamming distance between the consensus

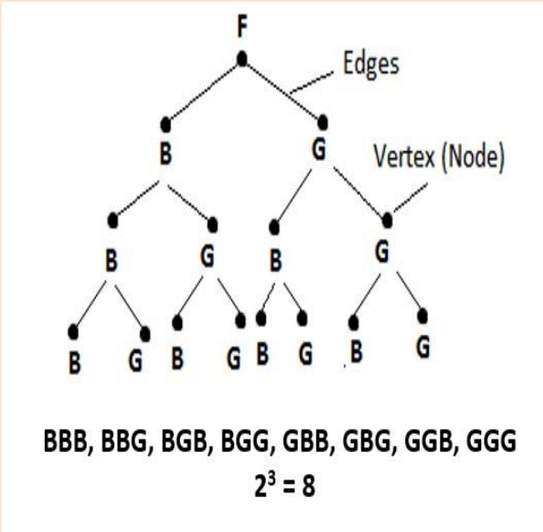
MUHAMMAD IMRAN

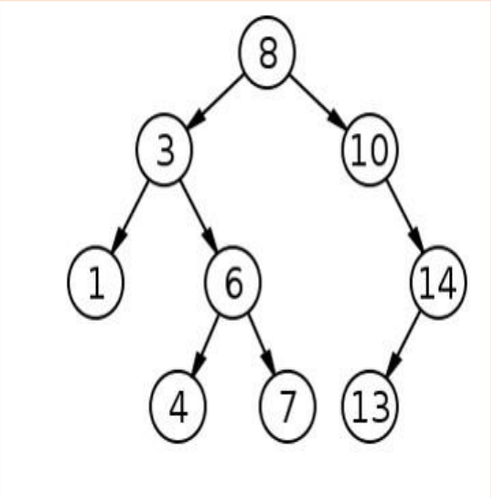
string w and each of the seven implanted patterns is 2, and $d_H(w, s)$ =2 x 7 = **14**

= 7 x 8 − 42 = **14**

Topic # 45 Search Trees-Introduction

Introduction





Topic # 46 Search Trees Best Alternative

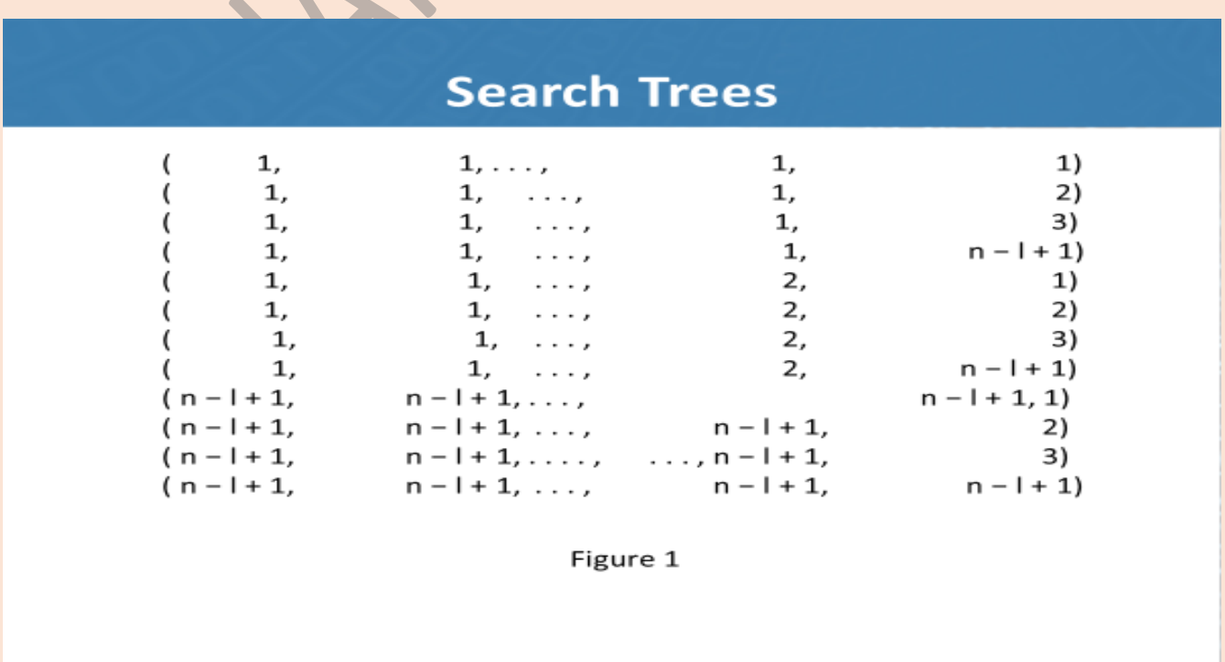
Median String and

Motif Finding Problems

Number of alternatives

To find best one

$(n - l + 1)^t$



Search Trees

(1,	1, . . . ,	1,	1)
(1,	1, . . . ,	1,	2)
(1,	1, . . . ,	1,	3)
(1,	1, . . . ,	1,	n − l + 1)
(1,	1, . . . ,	2,	1)
(1,	1, . . . ,	2,	2)
(1,	1, . . . ,	2,	3)
(1,	1, . . . ,	2,	n − l + 1)
(n − l + 1,	n − l + 1, . . . ,		n − l + 1, 1)
(n − l + 1,	n − l + 1, . . . ,	n − l + 1,	2)
(n − l + 1,	n − l + 1, ... ,	. . . , n − l + 1,	3)
(n − l + 1,	n − l + 1, . . . ,	n − l + 1,	n − l + 1)

Figure 1

Figure 1

AA· · · AA

AA· · · AT

AA· · · AG

AA· · · AC

AA· · · TA

AA· · · TT

AA· · · TG

AA· · · TC

...

CC· · · GG

CC· · · GC

CC· · · CA

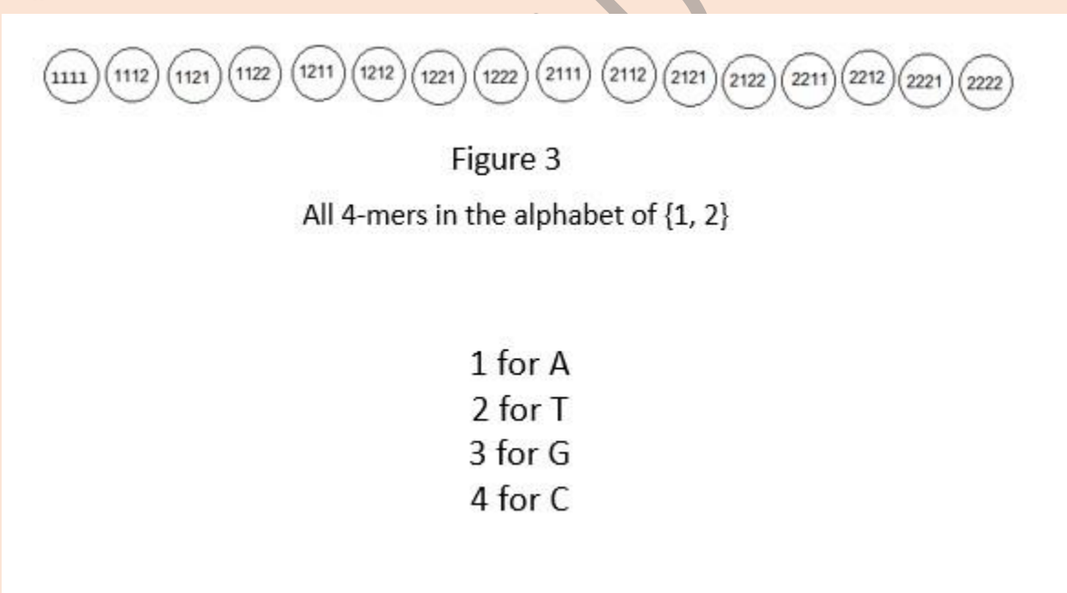
CC· · · CT

CC· · · CG

CC· · · CC

All 4^l

Figure 2



(1, 1, . . . , 1, 1)

(1, 1, . . . , 1, 2)

(1, 1, . . . , 1, 3)

(1, 1, . . . , 1, 4)

(1, 1, . . . , 2, 1)

(1, 1, . . . , 2, 2)

(1, 1, . . . , 2, 3)

(1, 1, .

. . , 2,

4) ...

(4, 4, . . . , 3, 3)

(4, 4, . . . , 3, 4)

(4, 4, . . . , 4, 1)

(4, 4, . . . , 4, 2)

(4, 4, . . . , 4, 3)

(4, 4, . . . , 4, 4)

1 for A

2 for T

3 for G

4 for C

Consider all k^L *L*-mers in a *k*-letter alphabet

For Motif Finding problem, $k = n-l+1$, For Median String problem, k = 4.

All 2^4 4-mers in the two-letter alphabet of 1 and 2.

Topic # 47 Algorithm for Search Trees 1

Search tree Algorithms

Next leave

All leaves

Preorder

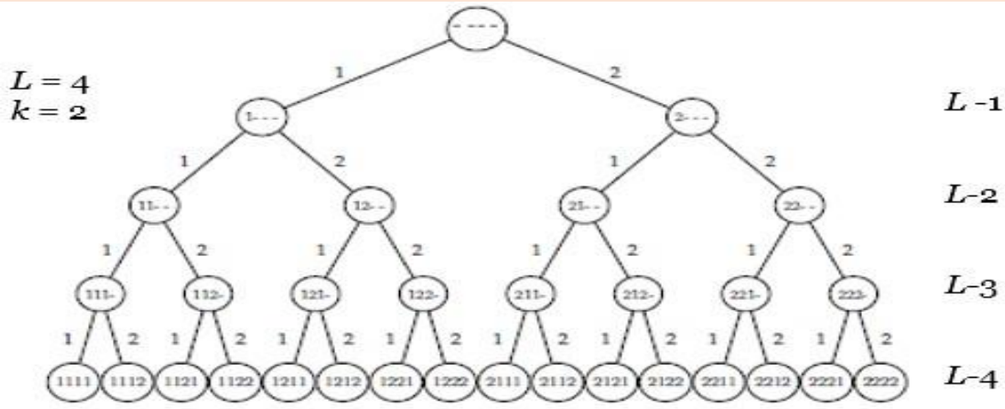
L-mer a = (a_1 a_2…a_L)

```
NEXTLEAF(a,L, k)
    for i ← L to 1
        if a_i < k
            a_i ← a_i + 1
            return a
        a_i ← 1
    return a
```

a = elements of *L*-mer
L = *l*-mer
k = number of elements

```
ALLLEAVES(L, k)
    a ← (1, . . . , 1)
    while forever
        output a
    a ← NEXTLEAF(a,L, k)
    if a = (1, 1, . . . , 1)
        return
```

MUHAMMAD IMRAN



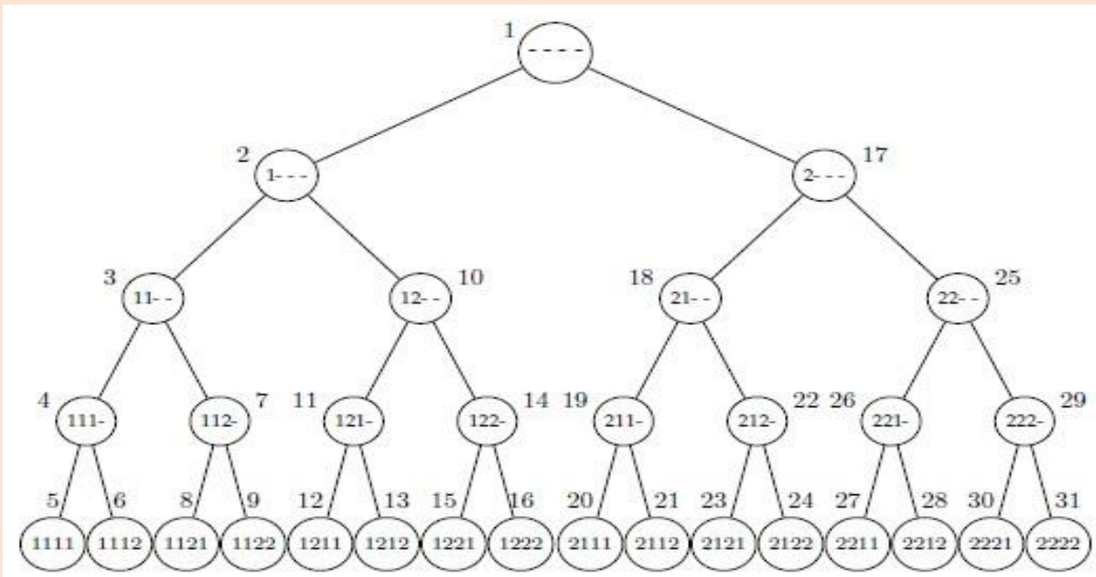
Motif Finding Problem , $L = t$ levels $k = n\text{-}l + 1$
Median String Problem , $L = l$ and $k = 4$
t = number of DNA sequences
n= length of each sequence, l = length of the profile

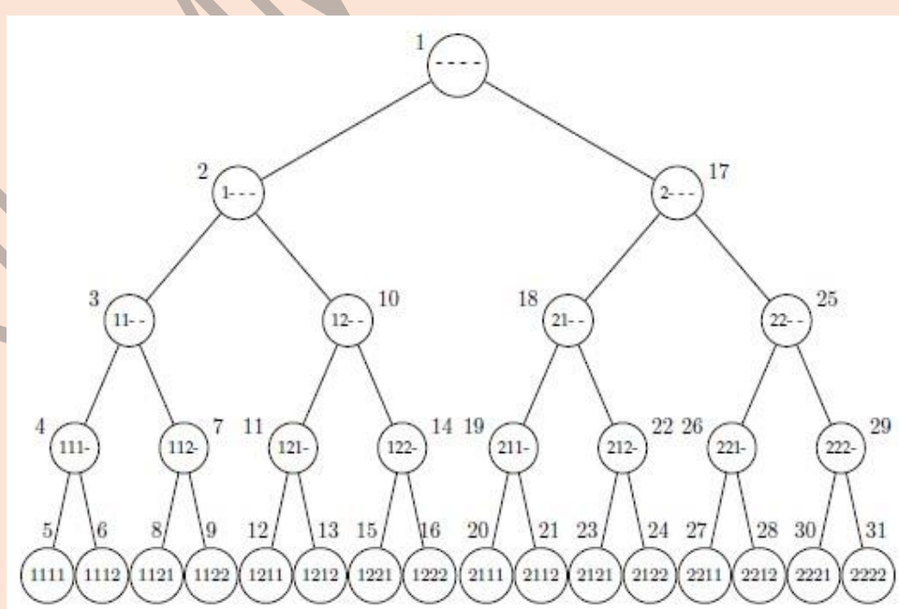
MUHAMMAD IM

Topic # 48 Algorithm for Search Trees 2

PREORDER(v)

1. output *v*
2. if *v* has children
3. PREORDER(left child of *v*)
4. PREORDER(right child of *v*)





L

=

4

k

=

2

1. (-,-,-,-)
2. (1,-,-,-)
3. (1,1,-,-)
4. (1,1,1,-)
5. (1,1,1,1)
6. (1,1,1,2)
7. (1,1,2,-)
8. (1,1,2,1)
9. (1,1,2,2)
10. (1,2,-,-)
11. (1,2,1,-)
12. (1,2,1,1)
13. (1,2,1,2)
14. (1,2,2,-)
15. (1,2,2,1)
16. (1,2,2,2)
17. (2,-,-,-)
18. (2,1,-,-)
19. (2,1,1,-)
20. (2,1,1,1)
21. (2,1,1,2)
22. (2,1,2,-)
23. (2,1,2,1)
24. (2,1,2,2)
25. (2,2,-,-)
26. (2,2,1,-)
27. (2,2,1,1)
28. (2,2,1,2)
29. (2,2,2,-)
30. (2,2,2,1)
31. (2,2,2,2)

Topic # 49 Next Vertex Algorithm

Traversing vertices

MUHAMMAD IMRAN

Input: a = (a_1, . . . , a_L) at

level i Output: Next vertex in

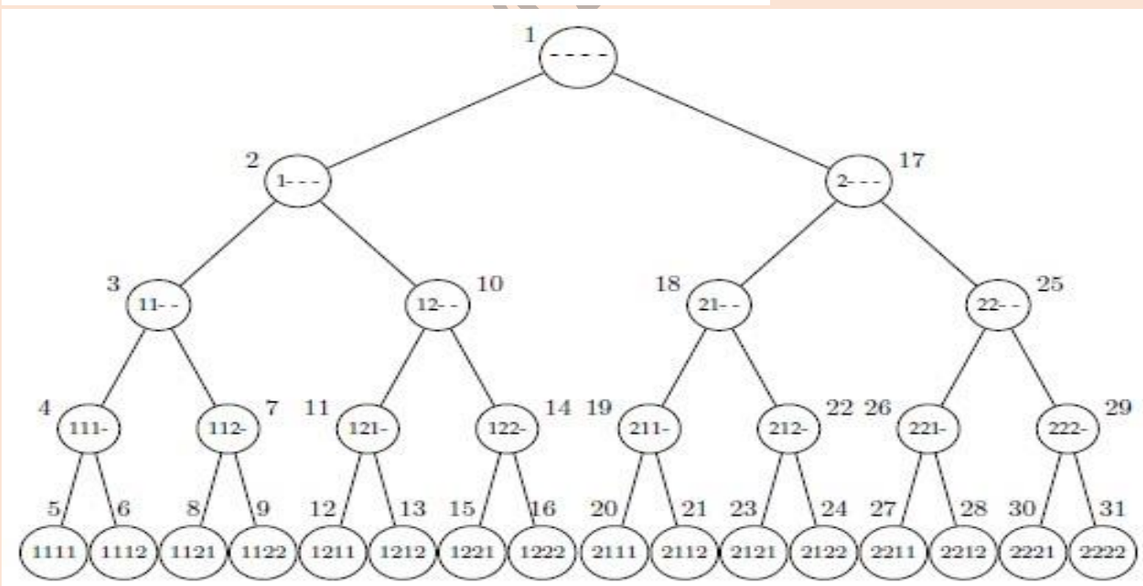
the tree values a_1, . . . , a_i and

ignores a_{i+1}, . . . , a_L

NEXTVERTEX takes inputs that are similar to NEXTLEAF, with the exception that the
"current leaf" is now the "current vertex," and uses

parameter i for vertices

```
NEXTVERTEX(a, i, L, k)
if i < L
    a_{i+1} ← 1
    return (a, i + 1)
else
    for j ← L to 1
        if a_j < k
            a_j ← a_j + 1
            return (a, j)
return (a, 0)
```



When $i < L$, NEXTVERTEX (**a**, i, L, k) moves down to the next lower level and explores that subtree of **a**. If $i = L$, NEXTVERTEX either moves along the lowest level as long as $a_L < k$ or jumps back up in the tree.

Topic # 50 Advanced Computing Approaches

Algorithm

- Some entity needs to carry out the steps specified by the algorithm
- Humans are generally slow

- A computer is less intelligent but can perform simple steps quickly and reliably
- Algorithm must be rephrased in programming language
- Pseudocode: language often used to describe algorithm
- Complex operations are grouped together into mini-algorithms called subgroups
- **Variable** is written as *x* or *total*
- An **array** of *n* elements is an ordered collection of *n* variables a_1, a_2,……..a_n
- An algorithm is a pseudocode is denoted by a name, followed by the list of arguments

Topic #51 ByPass Algorithm

Branch and Bound Algorithm

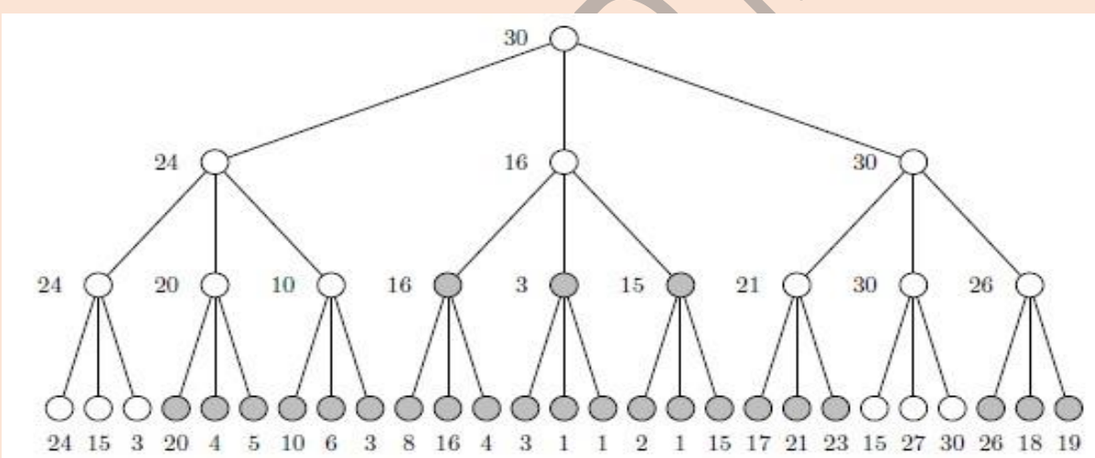
NEXTVERTEX Algorithm

BYPASS Algorithm

Skip the subtree rooted at

vertex (*a*, *i*) Increment a_i

(unless $a_i = k$



A tree that has uninteresting subtrees. The numbers next to a leaf represent the "score" for that L-mer. Scores at internal vertices represent the maximum score in the subtree rooted at that vertex. To improve the brute force algorithm, we can "prune" subtrees that do not contain highscoring leaves. For example, since the score of the very first leaf is 24, it does not make sense to analyze the 4th, 5th, or 6th leaves whose scores are 20, 4, and 5, respectively. Therefore, the subtree containing these vertices can be ignored.

```
BYPASS(a, i,L, k)
1.   for j ⟵ i to 1
2.       if a_j < k
3.           a_j ⟵ a_j + 1
4.           return (a, j)
5.   return (a, 0)
```

Topic # 52 Finding Motifs

Brute force approach

BRUTEFORCEMOTIFSEARCH(*DNA, t, n, l*)

1. *bestScore* 0
2. for each $(s_1, \ldots, s_t)$ from $(1, \ldots, 1)$ to $(n - l + 1, \ldots, n - l + 1)$
3. if *Score(s,DNA)* > *bestScore*
4. *bestScore* *Score(s,DNA)*

5 bestMotif $(s_1, s_2, \ldots, s_t)$

6. return bestMotif

$n-l+1$ choices for the first index (s_1), then for s_2, s_3) number of positions is $(n - l + 1)^t$

Score(*s,DNA*), which requires $O(l)$ operations-$O(ln^t)$.

BRUTEFORCEMOTIFSEARCHAGAIN(*DNA, t, n, l*)

1. s $(1, 1, \ldots, 1)$
2. *bestScore* *Score*(s,*DNA*)
3. while forever
4. s NEXTLEAF($s, t, n - l + 1$)
5. if *Score(s,DNA)* > *bestScore*
6. *bestScore* Score(s,DNA)
7. bestMotif $(s_1, s_2, \ldots, s_t)$
8. if s = (1, 1, . . . , 1)
9. return bestMotif

Topic # 53 Simple Motif Search Algorithm

Simple Motif Search Algorithm

SIMPLEMOTIFSEARCH(*DNA, t, n, l*)

MUHAMMAD IMRAN

1. s (1, . . . , 1)
2. *bestScore* 0
3. *i* 1
4. while *i* > 0
5. if *i* < t
6. (s, *i*) NEXTVERTEX(*s, i, t, n − l* + 1)
7. else
8. if *Score(s,DNA)* > *bestScore*
9. *bestScore* *Score(s,DNA)*
10. bestMotif (s_1, s_2, . . . , s_t)
11. (s, *i*) NEXTVERTEX(s, *i, t, n − l* + 1)
12. return bestMotif

Simple Motif Search Algorithm

Some sets of starting positions can be ruled out

If the first *i* of *t* starting positions [i.e., (s_1, s_2,... , s_i)]

Sequences *i*+1, *i*+2, . . . , t,

s = (s_1, s_2,... , s_t), define the partial consensus score, Score(*s, i, DNA*)- *i*×*l* alignment matrix

Partial consensus score for s_1, . . . , s_i, remaining *t*−*i* rows can only improve the consensus score by (*t* − *i*) · l

First *i* starting positions (s_1, . . . , s_1) could be at most Score(*s, i,DNA*)+(*t*−*i*) · *l*

Score(s, i,DNA)+ (t− i) · l

is less than the currently best score,

bestScore *t* − *i* sequences in the

sample

Score(*s, i,DNA*) + (*t* − *i*) · *l*

$(n - l + 1)^{t-i}$

Topic # 54 Branch and Bound Algorithm

Branch and Bound

Motif Search

BRANCHANDBOUNDMOTIFSEARCH(*DNA, t, n, l*)

1. s (1, . . . , 1)
2. bestScore 0
3. *i* 1
4. while *i* > 0
5. if *i* < t
6. *optimisticScore* *Score* (s, *i, DNA*)+

$(t - i) \cdot l$

if optimisticScore < bestScore

(s, i) BYPASS$(s, i, t, n - l + 1)$

else

(s, i) NEXTVERTEX$(s, i, t, n - l + 1)$

else

if *Score(s,DNA)* > *bestScore*

bestScore *Score*(s)

bestMotif $(s_1, s_2, \ldots, s_t)$

(s, i) NEXTVERTEX$(s, i, t, n - l + 1)$

return bestMotif

Topic # 55 Brute Force Algorithm

Brute Force

Median Search

BRUTEFORCEMEDIANSEARCH(*DNA, t, n, l*)

bestWord AAA· · · AA

bestDistance

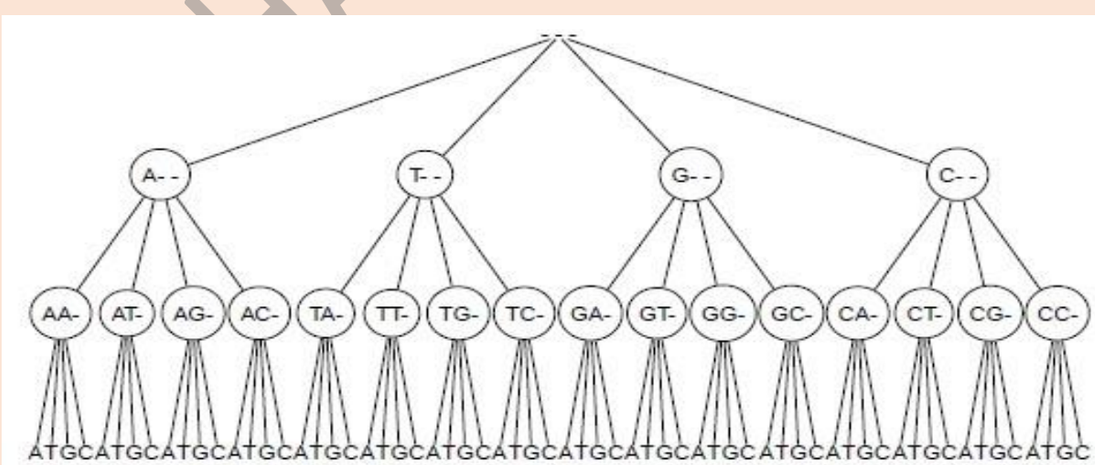
for each *l*-mer *word* from AAA...A to TTT...T

if TOTALDISTANCE(*word,DNA*) < *bestDistance*

bestDistance TOTALDISTANCE(word,DNA)

bestWord word

return *bestWord*



A search tree for the Median String problem. Each branching point can give rise to only four children, as opposed to the n−l+1 children in the Motif Finding problem.

SIMPLEMEDIANSEARCH(*DNA, t, n, l*)

s $(1, 1, \ldots, 1)$

bestDistance

```
i        1
while i > 0
if i < l
(s, i)   NEXTVERTEX(s, i, l, 4)
else
word        nucleotide string corresponding to (s1, s2, . . . sl)
if TOTALDISTANCE(word,DNA) < bestDistance
bestDistance   TOTALDISTANCE(word,DNA)
bestWord        word
(s, i)        NEXTVERTEX(s, i, l, 4)
return bestWord
```

BRANCHANDBOUNDMEDIANSEARCH(DNA, t, n, l)

```
s       (1, 1, . . . , 1)
bestDistance
i        1
while i > 0
if i < l
prefix        nucleotide string corresponding to (s1,
                s2, . . . , si)
optimisticDistance       TOTALDISTANCE(prefix,DNA)
if optimisticDistance > bestDistance
(s, i)   BYPASS(s, i, l, 4)
else
(s, i)        NEXTVERTEX(s, i, l, 4)
else
word        nucleotide string corresponding to
                (s1, s2, . . . sl)
if TOTALDISTANCE(word,DNA) < bestDistance
bestDistance       TOTALDISTANCE(word,DNA)
bestWord          word
(s, i)        NEXTVERTEX(s, i, l, 4)
return bestWord
```

Topic # 56 Genomic Rearrangements

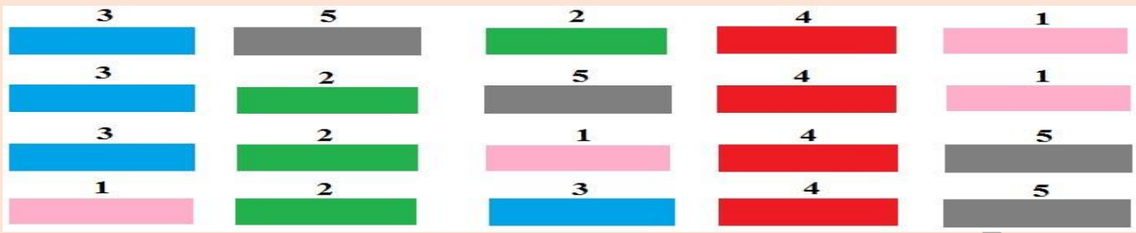
Waardenburg's syndrome

- Hearing loss
- Two Different colored eyes
- Gene present on chromosome 2

MUHAMMAD IMRAN

- Splotch gene in mice
- Human genome-mouse genome
- Cut into 300 genomic fragments-Synteny blocks
- Chromosome 2 in humans-mouse chromosomes 1, 2, 3, 5, 6, 7, 10, 11, 12, 14, and 17
- Genome rearrangement results in a change of gene ordering
- Analysis of human and mouse genomes-250 genomic rearrangements

Mouse X chromosome



Human X chromosome

Transformation of the mouse gene order into the human gene order on the X chromosome

Topic # 57 Greedy Approach for Motif Search

Approximation algorithm

Brute force algorithm to solve the Motif Finding

problem running time of $O(l \cdot n^t)$

Cannot run it on

biological samples

Faster greedy

technique-not correct,

good performance

Approximation

algorithm

CONSENSUS- as good

or better

GREEDYMOTIFSEARCH scans each DNA sequence only once. Once we have scanned a particular sequence *i*, we decide which of its *l*-mer has the

best contribution to the partial alignment score Score(s, *i*, *DNA*) for the first *i* sequences and immediately claim that this *l*-mer is part of the alignment.

GREEDYMOTIFSEARCH (*DNA, t, n, l*)

1. bestMotif (1, 1, . . . , 1)
2. s (1, 1, . . . , 1)
3 for s_1 1 to $n - l + 1$
4 for s_2 1 to $n - l + 1$
5. if Score(s, 2,DNA) > *Score(bestMotif* , 2,DNA)

6. *BestMotif1* s_1

7. *BestMotif2* s_2

8. s_1 $BestMotif_1$

9. s_2 $BestMotif_2$

10. for *i* 3 to t

11 for s_i 1 to $n - l + 1$

12. if Score(s, *i*, *DNA*) > Score (*bestMotif* , *i*, *DNA*)

13. $BestMotif_i$ s_i

14. s_i *bestMotifi*

15. return *bestMotif*

Approximation algorithm

Two

closest l-

mers $2 \times l$

seed

matrix $l(n$

$- l + 1)^2$

operations

t − 2 iterations-by scanning the *i*th sequence (for $3 \leq i \leq t$)

t−2 sequences and selecting the one *l*-mer that has

themaximumScore(*s*, *i*) $l \cdot (n - l + 1)$ operations

Approximation algorithm

Running time of this algorithm is $O(ln^2 + lnt)$, which is vastly better than the $O(lnt)$ of
IMPLEMOTIFSEARCH or even the $O(4lnt)$ of BRUTEFORCEMEDIAN-

STRING

Topic # 58 The Power of DNA Sequence Comparison

Introduction

Discovery of new gene-no idea of functions

Find similarities with genes of known function

Newly discovered cancer-causing -sis oncogene matched a normal gene involved in growth and development called platelet-derived growth factor

Oncogene *v*-sis is the simian sarcoma virus

Scientists became suspicious that cancer might be caused by a normal growth gene

Discovery of cystic fibrosis gene

MUHAMMAD IMRAN

Abnormal secretions, and is diagnosed in children produce abnormally thick mucus that clogs the lungs

10 million Americans are carriers of the cystic fibrosis gene

There is a 25% chance that the child will have cystic fibrosis

In 1989, 1 million nucleotides on the chromosome 7

The area around the cystic fibrosis gene was sequenced

Database of all known genes

Similarities between a gene that had already been discovered, code for *adenosine triphosphate (ATP) binding proteins*

These proteins constitute the ion transport channel

The disease involves sweat secretions with abnormally high sodium content

Link between cancer-causing genes and normal growth genes and elucidating the nature of cystic fibrosis

Topic # 59 Brute Force vs Greedy Algorithm

Dynamic Programming

Manhattan Tourist Problem:

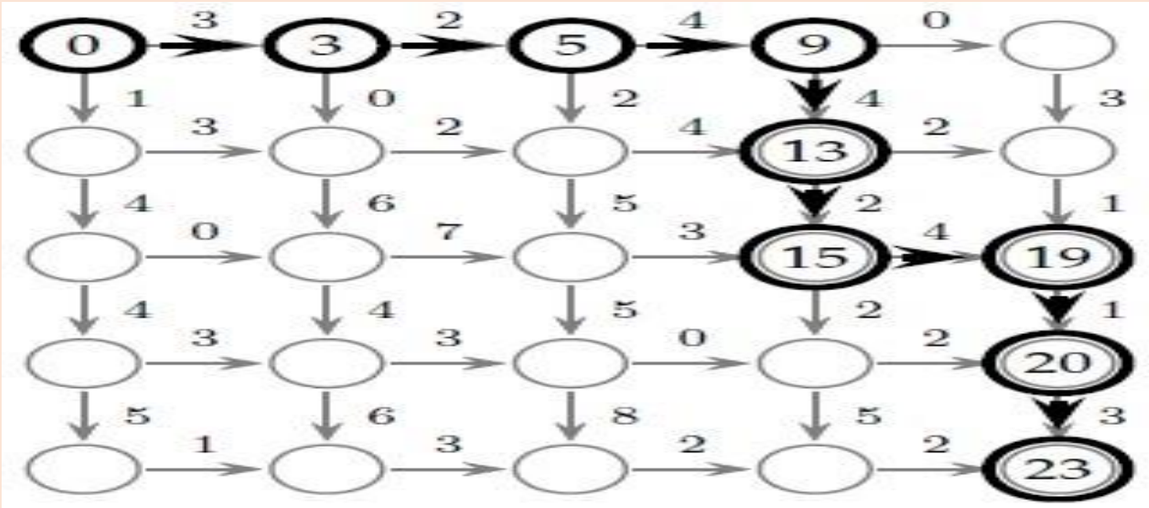
Find a longest path in a weighted grid **Input**: A weighted grid *G* with two distinguished vertices: a *source* and a *sink*

Output: A longest path in *G* from *source* to *sink*

Tourists only move south and east, any grid positions west or north of the source are unusable

Any grid positions south or east of the sink are unusable, the source vertex is at (0, 0) Sink vertex at (*n*, *m*) defines the southeastern most corner of the grid

MUHAMMAD IMRAN



The brute force approach search for the longest path

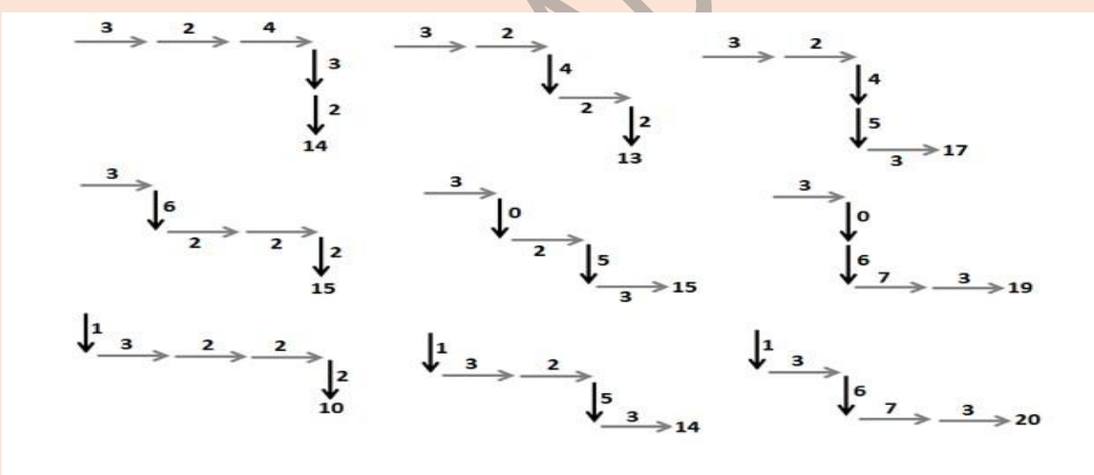
Not an option for medium to large grid

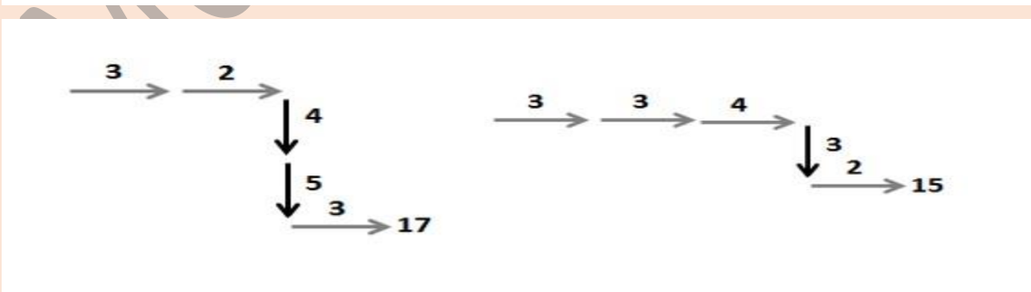
Use a greedy strategy, choose between two possible directions (south or east) by comparing them and selecting one with large increase for one step (local maximum)

Good at beginning

May lead to area with few attractions

No known greedy strategy for the Manhattan Tourist problem provides an optimal solution to the problem





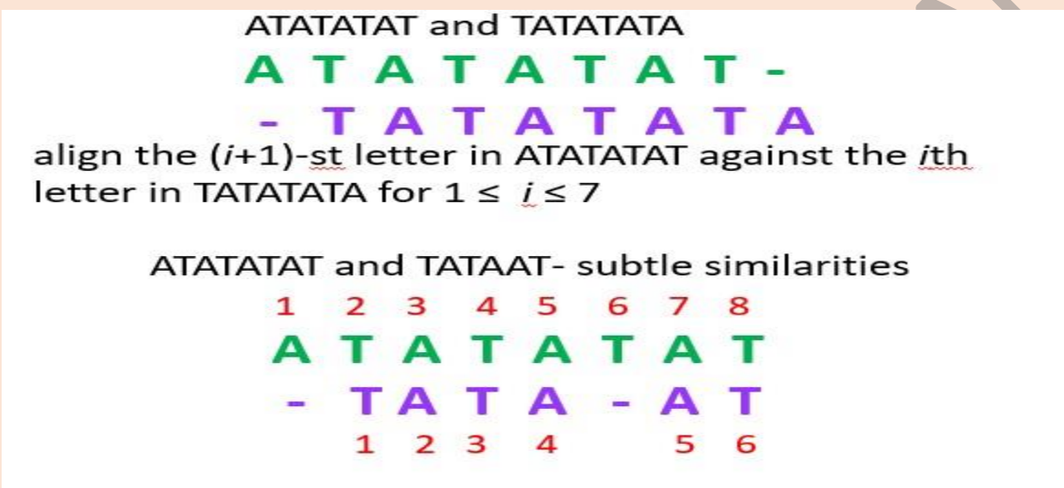
Instead of solving the Manhattan Tourist problem directly, that is, finding the longest path from source (0, 0) to sink (n,m), we solve a more general problem: find the longest path from source to an arbitrary vertex (i, j) with

$0 \le i \le n$, $0 \le j \le m$. We will denote the length of such a best path as $s_{i,j}$, noticing that $s_{n,m}$ is the weight of the path that represents the solution to the Manhattan Tourist problem

Topic #60 Sequence Similarity

Meaning of "sequence similarity or "distance" between DNA sequences. Hamming distance is not typically used to compare DNA or protein sequences. Calculation rigidly assumes that the *i*th symbol of one sequence is already aligned against the *i*th symbol of the other

It is common case that the *i*th symbol in one sequence corresponds to a symbol at different position in other. Mutation in DNA-evolutionary process: DNA replication- substitutions, insertions, and deletions of nucleotides, leads to "edited" DNA texts. Whether the *i*th symbol in one DNA sequence corresponds to the *i*th symbol in the other



Topic # 61 Edit Distance

Edit distance between two strings as the minimum number of editing operations needed to transform one string into another, where the edit operations are insertion, deletion, and substitution of one symbol for another

It is often the case that the *i*th symbol in one sequence corresponds to a symbol at different position in other. Mutation in DNA-evolutionary process: DNA replication- substitutions, insertions, and deletions of nucleotides, leads to "edited" DNA texts. Whether the *i*th symbol in one DNA sequence corresponds to the *i*th symbol in the other

A T A T A T A T –
- T A T A T A T A

align the (i+1)-st letter in ATATATAT against the ith letter in TATATATA for $1 \leq i \leq 7$

1 2 3 4 5 6 7 8
A T A T A T A T
- T A T A - A T
1 2 3 4 5 6

TGCATAT can be transformed into ATCCGAT with five editing operations

TGCATAT
↓ delete last T
TGCATA
↓ delete last A
TGCAT
↓ insert A at front
ATGCAT
↓ substitute C for G at position 3
ATCCAT
↓ insert a G before the last A
ATCCGAT

Edit distance between TGCATAT and ATCCGAT is at most 5

TGCATAT can also be transformed into ATCCGAT with four editing operations

TGCATAT
↓ insert A at front
ATGCATAT
↓ delete T at position 6
ATGCAAT
↓ substitute G for A at position 5
ATGCGAT
↓ substitute C for G at position 3
ATCCGAT

Edit distance between TGCATAT and ATCCGAT is 4

Topic # 62 Alignment

The *alignment* of the strings v (of *n* characters) and w (of *m* characters, with *m* not necessarily the same as *n*) is a two-row matrix such that the first row contains the characters of v in order while the second row contains the characters of w in order, where spaces may be interspersed throughout the strings in different places

As a result, the characters in each string appear in order, though not necessarily adjacently.

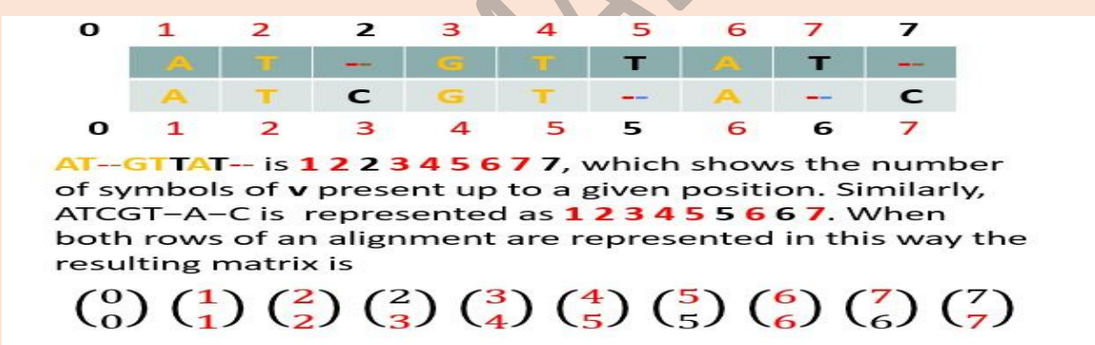
No column of the alignment matrix contains spaces in both rows, so that the alignment may have at most n + m columns.

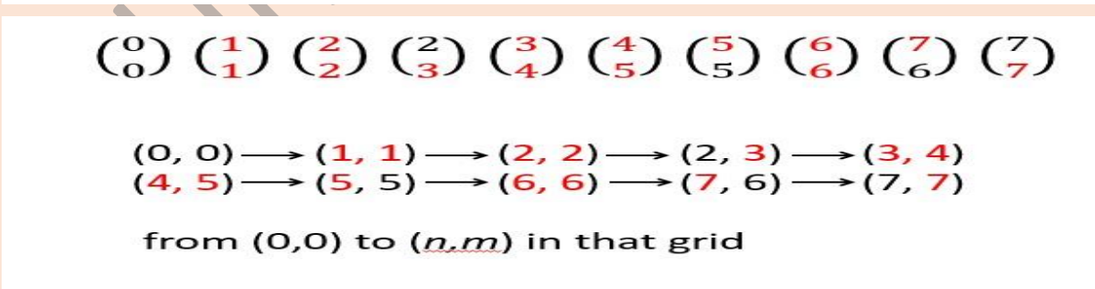
A	T	--	G	T	T	A	T	--
A	T	C	G	T	--	A	--	C

Columns that contain the same letter in both rows are called *matches*, while columns containing different letters are called *mismatches*. The columns of the alignment containing one space are called *indels*, with the columns containing a space in the top row called *insertions* and the columns with a space in the bottom row *deletions*. Five matches, zero mismatches, and four indels. The number of matches plus the number of mismatches plus the number of indels is equal to the length of the alignment matrix and must be smaller than $n + m$

A	T	--	G	T	T	A	T	--
A	T	C	G	T	--	A	--	C

Each of the two rows in the alignment matrix is represented as a string interspersed by space symbols "–"; for example AT--GTTAT-- is a representation of the row corresponding to v = ATGTTAT, while ATCGT--A--C is a representation of the row corresponding to w = ATCGTAC



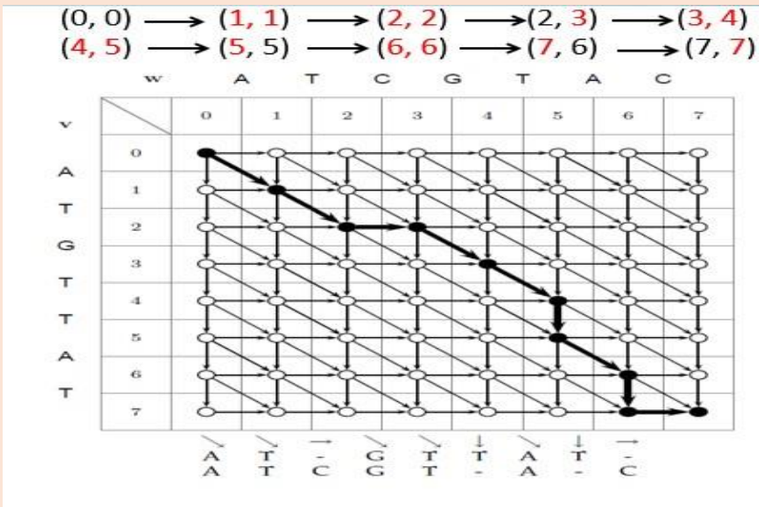


Topic # 63 Edit Graph 1

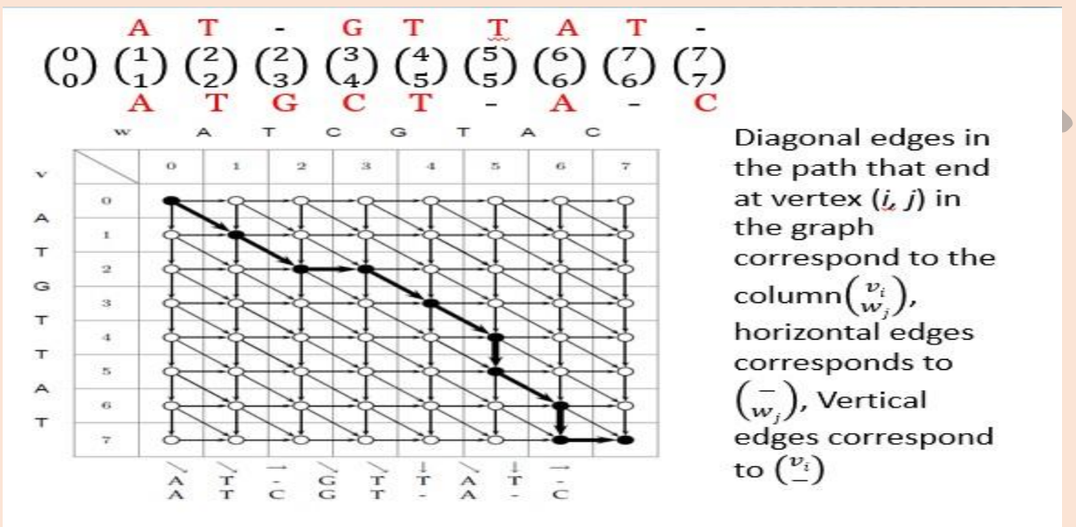
The grid that is achieved after alignment is similar to the Manhattan grid where each entry in the grid looks like a city block

The graph called Edit Graph

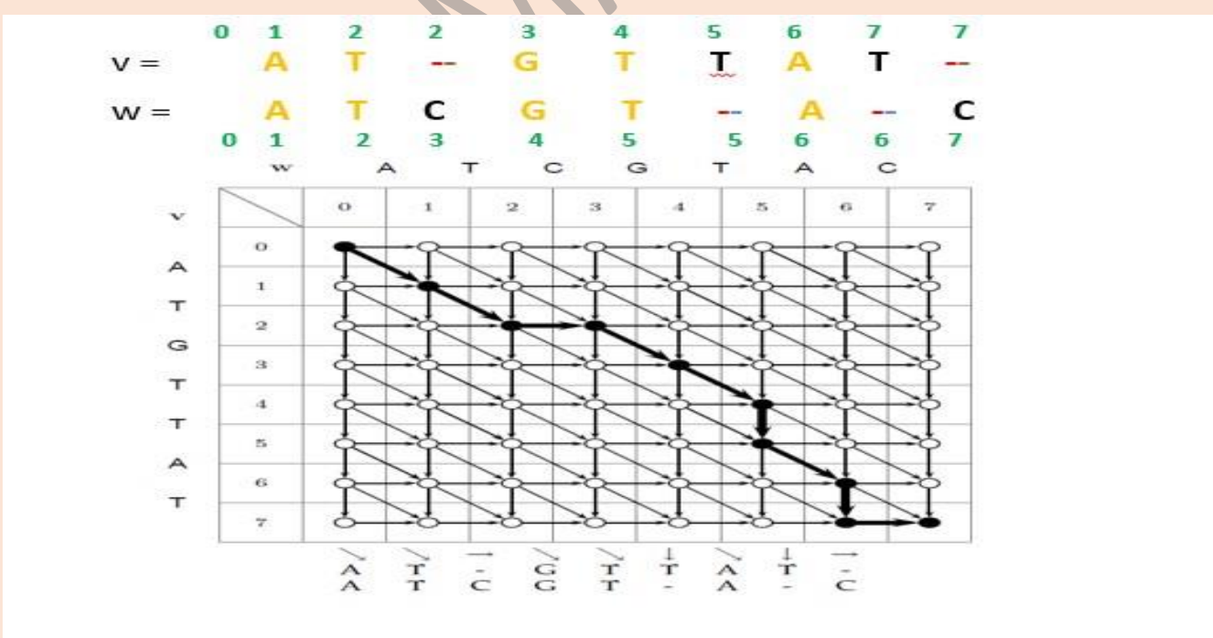
The main difference between the Manhattan and Edit Graph is that we can move along the diagonals in the Edit Graph



Edit Graph by introducing a vertex for every intersection of streets in the grid.
This graph will help in calculating edit distance
Alignment-path
Path-Alignment
Edge-one column



Diagonal edges in the path that end at vertex (i, j) in the graph correspond to the column $\binom{v_i}{w_j}$, horizontal edges corresponds to $\binom{-}{w_j}$, Vertical edges correspond to $\binom{v_i}{-}$



Topic # 64 Edit Graph 2

Analyzing the merit of an alignment-corresponding path in the edit graph. For any two strings- different alignment matrices and corresponding paths.

Surplus of mismatches and indels and a small number of matches, while others have many matches and few indels and mismatches.

Relative merits of one alignment over another-scoring function- input an alignment matrix and produces a score that determines the "goodness" of the alignment.

Variety of scoring functions-higher scores to alignments with more matches.

Column as a positive number if both letters are the same, and as a negative number if the two letters are different. The score for the whole alignment is the sum of the individual column scores**.**

Topic # 65 Longest Common Subsequences

The simplest form of a sequence similarity analysis is the Longest Common Subsequence (LCS) problem, where we eliminate the operation of substitution and allow only insertions and deletions. A subsequence of a string v is simply an (ordered) sequence of characters (not necessarily consecutive) from v.

v = ATTGCTA AGCA and ATTA are subsequences

TGTT and TCG are not subsequences

A common subsequence of two strings is a subsequence of both of them. Common subsequence of strings

v = $v_1 \ldots v_n$ and w = $w_1 \ldots w_m$ as a sequence of positions in v, $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ and a sequence of positions in w, $1 \leq j_1 < j_2 < \cdots < j_k \leq m$ such that the symbols at the corresponding positions in v and w coincide:

$v_{it} = w_{jt}$ for $1 \leq t \leq k$

TCTA is a common to both ATCTGAT and TGCATA

Typically many common subsequences between two strings v and w-how to find the longest one

s(v,w)-be the length of the longest common subsequence of v and w

edit distance between v and w—under the assumption that only insertions and deletions are allowed—is $d(v,w) = n + m - 2s(v,w)$ and corresponds to the minimum number of insertions and deletions needed to transform v into w

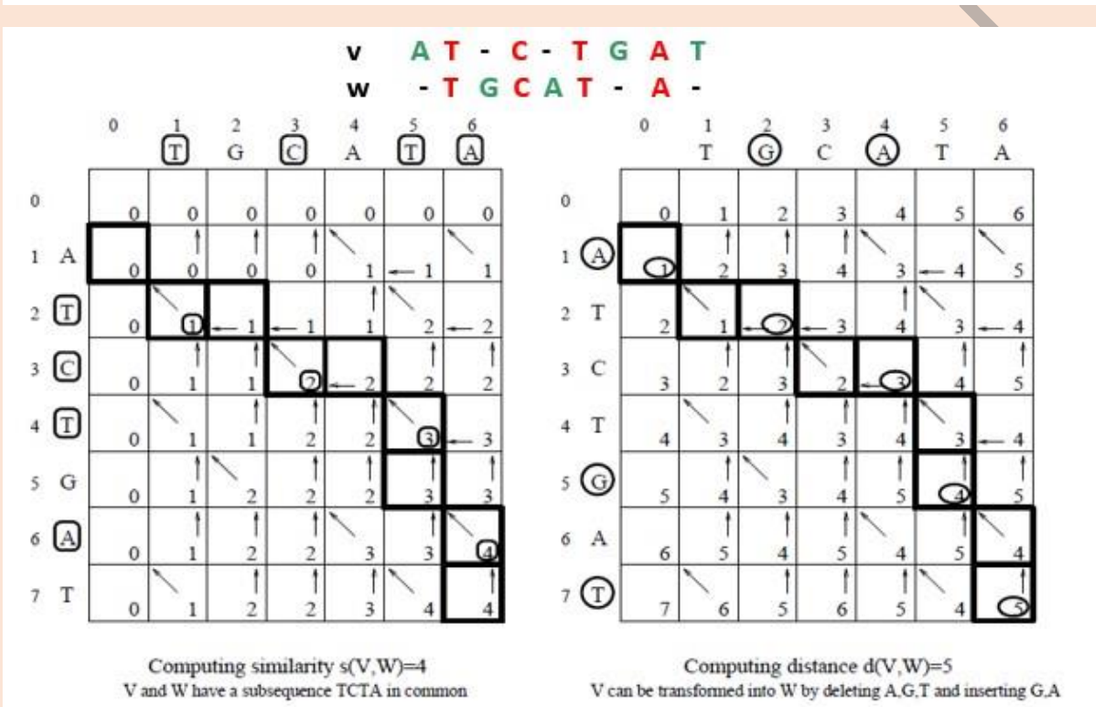
MUHAMMAD IMRAN

Alignment

v — A T - C - T G A T

w — - T G C A T - A -

Above figure presents an LCS of length 4 for the strings v = ATCTGAT and w = TGCATA and a shortest sequence of two insertions and three deletions transforming **v** into **w**



Longest Common Subsequence Problem:

Find the longest subsequence common to two strings

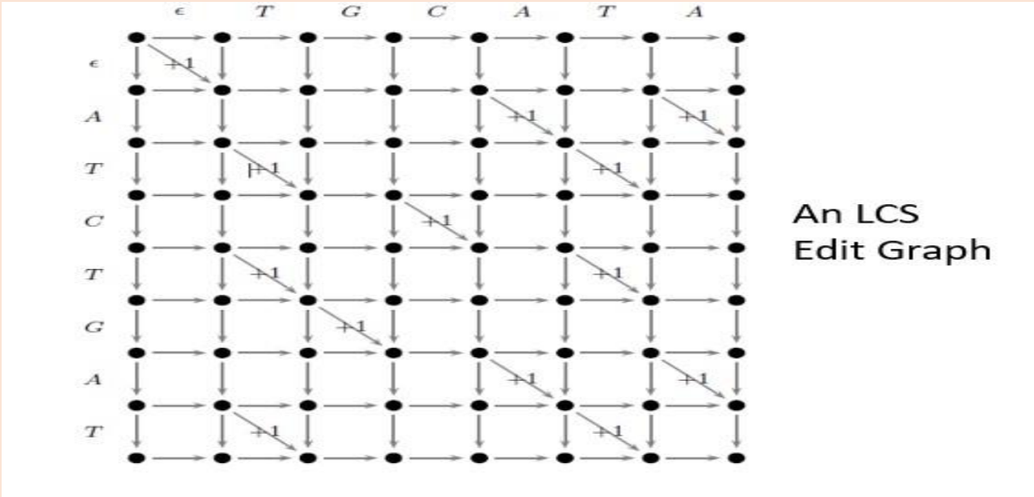
Input: Two strings, v and w

Output: The longest common subsequence of v and w

Topic # 66 Recurrence for LCS problem

Every common subsequence corresponds to an alignment with no mismatches.

This can be obtained simply by removing all diagonal edges from the edit graph whose characters do not match, thus transforming it into a graph like that shown in the figure



The relationship between

the Manhattan Tourist problem and the LCS Problem is further illustrate by showing that these two problems lead to very similar recurrences

Define $s_{i,j}$ to be the length of an LCS between $v_1 \ldots v_i$, the i-prefix of v and $w_1 \ldots w_j$, the $j$prefix of w. Clearly, $s_{i,0} = s_{0,j} = 0$ for all $1 \le i \le n$ and $1 \le j \le m$ $s_{i,j}$ satisfies the following recurrence

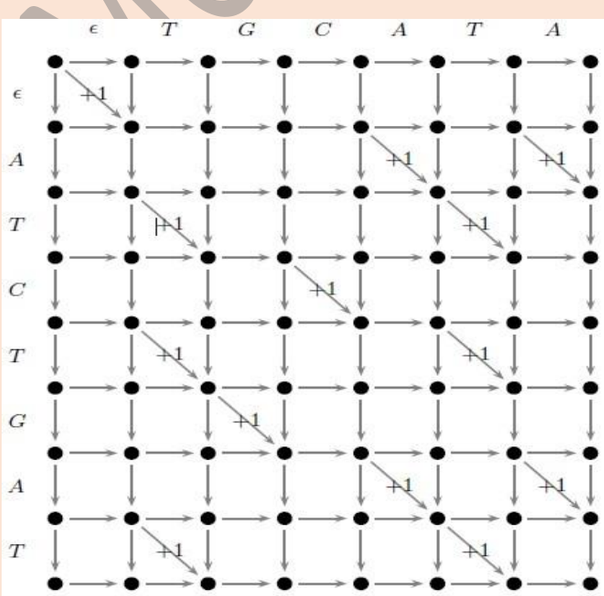
$$s_{i,j} = \max \begin{cases} s_{i-1,j} \\ s_{i,j-1} \\ s_{i-1,j-1} + 1, \quad \text{if } v_i = w_j \end{cases}$$

The first term- when v_i is not present in the LCS

of the i-prefix of v and j-prefix of w (deletion of v_i); the second term- when w_j is not present in this LCS (an insertion of w_i); and the third term-when both v_i and w_j are present in the LCS (v_i matches w_j).

These recurrences can be rewritten by adding some zeros here and there as

$$s_{i,j} = \max \begin{cases} s_{i-1,j} + 0 \\ s_{i,j-1} + 0 \\ s_{i-1,j-1} + 1, \quad \text{if } v_i = w_j \end{cases}$$



‘s’ is used to represent dynamic programming table, the data structure

The length of an LCS between v and w can be read from the element (n,m) of the dynamic programming table, but to reconstruct the LCS from the dynamic programming table, one must keep some additional information about which of the three quantities, $s_{i-1,j}$, $s_{i,j-1}$, or $s_{i-1,j-1} + 1$, corresponds to the maximum in the recurrence for $s_{i,j}$.

Topic #67 Algorithms for LCS

The length of an LCS

Some additional information about which of the three quantities , $s_{i-1,j}$, $s_{i,j-1}$, or $s_{i-1,j-1} + 1$

The following algorithm achieves this goal by introducing backtracking pointers that take one of the three values , or .

Algorithms for LCS

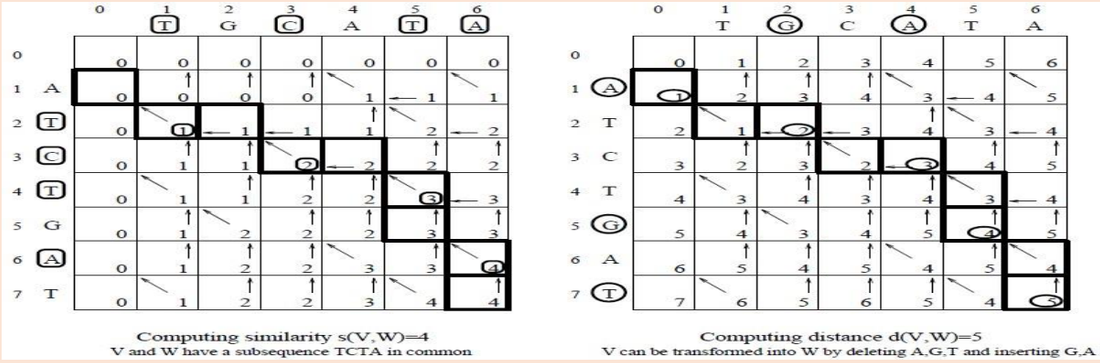
LCS(**v**,**w**)

1. **for** $i \leftarrow 0$ **to** n
2. $s_{i,0} \leftarrow 0$
3. **for** j $\leftarrow 1$ **to** m
4. $s_{0,j} \leftarrow 0$
5. **for** $i \leftarrow 1$ to n
6. **for** $j \leftarrow 1$ to m
7. $s_{i,j} \leftarrow \max \begin{cases} s_{i-1,j} \\ s_{i,j-1} \\ s_{i-1,j-1} + 1, & \text{if } v_i = w_j \end{cases}$
8. $b_{i,j} \leftarrow \begin{cases} \uparrow & \text{if } s_{i,j} = s_{i-1,j} \\ \leftarrow & \text{if } s_{i,j} = s_{i,j-1} \\ \nwarrow & \text{if } s_{i,j} = s_{i-1,j-1} + 1 \end{cases}$
9. **return** $(s_{n,m}, \mathbf{b})$

Algorithms for LCS

PRINTLCS(**v**,**w**, i,j)

1. **for** $i = 0$ **or** $j =$
2. **return**
3. $b_{i,j} = \nwarrow$
4. PRINTLCS (**b**, v, i-1, j-1)
5. **print** v_i
6. **Else**
7. **if** $b_{i,j} = \uparrow$
8. PRINTLCS (**b**, **v**, i-1, j)
9. **Else**
10. PRINTLCS (**b**,**v** i, j-1)



Dynamic programming table-computation of the similarity score *s*(v,w) between v and w, while the table on the right-computation of the edit distance between v and w-insertions and deletions are the only allowed operations. The edit distance d(v,w) is computed according to the initial conditions $d_{i,0} = i$, $d_{0,j} = j$ for all $1 \le i \le n$ and 1 j m and the following recurrence:

The edit distance d(v,w) is computed according to the initial conditions $d_{i,0} = i$, $d_{0,j} = j$ for all $1 \le i \le n$ and 1 j m and the following recurrence:

$$d_{i,j} = \min \begin{cases} d_{i-1,j+1} \\ d_{i,j-1} + 1 \\ d_{i-1,j-1}, & \text{if } v_i = w_j \end{cases}$$

Topic #68 Scoring Alignments 1

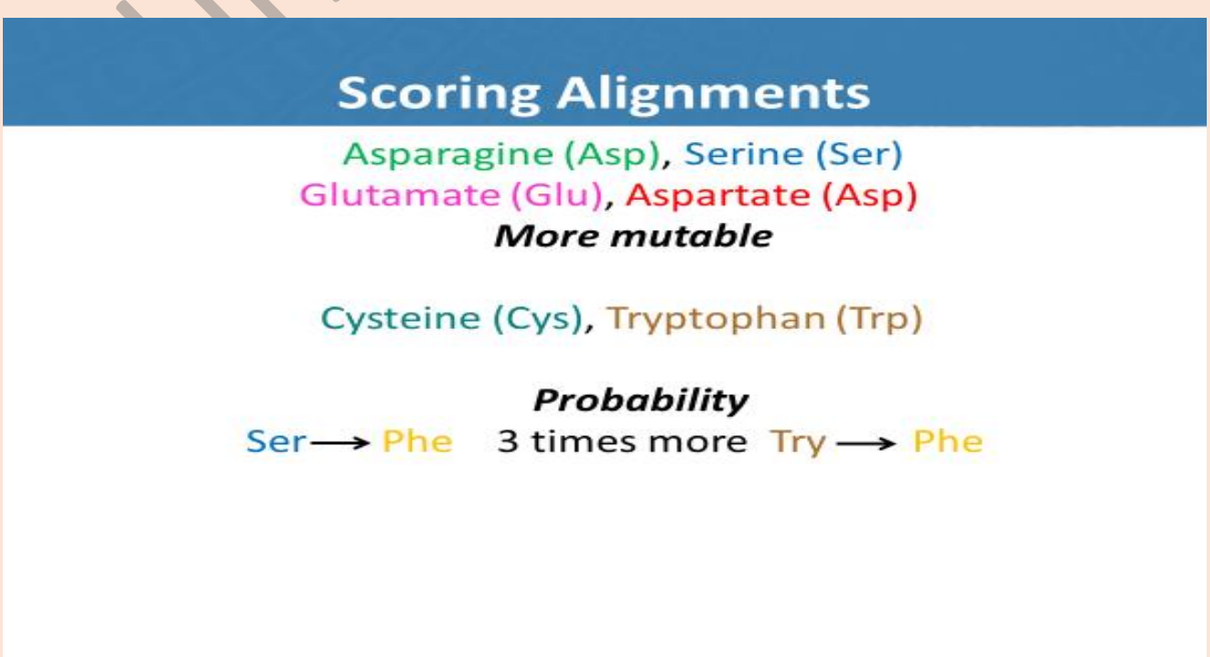
Scoring matrices for DNA sequence- μ and σ

Scoring matrices for protein sequences are complicated – *pointed accepted mutation* (PAM) and *block substitution* (BLOSUM), frequency amino acid '*x*' replaces amino acid '*y*' in evolutionary related sequences.

Random mutagenesis-change amino acid sequence

Some mutations- do not alter but other do

Some amino acid substitutions are commonly found through the process of molecular evolution-



MUHAMMAD IMRAN

Types of changes- most and least common- amino acid scoring matrix, sequence alignment

Amino acids sequences- very few matches-scoring matrix $\delta(i, j)$ – how often a. a '*i*' substitutes a. a '*j*'

Topic # 69 Scoring Alignments 2

Large set of alignments of related sequences

Computing $\delta(i,j)$

Amounts to counting how many times the amino acid '*i*' is aligned with amino acid '*j*'

Needs to know scoring matrix

Met Ala Phe Ser Gly Asp Glu Ser. . . . . .

Met Ala Phe Ser -- Asp Glu Ser. . . . . .

If proteins are 90% identical, premium +1 for matches and -1 for mismatches and indels will do the job. Then "obvious" alignments are constructed that are used to compute scoring matrix δ.

The simplified description hides subtle details are important in the construction of scoring matrix

Ser Phe Try Phe

(related proteins in mouse and rat) *LESS* (related proteins in mouse and human)

15 million years 80 million years

The best scoring matrix to compare two proteins depends on similarity of these organisms

Topic #70 PAM matrix

Problem is rectified- analyzing similar proteins e.g, one mutation/100 a.a. Proteins- human and chimpanzee, Such sequences, *one PAM unit diverged.* PAM unit as the amount of time in which an "average" protein mutates 1% of it's a.a. *PAM1* scoring matrix- many alignments of extremely similar **proteins** For a given set of base alignments

define $f(i,j)$ as the total number of times amino acids *i* and *j* are aligned against each other, divided by the total number of aligned positions.

Also define $g(i,j)$ as $\frac{f(i,j)}{f(i)}$ where $f(i)$ is the frequency of the amino acid *i* in all proteins from data set.

$g(i,j)$ defines the probability that an amino acid *i* mutates into amino acids *j* within 1 PAM unit. The (i,j) entry of the PAM 1 matrix is defined as $\delta(i,j)$ =

$\log \frac{f(i,j)}{f(i).f(j)} = \log \frac{g(i,j)}{f(j)}$ ($f(i)$. $f(j)$, the frequency of aligning amino acid *i* against amino acid *j* that one expects simply by chance). The *PAM n* matrix can be defined as the result of applying the PAM 1 matrix n times

MUHAMMAD IMRAN

If g is the 20 x 20 matrix of frequencies $g(i,j)$, then g^n (multiplying this matrix by itself n times) gives the probability that amino acid i mutates into amino acid j during n PAM units. The (i,j) entry of the PAM n matrix is defined as $\log \frac{gni,j}{f(j)}$

For large n, the resulting PAM matrices often allow one to find related proteins even when the practically no matches in the alignment. In this case the underlying nucleotide sequences are so diverged that their comparison usually fails to find any statistically significant similarities.

Similarity between the

Cancer –causing *v*-sis oncogene

Growth factor PDGF (Platelet derived growth factor)

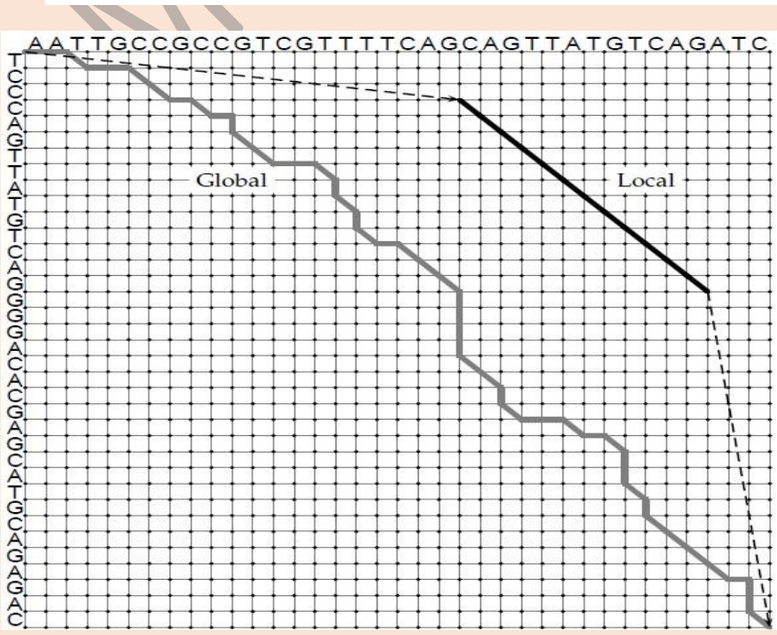
Remained undetected

Russell Doolittle and colleagues has not transformed the nucleotide sequences into amino acids sequences prior to performing the comparison.

Topic #71 Local Sequence Alignment

Global Alignment- similarity 2 strings- extends over entire length, eg, same protein family- very conserved, same length – fruit flies to humans Score of alignment- 2 substrings 'v', 'w'- larger than score of alignment between entire length of 'v' and 'w'.

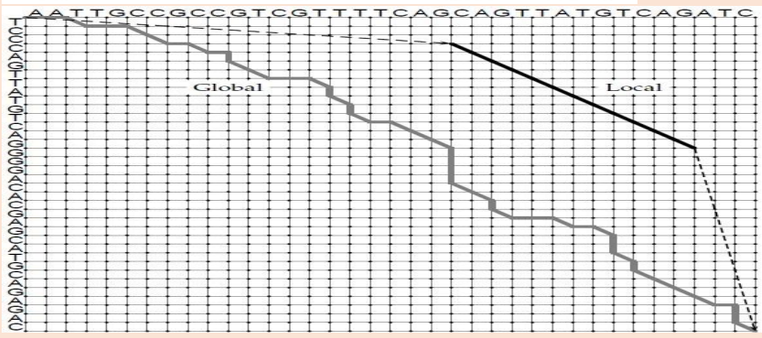
Homeobox genes, regulate embryonic development- variety of species. Different in different species- one region- *homeodomain*- highly conserved. How to find conserved area- ignore areas-little similarity. Temple Smith and Micheal Waterman proposed modification of the global sequence alignment dynamic programming algorithm-LAP



The conserved domain are identical and cover one third of the entire length, *n*, of these genes. Path from *source* to *sink* will include approx $\frac{2}{3}n$ horizontal edges, $\frac{1}{3}n$ diagonal match edges (corresponding to homeodomains) and $\frac{2}{3}n$ vertical edges

The score of this path is

$$-\frac{2}{3}n\sigma + \frac{1}{3}n - \frac{2}{3}n\sigma = n\left(\frac{1}{3} - \frac{4}{3}\sigma\right)$$



The score of this path is

$$-\frac{2}{3}n\sigma + \frac{1}{3}n - \frac{2}{3}n\sigma = n\left(\frac{1}{3} - \frac{4}{3}\sigma\right)$$

This path contains so many indels that it is unlikely to be the highest scoring alignment.
Biologically irrelevant diagonal paths – likely have score- mismatches are

penalized less than indels. Score of diagonal path is $n(\frac{1}{4} - \frac{3}{4}\mu)$, since every diagonal edge $\frac{1}{4}$ and mismatch with probability $\frac{3}{4}$. . Since $(\frac{1}{3} - \frac{4}{3}\sigma) < (\frac{1}{4} - \frac{3}{4}\mu)$ for indels and mismatch penalties, global alignment- miss correct solution of real biological problem, biologically irrelevant near-diagonal path.

Topic # 72

same TOPIC 71

Topic #73 Local Alignment Problem

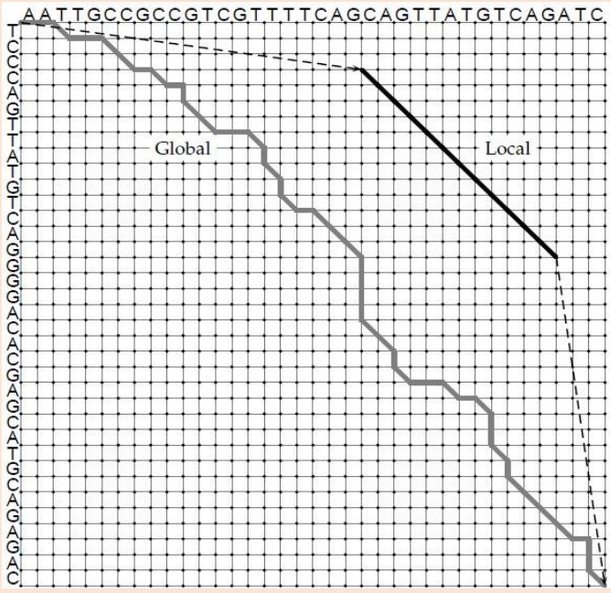
Biological significant in certain parts of DNA fragments, maximize the alignment score $s(v_i \ldots v_{i'}, w_j \ldots w_{j'})$ over all substrings $v_i \ldots v_{i'}$ of v and $w_j \ldots w_{j'}$ of w. Local Alignment Problem- not extend over entire length as in Global Alignment Problem

Local Alignment Problem:

Find the best local alignment between two strings

Input: Strings v and w and a scoring matrix δ

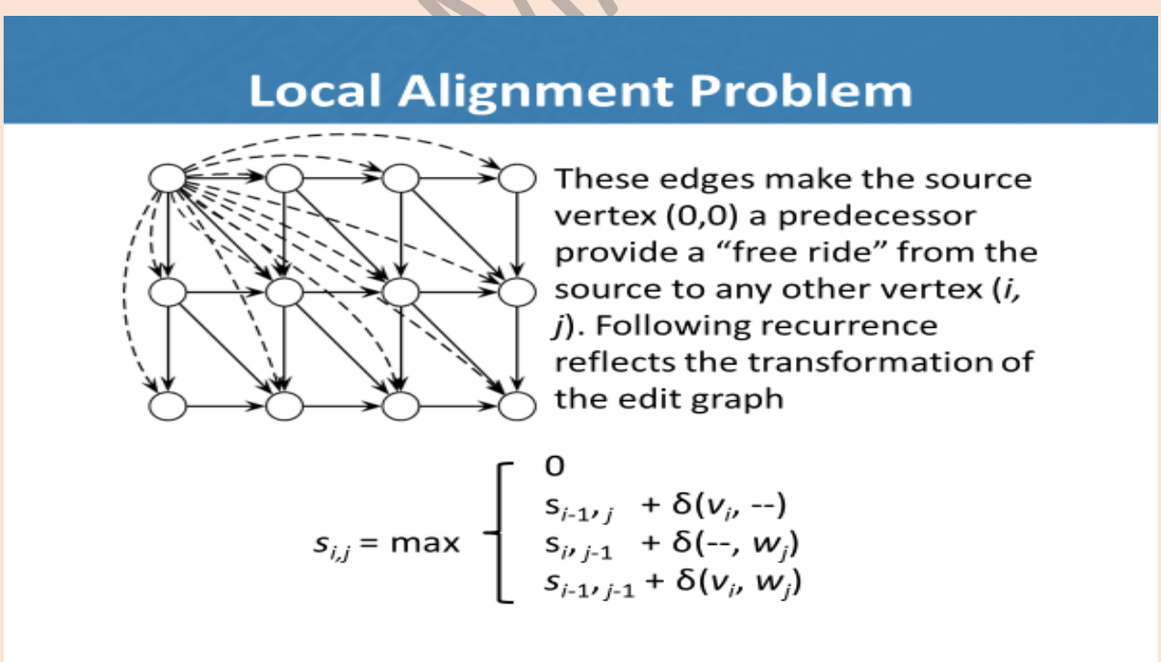
Output: Substring of v and w whose global alignment, as defined by δ, is maximal amoung all global alignment of all substrings of v and w.



Global Sequence Alignment- finding the longest path between vertices (0,0) and (*n*,*m*) in the edit graph

Local Alignment-finding the longest path among the paths between *arbitary vertices* (*i*, *j*) and (*i′*, *j′*) in the edit graph.

Find the longest path between every pair of vertices (*i*, *j*) and (*i′*, *j′*)- then select longest of these computed paths. Instead of finding the longest path from (*i*, *j*) to (*i′*, *j′*), LAP- finding the longest path from the *source* (0,0) to every other vertex by adding edges to weight 0



Topic #74 Progressive Multiple Alignment

Another approach-strong pairwise alignment Greedy *progressive multiple alignment* heuristic-pair of strings with greatest similarity-new string- "once a gap, always a gap." Multiple alignment of *k* sequences is reduced to the multiple alignment of *k*−1 sequences.

The motivation for the choice of the closest strings at the early steps of the algorithm is that close strings often provide the most reliable information about a real alignment

Many popular iterative multiple alignment algorithms including the tool CLUSTAL, use similar strategies

ATGTCATATTCGGAC

ATGTCATATTCGGAC

ATGTCATATTCGGAC

ATG- CATA

ATGTCATA

Progressive multiple alignment algorithms- problem with CLUSTAL-may be misled by some spuriously strong pairwise alignment effect, a bad seed. The error in initial pairwise alignment will propagate all the way through to the whole multiple alignment. Many algorithms have been proposed,-even with systematic deficiencies are quite useful in computational biology

Multiple alignment for *k* sequences

Generalization of the Pairwise Alignment problem Existence of a k-dimensional scoring matrix *k*-dimensional scoring matrices are not very practical Describe two other scoring approaches that are

more biologically relevant. The choice of the scoring function can drastically affect the quality of the resulting alignment, and no single scoring approach is perfect in all circumstances.

Multiple alignment of *k* sequences-a path of edges in a *k*-dimensional-Manhattan gridlike edit graph.

The weights of the edges-scoring function

Intuitively, assign higher scores to the columns with a low variation in letters-high scoreshighly conserved sequences

Multiple Longest Common Subsequence problem, the score of a column is set to 1 if all the characters in the column are the same, and 0 if even one character disagrees

Topic # 75 Gene Prediction 1

Sydney Brenner and Francis Crick

Every triplet codes for one amino acid

Introduce deletions in DNA-dramatically alters its protein product. Deleting three consecutive nucleotides results in minor changes in the protein

The phrase

THE SLY FOX AND THE SHY DOG

(written in triplets)

Turns into nonsense after deleting one letter Y from SLY

THE SYF OXA NDT HES HYD OG

or two letters LY from SLY

MUHAMMAD IMRAN

THE SFO XAN DTH ESH YDO G

but makes some sense after deleting three nucleotides SLY

THE SOX AND THE SHY DOG

Charles Yanofsky proved that a gene and its protein product are collinear. Yanofsky's experiment was so influential that nobody even questioned about codons and for almost 2 decades biologists believed that a protein was encoded by a long string of adjacent triplets.

Discovery of split human genes-collection of substrings

Raised the computational problem-prediction of genes

Human genome is larger and complex than bacterial genome

Salamander genome is ten times larger than the human genome Large amounts of so-called junk DNA

Human genes - exons that are separated by this junk DNA. The difference in the sizes of the salamander and human genomes thus presumably reflects larger amounts of junk DNA and repeats in the salamander genome.

Split genes are analogous to a magazine article- page 1, 13, 43, 51, 74, 80, and 91, with pages of advertising appearing in between. Junk DNA represents "advertising" that separates exons.

Topic # 76 Gene Prediction 2

The jump is inconsistent from species to species.

A gene in insect genome- different in worm

Number of exons

The information in one part in human-broken up into two in the mouse.

While the genes themselves are related, they may be quite different in terms of the parts' structure

Split genes-1977

Phillip Sharp and Richard Roberts- Adenovirus

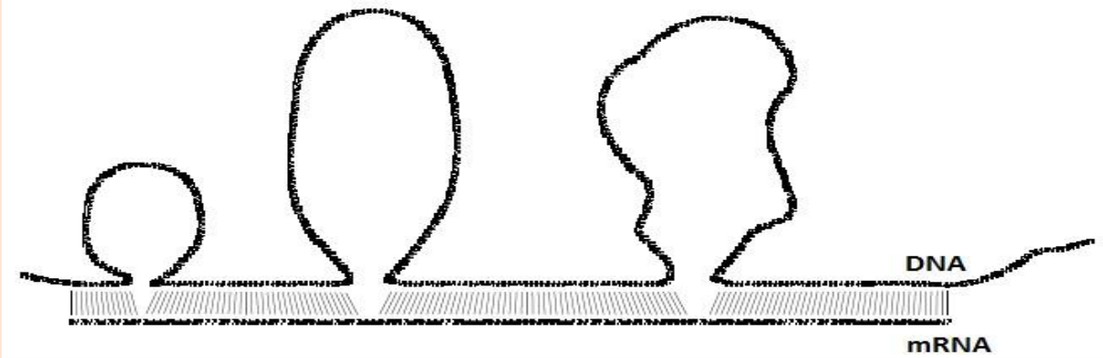
"An Amazing Sequence Arrangement at the 5' End of Adenovirus 2Messenger

RNA." Sharp's group- hexon.

Hexon mRNA, mRNA was hybridized to adenovirus DNA- the hybrid molecules-electron microscopy.

mRNA-DNA hybrids-three loop structures-continuous duplex segment- classic continuous gene model

MUHAMMAD IMRAN



An electron microscopy experiment led to the discovery of split genes

Hexon mRNA is built from four separate fragments of the adenovirus genome

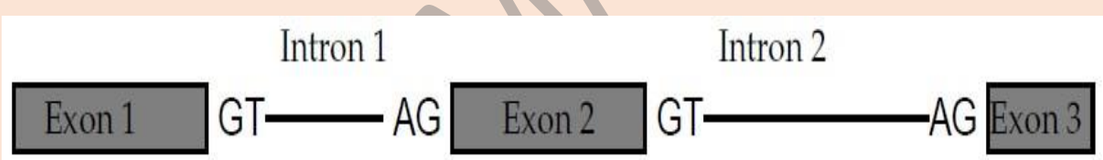
These four continuous segments, exons, in the adenovirus genome are separated by three "junk" fragments called introns**.**

77. One approach of Gene Prediction

Human genes-3% of the human genome- no in silico gene recognition algorithm The intronexon model-in eukaryotic organisms Prokaryotic organisms Gene prediction algorithms for prokaryotes tend to be somewhat simpler than those for eukaryotes

Two categories-predicting gene location. The statistical approach to gene prediction *Splicing signals*- exon-intron junctions

Dinucleotides AG and GT on the left- and right-hand sides of an exon are highly conserved



Less conserved positions on both sides of the exons

The simplest way to represent such binding sites is by a profile describing the propensities of different nucleotides to occur at different positions

Using profiles-detect splice sites

Profiles are quite weak-match frequently in the genome at non splice sites.

Attempts to improve the accuracy of gene prediction- second category-based on similarity.

Topic #78 Second Approach for Gene Prediction

The similarity-based approach to gene prediction relies on the observation that a newly sequenced gene has a good chance of being related to one that is already known. For example, 99% of mouse genes have human analogs.

Simply look for a similar sequence-based on the genes known in another-exon sequence and the exon structure are different

The commonality-produce similar proteins

Similarity-based methods attempt to solve a combinatorial puzzle-putative exons in a genomic sequence – mouse-human

MUHAMMAD IMRAN

Know a human protein, and we want to discover the exon structure of the related gene in the mouse genome. The more sequence data we collect, the more accurate and reliable similarity based

methods become. Consequently, the trend in gene prediction has recently shifted from statistically motivated approaches to similarity-based algorithms

Topic # 79 Statistical Approach to Gene Prediction 1

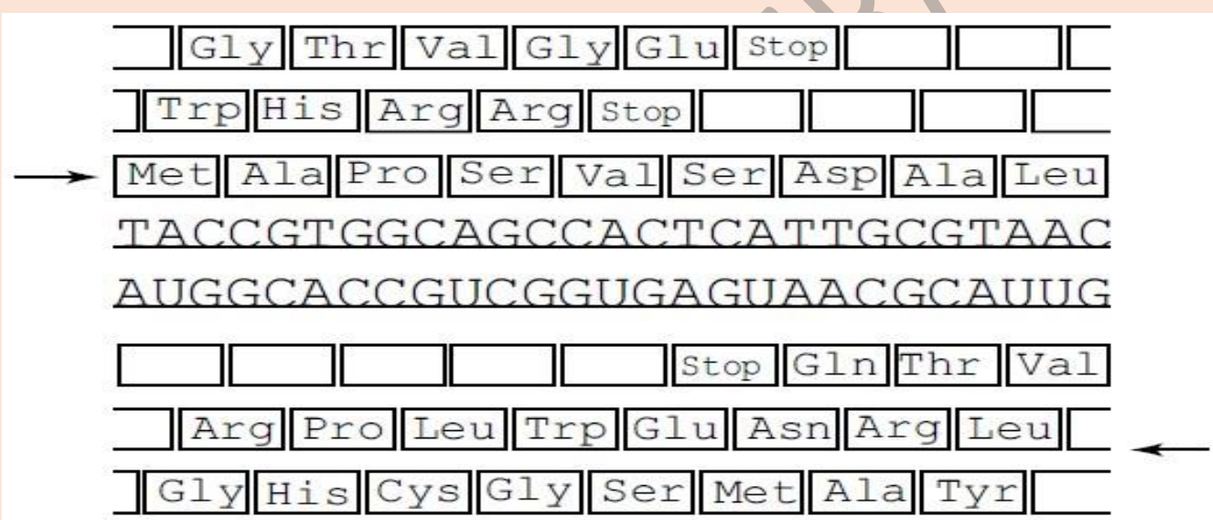
Statistical approaches to finding genes-statistical variations between coding (exons) and noncoding regions

Open reading frames (ORFs) Fenome of length *n* as a sequence of n/3 codons

The three "stop" codons, (**TAA**, **TAG**, and **TGA**)

The subsegments of these that start from a start codon, ATG, are ORFs.

ORFs within a single genomic sequence may overlap since there are six possible "reading frames": three on one strand starting at positions 1, 2, and 3, and three on the reverse strand



The six reading frames for the DNA sequence

One would expect to find frequent stop codons in noncoding DNA, since the average number of codons between two consecutive stop codons in "random"

DNA should be 64/3 ~ 21. This is much smaller than the number of codons in an average protein, which is roughly 300. Therefore, ORFs longer than some threshold length indicate potential genes. However, gene prediction algorithms based on selecting significantly long ORFs may fail to detect short genes or genes with short exons

Many statistical gene prediction algorithms rely on statistical features in protein-coding regions, such as biases in *codon usage*. We can enter the frequency of occurrence of each codon within a given sequence into a 64-element *codon usage array*

	U		C		A		G	
U	UUU Phe	57	UCU Ser	16	UAU Tyr	58	UGU Cys	45
	UUC Phe	43	UCC Ser	15	UAC Tyr	42	UGC Cys	55
	UUA Leu	13	UCA Ser	13	UAA Stp	62	UGA Stp	30
	UUG Leu	13	UCG Ser	15	UAG Stp	8	UGG Trp	100
C	CUU Leu	11	CCU Pro	17	CAU His	57	CGU Arg	37
	CUC Leu	10	CCC Pro	17	CAC His	43	CGC Arg	38
	CUA Leu	4	CCA Pro	20	CAA Gln	45	CGA Arg	7
	CUG Leu	49	CCG Pro	51	CAG Gln	66	CGG Arg	10
A	AUU Ile	50	ACU Thr	18	AAU Asn	46	AGU Ser	15
	AUC Ile	41	ACC Thr	42	AAC Asn	54	AGC Ser	26
	AUA Ile	9	ACA Thr	15	AAA Lys	75	AGA Arg	5
	AUG Met	100	ACG Thr	26	AAG Lys	25	AGG Arg	3
G	GUU Val	27	GCU Ala	17	GAU Asp	63	GGU Gly	34
	GUC Val	21	GCC Ala	27	GAC Asp	37	GGC Gly	39
	GUA Val	16	GCA Ala	22	GAA Glu	68	GGA Gly	12
	GUG Val	36	GCG Ala	34	GAG Glu	32	GGG Gly	15

Topic # 80 Statistical Approach to Gene Prediction 2

Some more facts about genetic code and codon usage in humans

	U		C		A		G	
U	UUU Phe	57	UCU Ser	16	UAU Tyr	58	UGU Cys	45
	UUC Phe	43	UCC Ser	15	UAC Tyr	42	UGC Cys	55
	UUA Leu	13	UCA Ser	13	UAA Stp	62	UGA Stp	30
	UUG Leu	13	UCG Ser	15	UAG Stp	8	UGG Trp	100
C	CUU Leu	11	CCU Pro	17	CAU His	57	CGU Arg	37
	CUC Leu	10	CCC Pro	17	CAC His	43	CGC Arg	38
	CUA Leu	4	CCA Pro	20	CAA Gln	45	CGA Arg	7
	CUG Leu	49	CCG Pro	51	CAG Gln	66	CGG Arg	10
A	AUU Ile	50	ACU Thr	18	AAU Asn	46	AGU Ser	15
	AUC Ile	41	ACC Thr	42	AAC Asn	54	AGC Ser	26
	AUA Ile	9	ACA Thr	15	AAA Lys	75	AGA Arg	5
	AUG Met	100	ACG Thr	26	AAG Lys	25	AGG Arg	3
G	GUU Val	27	GCU Ala	17	GAU Asp	63	GGU Gly	34
	GUC Val	21	GCC Ala	27	GAC Asp	37	GGC Gly	39
	GUA Val	16	GCA Ala	22	GAA Glu	68	GGA Gly	12
	GUG Val	36	GCG Ala	34	GAG Glu	32	GGG Gly	15

in-frame hexamer count Mark Borodovsky

Gene prediction in bacterial genomes-several conserved sequence motifs often found in the regions around the start of transcription

Such sequence motifs are more elusive in eukaryotes

The approaches-prokaryotes-eukaryoyes

Exons-130 nucleotides-reliable peaks in the likelihood ratio plot while analyzing ORFs-do not differ enough from random fluctuations to be detectable. Moreover,

codon usage and other statistical parameters probably nothing in common in splicing machinery recognizes exons

Topic # 81 Statistical Approach to Gene Prediction 3

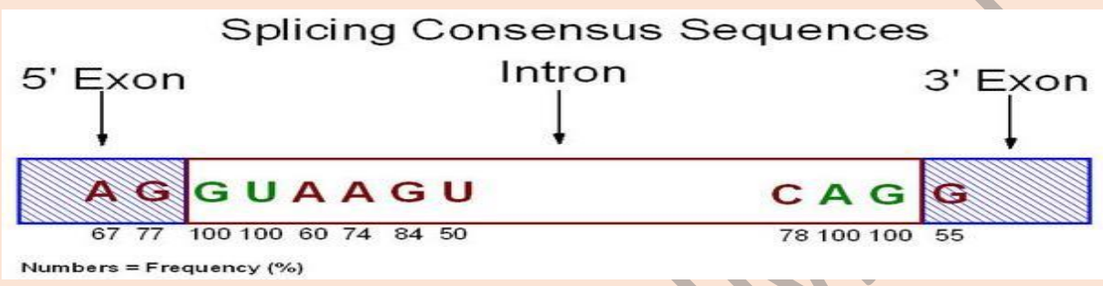
Biologically oriented

Approach

Recognize the locations of splicing signals at exon-intron junctions

There exists a weakly conserved sequence of eight

nucleotides at the boundary of an exon and an intron (donor splice site) and a sequence of four nucleotides at the boundary of an intron and exon (acceptor splice site)



Profiles for splice sites are weak-limited success Hidden Markov Model (HMM) approaches that capture statistical dependencies between sites GENSCAN developed by Chris Burge and Samuel Karlin. GENSCAN combines coding region and splicing signal predictions into a single framework.

Splice site prediction-coding region appear on one side of the site Such statistics are used in the HMM framework of GENSCAN that merges splicing site statistics, coding region statistics, and motifs near the start of the gene The accuracy of GENSCAN decreases for genes with many short exons or with unusual codon usage

Topic # 82 Similarity Based Approached to Gene Prediction 1

A similarity-based approach-Previously sequenced genes Unknown genes in newly sequenced DNA fragments Combinatorial puzzle-find a set of substrings (candidate exons) whose splicing best fits the target Brute force approach-find all local similarities

Each substring from the genomic sequence that exhibits sufficient similarity to the target protein could be considered a putative exon Exon-flanking dinucleotides **AG** and **GT**

Overlapping

Model a putative exon with a *weighted interval* in the genomic sequence, parameters (*l, r, w*)

l is the left-hand position, *r* is the right-hand position, and *w* is the weight of the putative exon "*w*"- local alignment score

Likelihood that this interval is an exon

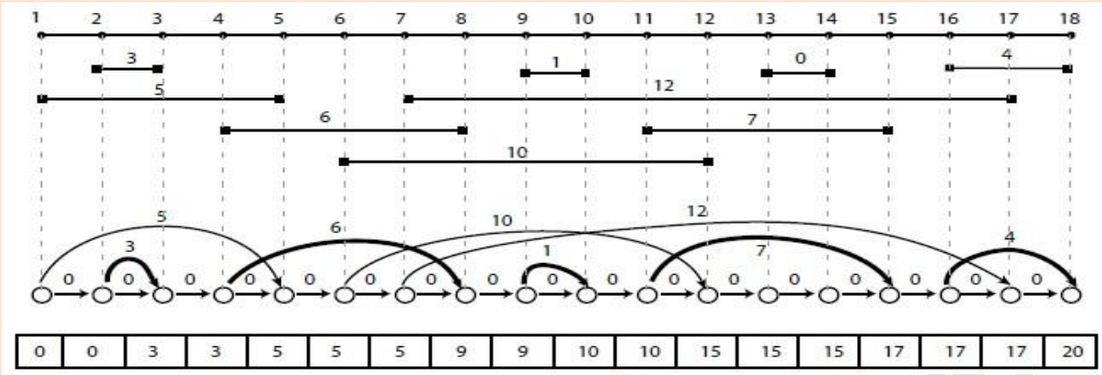
A chain is any set of non overlapping weighted intervals. Total weight of a chain

A maximum chain

Weights of all intervals are positive ($w > 0$)

Topic # 83 Similarity Based Approached to Gene Prediction 2

Model a putative exon with a *weighted interval* in the genomic sequence



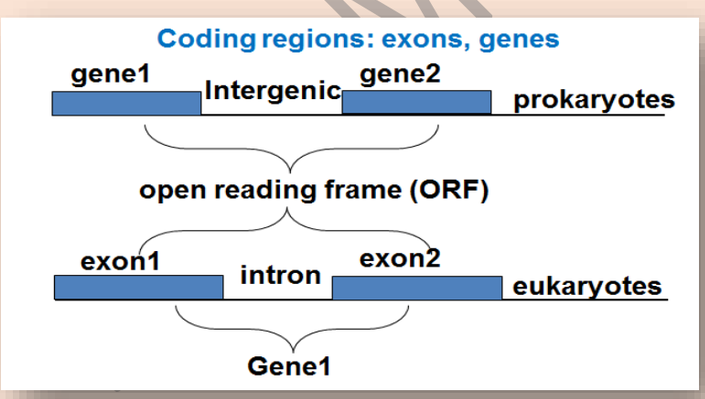
Five weighted intervals, (2, 3, 3), (4, 8, 6), (9, 10, 1), (11, 15, 7), and (16, 18, 4), shown by bold edges, form an optimal solution to the Exon Chaining problem. The array at the bottom shows the values s_1, s_2, . . . , s_{2n} generated by the EXONCHANING algorithm

Topic # 84 ORF Prediction

ORF (Open Reading Frame)

Gene finding, especially in prokaryotes starts form searching for open reading frames (ORF)

An ORF is a sequence of DNA that starts with start codon "ATG" (not always) and ends with any of the three termination codons (TAA, TAG, TGA)



Here, in this figure we see a comparison between prokaryotic and eukaryoticcells. Prokaryotes maybe bacteria or some related organisms whereas rest of the organisms is classified in eukaryotes.

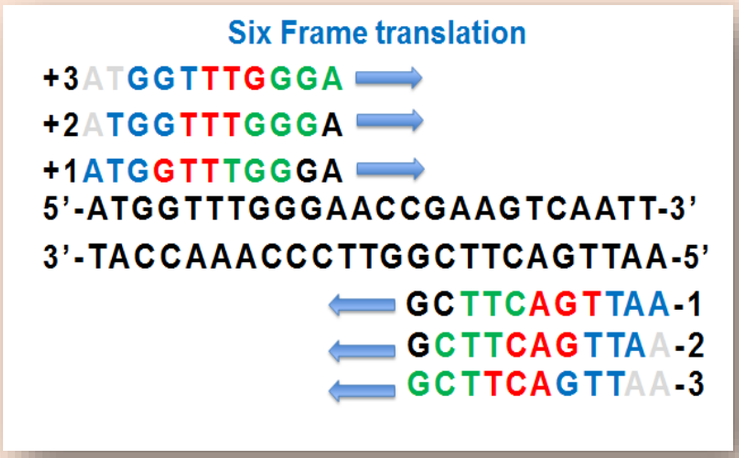
If we look into the prokaryotic situation, we have different genes which are separated by Intergenic regions and these ORF can be spanning these two genes whereas in case of the eukaryotes, we have only one gene, we have exon1 and exon2 whereas the ORF spans cross these different exons.
So on the top (in the figure), we have the longer ORF whereas in the bottom we have a shorter one as compared to the genome size those shorter ORF which are actually spanning different exons.

ORF and gene finding:

- ORF provide important evidence in gene finding.
- Generally longer ORFs are preferred.
- However presence of ORF not necessarily means the region is translated to a functional product

Reading Frames:

Depending on the start point, we can define different ORFs, so if we want to go for those triplet codons, we can start with any nucleotide, in this way we have three different possibilities for one of the strands and since we have two strands in the DNA, so in total we can have six ORFs (so 3 of them are from 3' to 5' direction whereas the other 3 are from 5' to 3' direction).



*Three on forward strand and three on complementary strand

This is how those 6 ORFs looks like.
So, there are the six frame translations (in this picture).
We observe that we have a top strand which is a forward strand and starts at 5' end and ends at 3' end.

A complementary runs in an anti-parallel fashion which starts with 3' end and ends at 5' end.

So, how can we get the reading frames out of it?
We can start with the position number 1 on the first strand and we label it as +1 (as shown), in this way we can start with A, and have triplet codons like ATG, GTT, TGG and so on.
In the second strand or second possibility (denoted by +2), we can start from the position number 2, so this frame starts from the second nucleotide like T and makes a triplet codon as TGG, TTT, GGG and so on.
So this how the third strand (3rd possible ORF; denoted by +3), where it starts from the position number 3 can be made. We don't start with the position number 4 because it will be same as the position number 1.

Similarly, we can do like this for the opposite direction strands with the possibility of position number 1, 2 and 3.

So, this how the 6 ORF are made.

Conclusions:

An ORF is a sequence of DNA that starts with start codon "ATG" (not always) and ends with any of the three termination codons (TAA, TAG, TGA) and there are 6 reading registers as far as the ORFs are considered from the sequence.

Reference:

Biological Sequence Analysis

R Durbin, S Eddy, A Krogh and G Mitchison

Cambridge University Press, 1998.

Bioinformatics The machine learning approach

P Baldi and S Brunak

The MIT Press, 1998

Post-Genome Informatics

M Kanehisa

MUHAMMAD IMRAN

Oxford University Press, 2000

Topic # 85 ORF Finders

ORF Finding:

- Long ORF may be a gene.
- Expected 64/3 ~ 21 codons before we see a stop codon.
- Genes are longer than this.
- We might scan for ORF longer than a threshold

Codon usage and likelihood ratio:

- An ORF is more 'reliable' if it has 'likely' codons
- We can do sliding window calculations (focus on some nucleotides like for example if the segment is 1000 long, we can make a window of say size 50 and we can look into the frequencies of different codons in that window) to ORF having 'likely' codon usage.
- An ORF is more 'reliable' if it has 'likely' codons
- However average vertebrate exon length (130 nucleotides) is too small for reliable peaks

An improvement may be;

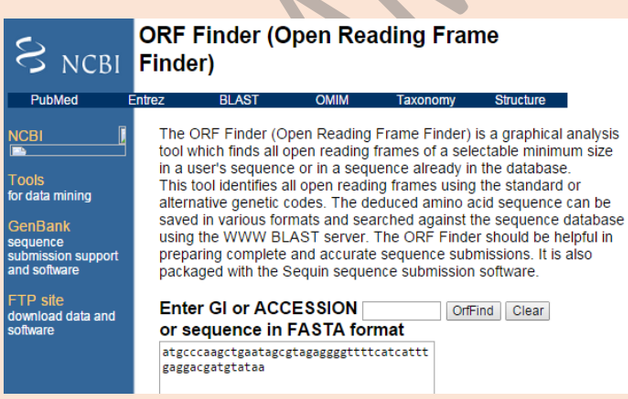
<u>In-Frame hexamer count</u>

i.e. frequencies of pairs of consecutive codons.

ORF Finders:

Tools are mainly based on pattern finding algorithms

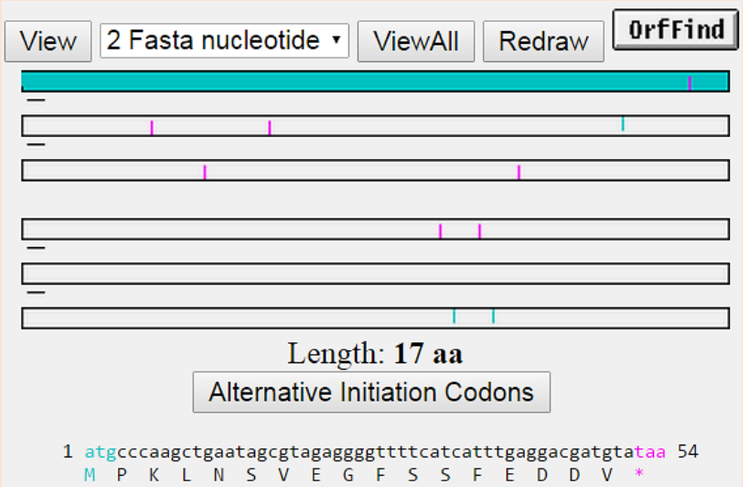
1. **NCBI's ORF Finder**
2. **ORF Investigator**
3. **OrfPredictor**



If you go to NCBI webpage, the ORF Finder is there.

You can place you sequence into the text box and upload it while simply clicking onto the 'OrfFind' button. You can even grasp the sequence automatically by writing the accession number in the space give after the word 'ACESSION'.

AMMAD IMRAN



When you click 'OrfFind', it will take you to this page where there are different registers or ORFs (explained in the previous lecture).

The steric (*) here means that the protein synthesis stops here.

Conclusions:

- ORF providesimportant evidence in gene finding.
- Generally longer ORFs are preferred.
- However presence of ORF not necessarily means the region is translated to a functional product

TOPIC 86 Translation Start Site (TSS)

Translation Start Site (TSS)

Translation starts with ATG that codes for methionine in a polypeptide

Here, in this example shown in the figure, we have those TSS in the middle (it's actually a file where sequences are aligned with one another), we look deeply, we can see that their neighbors are C and G which are most frequently seen (if not always). We might use these neighbors to identify the presence of these TSS.

Assumption:

- Certain nucleotides prefer to be around TSS than others.
- The "biased" nucleotide distribution is information is a basis for translation start prediction

Coding Potential:

Hexamer frequencies in coding versus non-coding regions may provide important insights Frequency of X(A,G,C,T) at position i is

F_i (X)=Σ log(C_i (X)/N_i(X))(*frequency of any nucleotide can be found by taking the sum of thelog of ratio of the counts of theparticular nucleotide in that particular position divided by the total)*

**where c is the counts.*

MUHAMMAD IMRAN

Frequency table (Biased Nucleotide distribution)

ATG

A	17.36,	19.01,	17.36,	48.76,	28.93,	15.70,		21.49,	23.14,	19.83,	21.49,	25.62,	15.70,
C	16.53,	28.93	57.85,	5.79	39.67	50.41,		22.31	38.84,	23.97,	27.27,	31.40,	38.02,
G	46.28,	29.75,	19.01,	42.98,	14.88,	26.45,		42.98,	23.97,	33.88,	32.23,	25.62,	29.75,
T	19.83,	22.31,	5.79,	2.48,	16.53,	7.44,		13.22,	14.05,	22.31,	19.01,	17.36,	16.53}
	-4	-3	-2	-1	ATG		+3	+4	+5	+6			

C A/G CC GCGG

Based upon the frequency equation, we can come up with a frequency table as shown in the figure.

The frequencies shown here indicates that you have the presence of true TSS here or you can expect that there are some Transcription Start Sites (TSS) here.

You can observe that on different positions like -4, -3, -2 and -1, similarly at +3, +4,+5, and +6 you have which nucleotides and what are their percentages (in the table).

Example

Which one is more probable to be a Translation Start?

CACC ATA GC

TCGA ATG TT

Solution

We can use frequency table and the scoring function as under;

$S_i = \Sigma \log (F_i (X)/0.25)$

-frequency from the frequency table divided by the expected frequency and then we convert it into log scale because we want to play with big numbers.

We can call this equation as the**Information Content (IC)**

Frequency table (Biased Nucleotide distribution)

ATG

A	17.36,	19.01,	17.36,	48.76	28.93,	15.70,		21.49,	23.14,	19.83,	21.49,	25.62,	15.70,
C	16.53,	28.93,	57.85,	5.79,	39.67,	50.41,		22.31,	38.84,	23.97,	27.27,	31.40,	38.02,
G	46.28,	29.75,	19.01,	42.98,	14.88,	26.45,		42.98,	23.97,	33.88,	32.23,	25.62,	29.75,
T	19.83,	22.31,	5.79,	2.48,	16.53,	7.44,		13.22,	14.05,	22.31,	19.01,	17.36,	16.53}
	-4	-3	-2	-1			+3	+4	+5	+6			

C A

Here, is our frequency table, we pick the frequencies from here.

MUHAMMAD IMRAN

CACC ATA GC	TCGA ATG TT
log (58/0.25) + log (49/0.25) + log (40/0.25) + log (50/0.25) + log (43/0.25) + log (49/0.25)	log (6/0.25) + log (6/0.25) + log (15/0.25) + log (7/0.25) + log (13/0.25) + log (14/0.25)
=13.69	= 9.44

And we add those frequencies here in the equation.
So, in this way we get a positive number in the end, so we have 13.69 on the left side whereas we have 9.44 on the right side.

So, which sequence has the strong evidence to have a translation site. In our case, we will prefer the one with higher value so its probably the green sequence.

Algorithm:

- Build a mathematical model, based on collected translation start sequence
- For each candidate translation start sequence, apply the model and get a score
- If the score is larger than zero, predict it is a "translation start"; the higher score, the higher the probability the prediction is true

Conclusions:

- TSS prediction can be an important step in gene prediction
- TSS can be predicted while using the frequency of neighboring nucleotides

References:

Biological Sequence Analysis

R Durbin, S Eddy, A Krogh and G Mitchison

Cambridge University Press, 1998.

Bioinformatics The machine learning approach

P Baldi and S Brunak

The MIT Press, 1998

Post-Genome Informatics

M Kanehisa

Oxford University Press, 2000

TOPIC 87 Prediction of splice junctions

Splice Junctions:

Donor site

- Coding region | GT (introns starts)

Acceptor

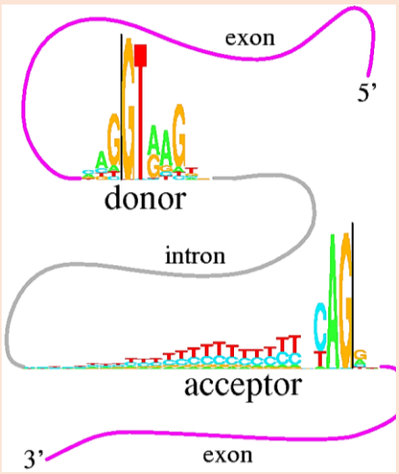
- (introns ends)YAG | coding region
  -Y can be any pyrimidine.
- **Canonical form**
  - GT-AG: 99.24%

There are also some non-canonical forms.

MUHAMMAD IMRAN

Like TSS, the flanks of splice junctions show “biased” distributions of nucleotides in certain positions

- These biased distributions of nucleotides are the basis for prediction of splice junctions



Here, is the example where we can see we have the exon then we have the donor site where we see a big GT (this representation is known as sequence logos), then we have the intron region which is followed by the acceptor region which ends up with AG and then again a next exon starts.

Sequence LOGOS:

- A visual representation of a position-specific distribution
- Easy for nucleotides, but we need colour to depict up to 20 amino acid proportions.

Overall height at position is proportional to the information content

- Proportions of each nucleotide/amino acid are in relation to their observed frequency, with most frequent on top, next most frequent below

Non Canonical Splice Junctions:

In addition to canonical **GT-AG (99.24%)**;

GC-AG: 0.69%

AT-AC: 0.05%

Others: 0.02%

Information Content (IC):

$S_i = \Sigma \log (F_i (X)/0.25)$

- If every nucleotide has 0.25 frequency in a position, then the position’s information content is ZERO.
- Use “information content as a criterion for determining the length of flanks

MUHAMMAD IMRAN

Accepter site prediction

	-6	-5	-4	-3	-2	-1	1
A	12.7	9.5	26.2	6.3	100	000	21.4
C	40.5	36.5	33.3	68.2	000	000	2.0
G	2.4	6.3	13.5	000	000	100	62.7
T/U	44.5	47.6	27.0	25.2	000	000	7.90

Multiple positions have high information content

Here, is the acceptor site distributions and we can observe that at position number -2 and -1, there are the presence of the acceptor sites.
100% frequencies of A and G, so these are predicted but if we look into the neighbors like at position -3, there are mostly Cs.

Donor site prediction

	-3	-2	-1	1	2	3	4
A	34.0	60.4	9.2	000	000	52.6	71.3
C	36.3	12.9	3.3	000	000	2.8	7.6
G	18.3	12.5	80.3	100	000	41.9	11.8
T/U	11.4	14.2	7.3	000	100	2.5	9.3

Multiple positions have high information content

We can look into the donor sites; we have 100% frequency for Gs and Ts, we label them as 1 and 2, and different positions which are neighboring to them, we can include 10 or 15 neighbors.

Algorithm:

Mathematical model: F_i (X): frequency of X (A, C, G, T) in position I

Score a segment as a candidate donor/acceptor site by

$\Sigma \log (F_i (X)/0.25)$

For each candidate sequence, apply the model and get a score

If the score if larger than zero, predict it is "donor/acceptor"; the higher score, the higher the probability the prediction is true

Conclusions:

Like TSS, the flanks of splice junctions show "biased" distributions of nucleotides in certain positions

- These biased distributions of nucleotides can be used for prediction of splice junctions

References:

Biological Sequence Analysis

R Durbin, S Eddy, A Krogh and G Mitchison

Cambridge University Press, 1998.

Bioinformatics The machine learning approach

P Baldi and S Brunak

The MIT Press, 1998

Post-Genome Informatics

M Kanehisa

Oxford University Press, 2000

MUHAMMAD IMRAN

TOPIC 88 Prediction of Exons

Introduction:

Exons can be predicted within ORF from the information gathered about

- splice junctions

Approach:

For each segment [acceptor, donor], we get three scores (coding potential, donor score, acceptor score)

<u>Various possibilities</u>

- all three scores are high – probably true exon.
- all three scores are low – probably not a real exon.
- all in the middle -- ?.
- some scores are high and some are low -- ??

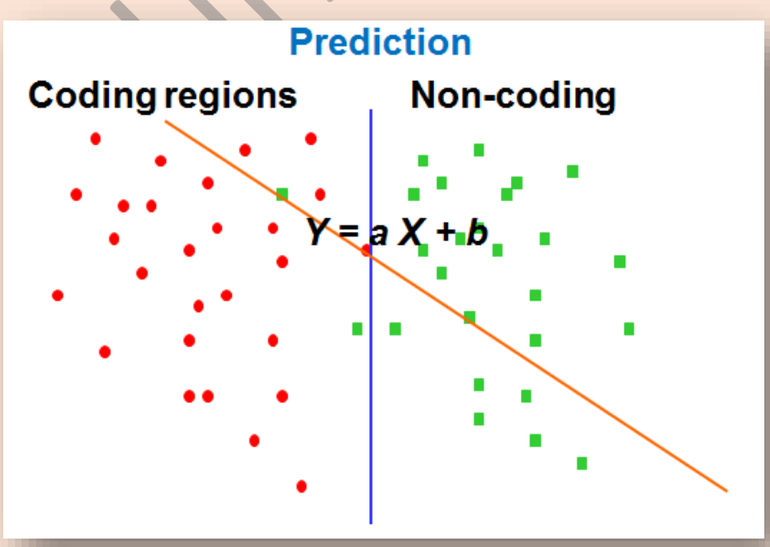
So here, we can get the evidence by the help of that information which we gathered from those splice sites.

Prediction:

- Collect a set of exons and non-exons
- Score them using our scoring schemes
- Plot them as follows
- "draw" a separating line between exons and non-exons

Linear Discriminate Analysis:

- linear discriminate analysis (LDA) finds an optimal plane surface that best separates points that belong to two classes.
- For example, if there are ten true exons and ten introns, and two feature.
- These samples could be represented by 20 points in a two-dimensional space.
- LDA would compute a straight line through the space that can best separate the two classes with the minimal classification error



For example, if we look into this picture, there is the coding region and the non-coding region and there is a line in the middle that tries to separate these two which helps in discrimination.

We can draw a central line or linear regression line, we can see the equation over there and by drawing this line we can have the prediction, this line fits in such a way that it fits into most of the data points.

MUHAMMAD IMRAN

If we don't know about data point, we can predict it while using this prediction line (angular line in the figure).

Conclusion:

- Collect a set of exons and non-exons
- Score them using our scoring schemes
- Plot them as follows
- "draw" a separating line between exons and non-exons

References:

Biological Sequence Analysis

R Durbin, S Eddy, A Krogh and G Mitchison

Cambridge University Press, 1998.

Bioinformatics The machine learning approach

P Baldi and S Brunak

The MIT Press, 1998

Post-Genome Informatics

M Kanehisa

Oxford University Press, 2000

TOPIC 89 Annotation of Assembled Genome

A genome sequence is useless without annotation

Three steps in genome annotation:

- Find features not associated with protein-coding genes (e.g. tRNA, rRNA, snRNA, SINE/LINE, miRNA precursors)

A genome sequence is useless without annotation

Three steps in genome annotation:

- Build models for protein-coding genes, including exons, coding regions, regulatory regions

A genome sequence is useless without annotation

Three steps in genome annotation:

- Associate biologically relevant information with the genome features and genes

Ab initio methods:

Based on sequence alone

- Gene prediction algorithms (e.g. AUGUSTUS, Glimmer, GeneMark)
- RepeatMasker(repeat families)

Evidence-based Methods:

- Require transcriptome data for the target organism (the more the better)
- Align cDNA sequences to assembled genome and generate gene models: TopHat/Cufflinks, Scripture

MUHAMMAD IMRAN

Biological Annotations:

BLAST of gene models against protein databases

- Sequence similarity to known proteins
- InterProScan of predicted proteins against databases of protein domains
  - Pfam, Prosite, HAMAP, PANTHER, …
- Mapping against Gene Ontology (function of those genes) terms
  - GO terms
  - BLAST2GO

Pattern Finding:

- Much of the data processing in bioinformatics involves searching and recognizing certain patterns within DNA, RNA or protein sequences.
- In Biology it means finding motifs in DNA or proteins while in computational means it is finding a pattern in a string

Conclusions:

After a genome is assembled, genome annotations are performed to identify gene and other features in a genome

TOPIC 90 Pattern Finding in a Genome

Vocabulary:

- A **pattern** (keyword) is an ordered sequence of symbols .
- Symbols of the pattern and the searched text are chosen from a predetermined finite set, called an **alphabet** (Σ)

Four Cases of Pattern Finding:

- Look for a perfect match

CGTA
CGTA

- Allow errors due to substitutions

CGTA
CGGA

- Allow errors due to insertions-deletions

(InDels).

C_GTA
CCGGA

MUHAMMAD IMRAN

Rank possible matches according to a weight function and keep matches above a certain threshold

Four Cases of Pattern Finding:

p1 and ***p2*** are two patterns of length 5

CTGTA
CCGGA

- ***W*** is the weight of complete patterns defined via a nucleotide-nucleotide weight function ***w( )***

$$W(p1,p2) = \sum_{i=1}^{5} w(p1[i],p2[i])$$

For every pair, aligned together, we come up with the weight function and in the end we combine all those weight functions

Generalized Algorithm:

Goal: Finding all occurrences of a pattern in a text

Input:

Pattern **p = [p1…pn]** of length ***n***

Text **t = [t1…tm]** of length ***m***

Output:

An indication that pattern P exist in T

or it does not exist in text T

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Text:	a	t	a	c	a	a	t	a	t	a	c	a	t	a
Pattern:								a	t	a	c	a	t	

Here P is a substring of T i.e., P=T[7,…,12]

Here we are using arrays of data structure, so we have text in those arrays and the indexes are the positions of those letters.

The total length is 14 (starting from 0 and ending at 13), and the pattern length is 6.

So, the pattern matches on the seventh index, so what we will get in the end is that P is a substring of T, starting from 7th index to 12th.

Conclusion:

Pattern searching algorithms search specific sequences in strands of DNA, RNA and proteins having important biological meaning

MUHAMMAD IMRAN

TOPIC 91 Pattern Finding Algorithms

Methods Devised for Pattern Finding:

- Exact searching methods
- Approximate searching methods
- Position weight matrices
- Suffix trees

Exact Pattern Matching:

- Given a pattern ***p*** of length **m**
- and a string or text ***T*** of length ***n*** ***(m ≤ n)***

- Find all the occurrences of ***p*** in ***T***

The matching needs to be exact, which means that the exact word or pattern is found

Exact Pattern Matching Algorithms:

- Naïve Brute Force algorithm
- Boyer-Moore algorithm
- Knuth Morris Pratt algorithm

Approximate Pattern Matching:

Also referred as approximate string matching or matches with ***k*** mismatches or differences

Also referred as approximate string matching or matches with ***k*** mismatches or differences

Given: a pattern ***p*** of length ***m***

and

a string or text ***T*** of ***length n (m ≤ n)***

Find: all the occurrences of substring ***X*** in ***T*** that are similar to ***p***, allowing a limited number, say ***k*** different characters in similar matches

Approximate Pattern Matching Algorithms

- Dynamic programming approach
- Automata approach
- Filtering and automation algorithms

Position Weight Matrices:

Also known as position specific scoring matrices (PSSM)

- A matrix representing the frequencies of residues observed for a position in multiple alignment

Suffix Trees:

It is a compressed tree containing all the suffixes and allows many problems on strings to be solved quickly

Conclusions:

- Exact searching or pattern matching methods
- Approximate searching or pattern matching methods
- Position weight matrices.
- Suffix trees

TOPIC 92 Methods Devised for Pattern Finding

Introduction:

Also known as exhaustive search algorithm

- All the possibilities are explored and the best one is chosen
- For the task with many possibilities brute force will take too much time

Working:

- Searches patterns by going through the whole sequence nucleotide per nucleotide.
- Always shifts the window by exactly one position to right
- Requires 2n expected text character comparisons

MUHAMMAD IMRAN

When a mismatch the comparison stops and starts again by moving the pattern one position forward

Algorithm:

Brute_Force(T,P)

n length[T]

m length[P]

For s 0 to n-m

Do if P[1..m]= T[s+1…s+m]

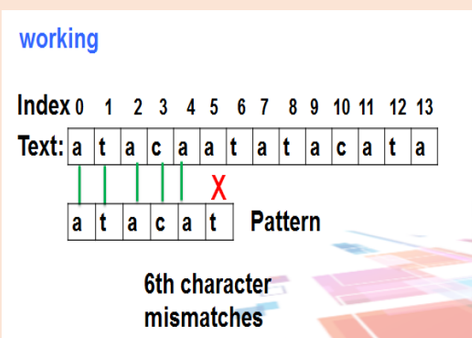
print "pattern occurs at position" s+1

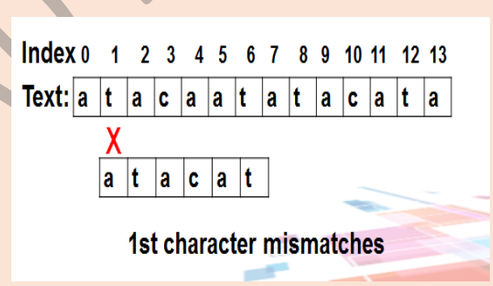
Here, Brute_Force is the function which has two arguments (T= text and P= pattern) where n records the length of T and m records the length of P and we start with a for loop which goes from 0 to n-m, say for example we have 'T' which is of length 10 and 'P' which is of length 5 so it starts from 0 and will go till 5.

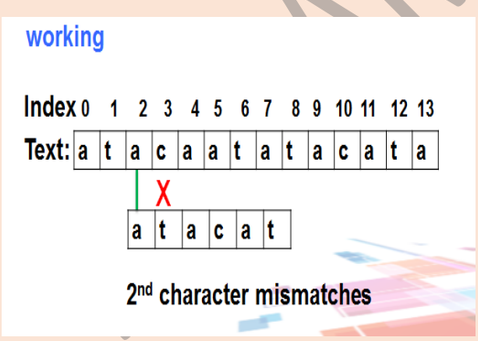
Do if P[1..m]= T[s+1…s+m]

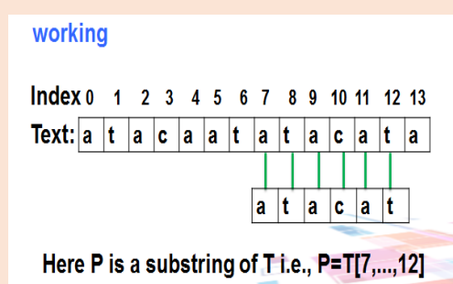
This line is where we are saying that nucleotide number 1 of the pattern and we go up to its whole length. In case of the text, we started with 0, so we add one over here, so 0+1 i.e. the 1st nucleotide is compared till the last nucleotide.

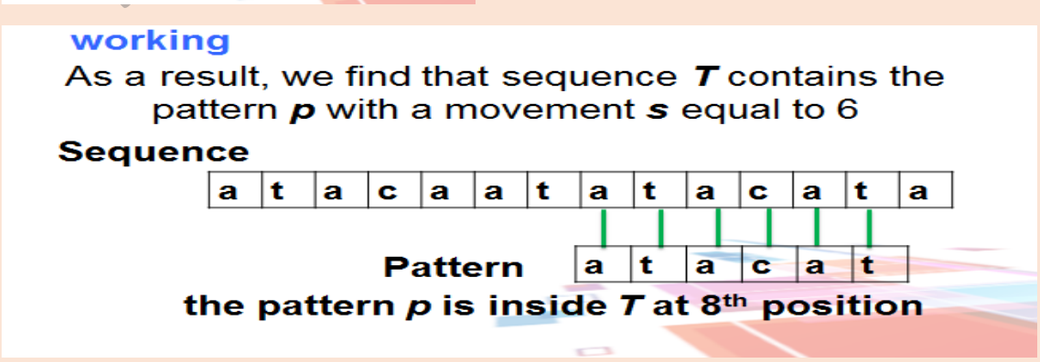
So, if we find the occurrence of the patterns we will put that in the print statement and s+1 will give its position.











Drawback:

The repetitive use of residues in comparison leads to runtime of **O(mn)**, which makes it very slow

Conclusion:

Brute force is an exhaustive search method that takes long time as it does nucleotide by nucleotide comparison

TOPIC 93 Knuth-Morris-Pratt Algorithm

Introduction:

A linear time algorithm for string matching

- Does not involve backtracking on string ***s*** i.e., repetitive comparison of nucleotide residues

Components:

- **The Prefix Function Π**
- **The KMP Matcher**

The Prefix Function Π:

Encapsulates knowledge about how the pattern matches against shifts of itself

- This information can be used to avoid useless shifts of the pattern ***'p'***
- This enables avoiding backtracking on the string ***'S'***

The KMP Matcher:

Given: string '***S***', pattern '***p***' and prefix function **'Π'**

Find: the occurrence of '***p***' in '***S***'

Return: the number of shifts of '***p***' after which occurrence is found

```
Compute-Prefix-Function (p)
m ← length[p]      //'p' pattern to be matched
Π[1] ← 0
k ← 0
   for q ← 2 to m
     do while k > 0 and p[k+1] != p[q]
       do k ← Π[k]
       If p[k+1] = p[q]
       then k ← k +1
      Π[q] ← k
    return Π
```

```
The KMP Matcher (S, p)
n ← length[S]
m ← length[p]
Π ← Compute-Prefix-Function(p)
q ← 0    //number of characters matched
for i ← 1 to n   //scan S from left to right
do while  q > 0 and p[q+1] != S[i]
 do  q ← Π[q]  //next character does not
 match
 if p[q+1] = S[i]
    then q ← q + 1      //next character
 matches
```

IMAD IMRAN

The KMP Matcher (S, p)

```
if q = m              //is all of p matched?
    then print "Pattern occurs at position"
        i–m+1
        q ← Π[ q]     // look for the next
match
```

Conclusions:

A linear time algorithm for string matching

- avoid useless shifts of the pattern ***'p'***
- Does not involve backtracking on string ***S***

Topic 94 Knuth-Morris-Pratt Algorith

Components

- **The Prefix Function Π**
- **The KMP Matcher**

Compute-Prefix-Function (p)

```
m ← length[p]
Π[1] ← 0
k ← 0
for q ← 2 to m
    do while k > 0 and p[k+1] != p[q]
        do k ← Π[k]
        If p[k+1] = p[q]
        then k ← k +1
    Π[q] ← k
return Π
```

q	1	2	3	4	5	6	7
p	a	t	a	t	a	c	a
Π	0						

Initially:

m = length[p] = 7
Π[1] = 0
k = 0

MUHAMMAD IMRAN

Compute-Prefix-Function (p)

```
m ← length[p]
Π[1] ← 0
k ← 0
    for q ← 2 to m
        do while k > 0 and p[k+1] != p[q]
            do k ← Π[k]
        If p[k+1] = p[q]
        then k ← k +1
        Π[q] ← k
    return Π
```

q	1	2	3	4	5	6	7
p	a	t	a	t	a	c	a
Π	0	**0**					

Step 1: **q = 2, k=0**

$p[0+1] \neq p[q_2]$

Π[2] = 0

Compute-Prefix-Function (p)

```
m ← length[p]
Π[1] ← 0
k ← 0
    for q ← 2 to m
        do while k > 0 and p[k+1] != p[q]
            do k ← Π[k]
        If p[k+1] = p[q]
        then k ← k +1
        Π[q] ← k
    return Π
```

q	1	2	3	4	5	6	7
p	a	t	a	t	a	c	a
Π	0	0	**1**				

Step 2: **q = 3, k = 0,**

$p[0+1] = p[q_3]$

K = 0+1 = 1

Π[3] = 1

Compute-Prefix-Function (p)

```
m ← length[p]
Π[1] ← 0
k ← 0
    for q ← 2 to m
        do while k > 0 and p[k+1] != p[q]
            do k ← Π[k]
            If p[k+1] = p[q]
            then k ← k +1
        Π[q] ← k
    return Π
```

q	1	2	3	4	5	6	7
p	a	t	a	t	a	c	a
Π	0	0	1	**2**			

Step 3: q = 4, k = 1

$p[1+1] = p[q_4]$

K = 1+1 = 2

Π[4] = 2

Compute-Prefix-Function (p)

```
m ← length[p]
Π[1] ← 0
k ← 0
    for q ← 2 to m
        do while k > 0 and p[k+1] != p[q]
            do k ← Π[k]
            If p[k+1] = p[q]
            then k ← k +1
        Π[q] ← k
    return Π
```

q	1	2	3	4	5	6	7
p	a	t	a	t	a	c	a
Π	0	0	1	2	**3**		

Step 4: q = 5, k =2

$p[2+1] = p[q_5]$

K = 2+1 = 3

Π[5] = 3

Compute-Prefix-Function (p)

```
m ← length[p]
Π[1] ← 0
k ← 0
    for q ← 2 to m
      do while k > 0 and p[k+1] != p[q]
        do k ← Π[k]
        If p[k+1] = p[q]
        then k ← k +1
      Π[q] ← k
    return Π
```

q	1	2	3	4	5	6	7
p	a	t	a	t	a	c	a
Π	0	0	1	2	3	**1**	

Step 5: **q = 6, k = 3**

$p[3+1] \neq p[q_6]$

K = Π[3] = 1

Π[6] = 1

Compute-Prefix-Function (p)

```
m ← length[p]
Π[1] ← 0
k ← 0
    for q ← 2 to m
      do while k > 0 and p[k+1] != p[q]
        do k ← Π[k]
        If p[k+1] = p[q]
        then k ← k +1
      Π[q] ← k
    return Π
```

q	1	2	3	4	5	6	7
p	a	t	a	t	a	c	a
Π	0	0	1	2	3	1	**0**

Step 6: **q = 7, k = 1**

$p[1+1] \neq p[q_7]$

K = Π[1] = 0

Π[7] = 0

Conclusions

- **The Prefix Function Π**
- **The KMP Matcher**

Topic # 95 Knuth-Morris-Pratt Algorithm

The KMP Matcher (S, p)

```
n ← length[S]
m ← length[p]
Π ← Compute-Prefix-Function(p)
q ← 0    //number of characters matched
for i ← 1 to n   //scan S from left to right
do while  q > 0 and p[q+1] != S[i] //if next    character does not match
do  q ← Π[q]
if p[q+1] = S[i] //next character matches
then q ← q + 1
if q = m            //is all of p matched?
then print "Pattern occurs at position"                    i – m+1
      q ← Π[ q]    // look for the next match
```

Let us execute the KMP algorithm to find whether 'p' occurs in 'S'.

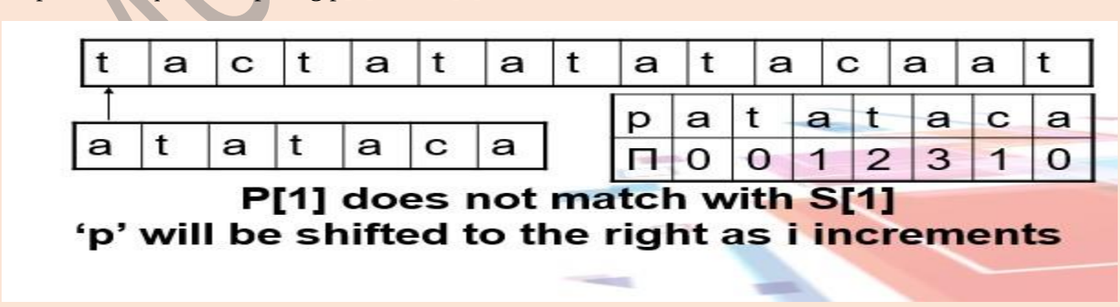
For 'p' the prefix function, Π was computed previously and is as follows:

q	1	2	3	4	5	6	7
p	a	t	a	t	a	c	a
Π	0	0	1	2	3	1	0

Initially: n = size of S = 15;

m = size of p = 7

Step 1: i = 1, q = 0 comparing p[1] with S[1]



MUHAMMAD IMRAN

Step 2: p[q+1] : S[i]

i = 2, q = 0 comparing p[1] with S[2]

S	t	a	c	t	a	t	a	t	a	t	a	c	a	a	t

p	a	t	a	t	a	c	a

if p[q+1] = S[i]
then q ← q + 1

P[1] matches S[2]

q=0+1 = 1

p	a	t	a	t	a	c	a
Π	0	0	1	2	3	1	0

Step 3: i = 3, q = 1 p[q+1] : S[i]

p[2] does not match with S[3]

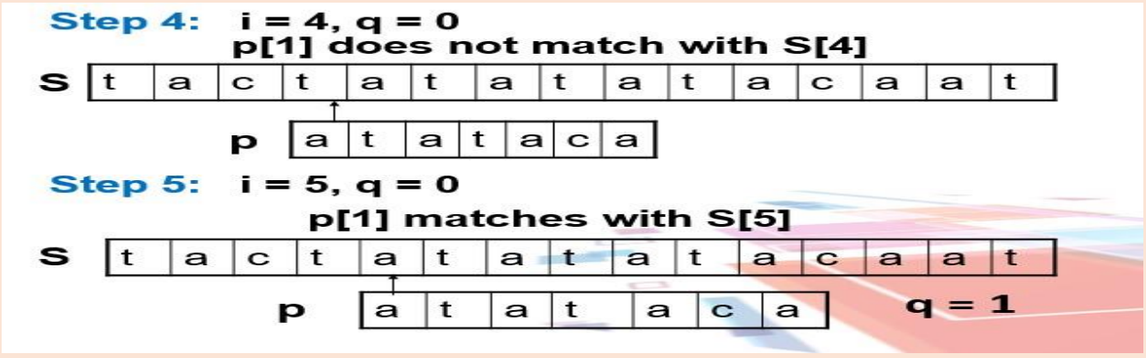
S	t	a	c	t	a	t	a	t	a	t	a	c	a	a	t

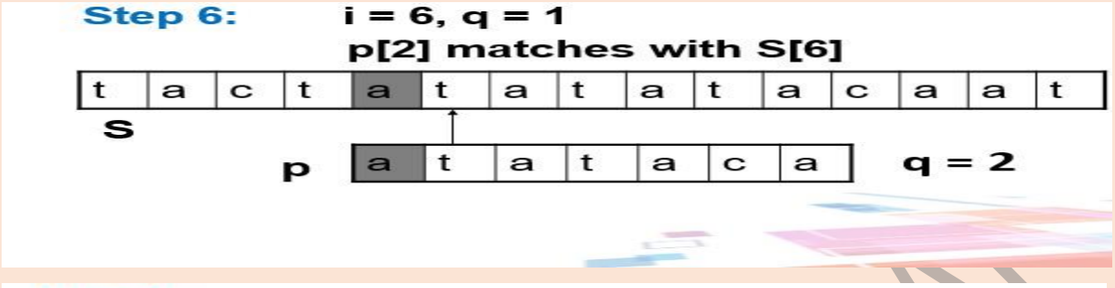
p	a	t	a	t	a	c	a

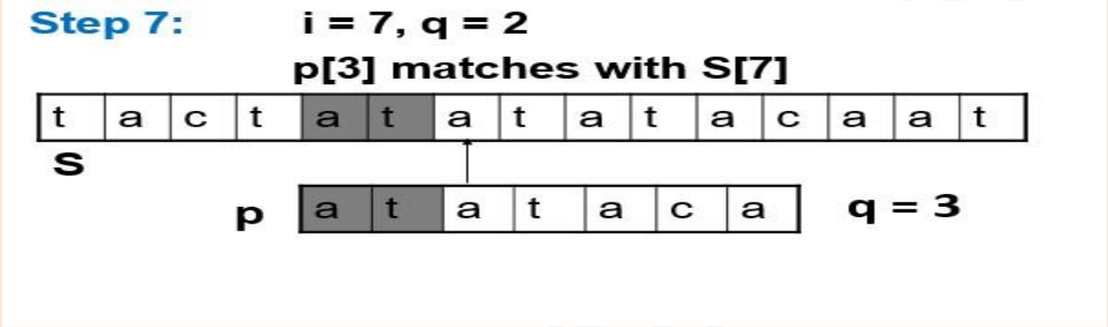
q > 0 and
p[q+1] != S[i]
do q ← Π[q]

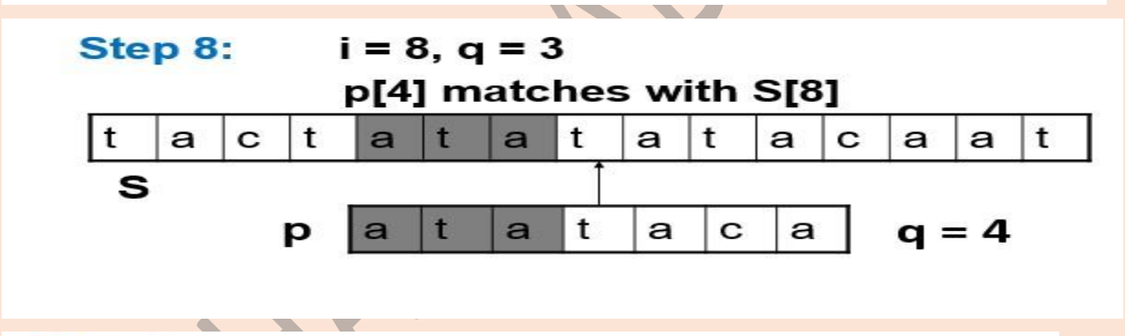
Backtracking on p, comparing p[1] and S[3]
because after mismatch q = Π[1] = 0

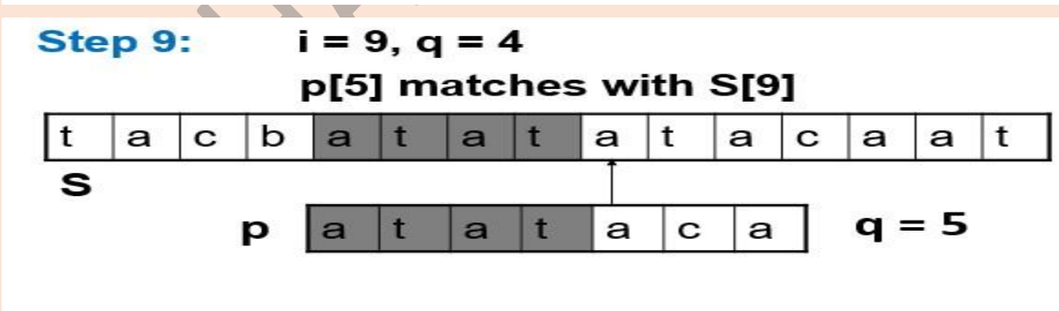
p	a	t	a	t	a	c	a
Π	**0**	0	1	2	3	1	0

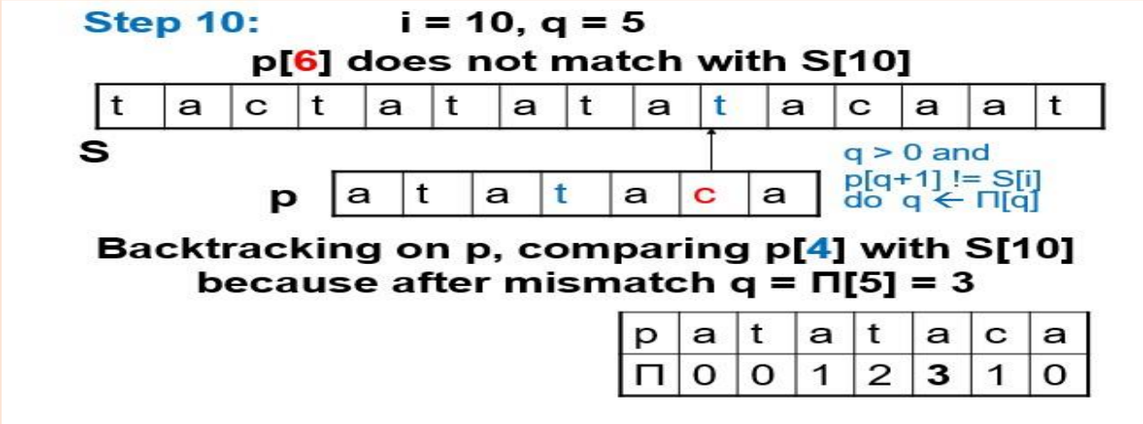

Step 4: i = 4, q = 0
p[1] does not match with S[4]
S t a c t a t a t a t a c a a t
p a t a t a c a
Step 5: i = 5, q = 0
p[1] matches with S[5]
S t a c t a t a t a t a c a a t
p a t a t a c a
q = 1

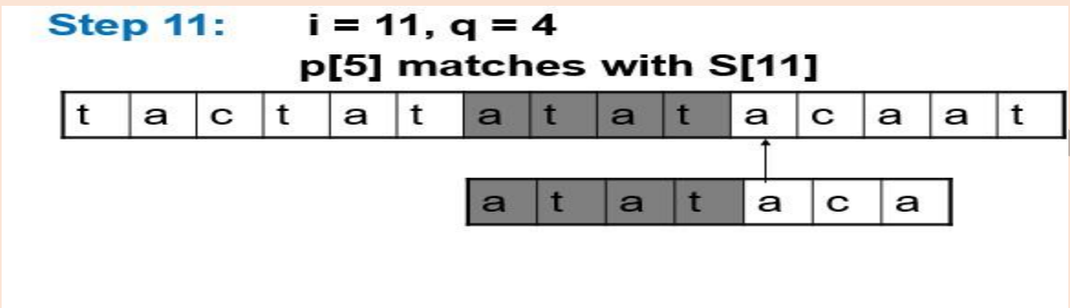

Step 6: i = 6, q = 1
p[2] matches with S[6]
t a c t a t a t a t a c a a t
S
p a t a t a c a
q = 2

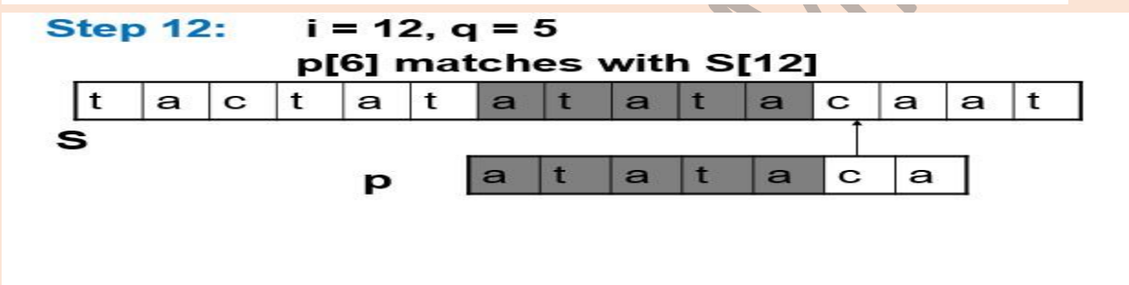

Step 7: i = 7, q = 2
p[3] matches with S[7]
t a c t a t a t a t a c a a t
S
p a t a t a c a
q = 3

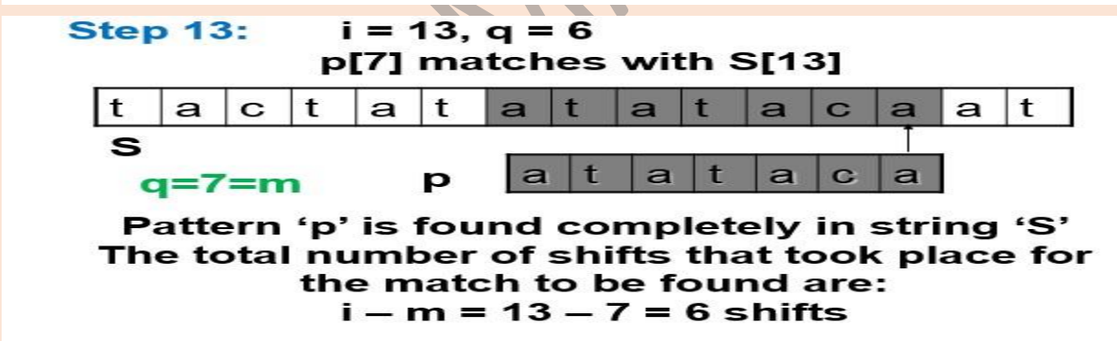

Step 8: i = 8, q = 3
p[4] matches with S[8]
t a c t a t a t a t a c a a t
S
p a t a t a c a
q = 4


Step 9: i = 9, q = 4
p[5] matches with S[9]
t a c b a t a t a t a c a a t
S
p a t a t a c a
q = 5









Complexity

- O(m) - It is to compute the prefix function values.
- O(n) - It is to compare the pattern to the text.
- Total of O(m + n) run time.

Advantages

- The running time of the KMP algorithm is optimal (O(m + n)), which is very fast
- The algorithm never needs to move backwards that makes the algorithm good for processing very large files

Drawback

- Doesn't work so well as the size of the alphabets increases Conclusion

- A fast linear time algorithm for string matching

96.Scoring Scheme

Scoring System: Introduction:

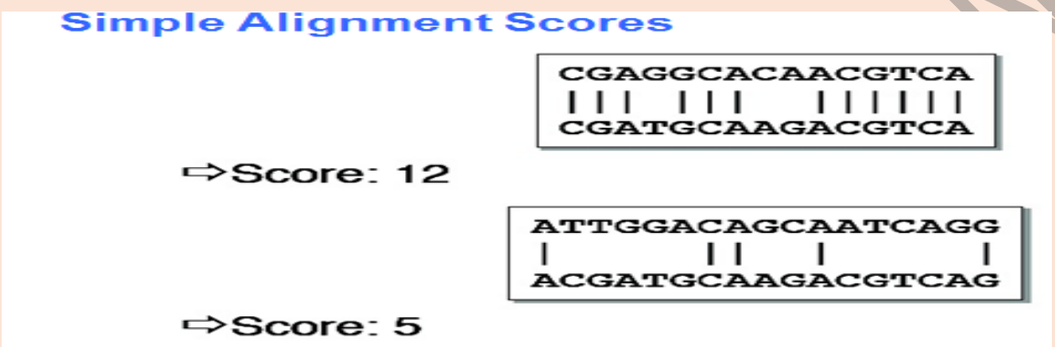
- Total score assigned to an alignment is sum of terms for each aligned pair of residues, plus terms for each gap

$$\sum S(x_i,y_j) + d$$

d = linear gap penalty

Simple Alignment Scores:

- A simple way (but not the best) to score an alignment is to count 1 for each match and 0 for each mismatch



Scoring or Substitution Matrices:

- Used for scoring amino acid substitutions in pairwise alignments
- They reflect substitution rates that are originated by evolutionary events

Some of the substitution matrices to compute sequence alignments are:

- PAM: Point Accepted mutations
- BLOSUM: BLOCK Substitution Matrix

Substitution Matrices:

For a set of well known proteins:

- Align the sequences
- Count the mutations at each position
- For each substitution set the score to the log-odds ratio

For each substitution set the score to the log-odds ratio

$$log\left(\frac{observed}{expected\ by\ chance}\right)$$

Positive Score:

The amino Acids are similar, mutations from one into the other occur more often than expected by chance during evolution

Negative Score:

The amino Acids are dissimilar, mutations from one into the other occur less often then expected by chance during evolution

Log Odd PAM 250 Matrix

	C Cys	S Ser	T Thr	P Pro	A Ala	G Gly	N Asn	D Asp	E Glu	Q Gln	H His	R Arg	K Lys	M Met	I Ile	L Leu	V Val	F Phe	Y Tyr	W Trp
C Cys	12																			
S Ser	0	2																		
T Thr	−2	1	3																	
P Pro	−3	1	0	6																
A Ala	−2	1	1	1	2															
G Gly	−3	1	0	−1	1	5														
N Asn	−4	1	0	−1	0	0	2													
D Asp	−5	0	0	−1	0	1	2	4												
E Glu	−5	0	0	−1	0	0	1	3	4											
Q Gln	−5	−1	−1	0	0	−1	1	2	2	4										
H His	−3	−1	−1	0	−1	−2	2	1	1	3	6									
R Arg	−4	0	−1	0	−2	−3	0	−1	−1	1	2	6								
K Lys	−5	0	0	−1	−1	−2	1	0	0	1	0	3	5							
M Met	−5	−2	−1	−2	−1	−3	−2	−3	−2	−1	−2	0	0	6						
I Ile	−2	−1	0	−2	−1	−3	−2	−2	−2	−2	−2	−2	−2	2	5					
L Leu	−6	−3	−2	−3	−2	−4	−3	−4	−3	−2	−2	−3	−3	4	2	6				
V Val	−2	−1	0	−1	0	−1	−2	−2	−2	−2	−2	−2	−2	2	4	2	4			
F Phe	−4	−3	−3	−5	−4	−5	−4	−6	−5	−5	−2	−4	−5	0	1	2	−1	9		
Y Tyr	0	−3	−3	−5	−3	−5	−2	−4	−4	−4	0	−4	−4	−2	−1	−1	−2	7	10	
W Trp	−8	−2	−5	−6	−6	−7	−4	−7	−7	−5	−3	2	−3	−4	−5	−2	−6	0	0	17
	C Cys	S Ser	T Thr	P Pro	A Ala	G Gly	N Asn	D Asp	E Glu	Q Gln	H His	R Arg	K Lys	M Met	I Ile	L Leu	V Val	F Phe	Y Tyr	W Trp

Raw score of an alignment:

T P E A

A P G A

Score: 1+6+0+2=9

Conclusion

- Substitution matrices are the log-odds matrices used for scoring amino acid substitutions in pairwise alignments

97.Substitution Matrices

Introduction:

- Substitution scores can be derived from probabilistic model

Notations:

- Let a pair of sequences x and y of length *n* and *m*
- X_i be the ith symbol in x y_j be the jth symbol in y
- Symbols are from alphabet A
  - A={A,T,G,C}
- A={twenty amino acids}
- Symbols from alphabet be a,betc

Objective:

Given a pair of aligned sequences, we want to assign a score to the alignment that gives a measure of the relative likelihood that the sequences are related as opposed to being unrelated

Unrelated or random Model R:

Letter a occurs independently with frequency $\mathbf{q_a}$and the probability of two sequences is the product of probability of each amino acid

$$P(x,y|R)=\prod_i q_{xi} \prod_j q_{yj}$$

MUHAMMAD IMRAN

Match Model M:

- Aligned pairs occur with a joint probability p_{ab}.
- p_{ab} can be thought of as the probability that the residues ***a*** and ***b*** have been independently derived from some unknown original residue c in their common ancestor .
- The probability of the alignment is

$$P(x,y|M) = \prod_i p_{xiyj}$$

- The ratio of two likelihoods can be calculated as;

$$\frac{P(x,y|M)}{P(x,y|R)}$$

Odds ratio:

- The ratio of two likelihoods can be calculated as;

$$\frac{P(x,y|M)}{P(x,y|R)} = \frac{\prod_i p_{xiyj}}{\prod_i q_{xi} \prod_j q_{yj}} = \prod_i \frac{p_{xiyj}}{q_{xi} q_{yj}}$$

- Logarithm can be used to have an additive score

$S = \sum S(x_i, y_j)$

where $S(a,b) = \log(p_{ab}/q_a q_b)$

log likelihood of (a,b) as aligned vs unaligned pair

Dayhoff PAM matrices:

Dayhoff, Schwartz and Orcutt (1978) presented their famous PAM (Point accepted mutations) using substitution data from similar proteins then extrapolating this information to longer evolutionary distances

$S(a,b) = \log(p_{ab}/q_a q_b)$

incorporating the evolutionary time

$S(a,b|t) = \log P(b|a,t)/q_b$

Since $p_{ab}/q_a = P(b|a)$

Values are rounded to near integer for computational convenience

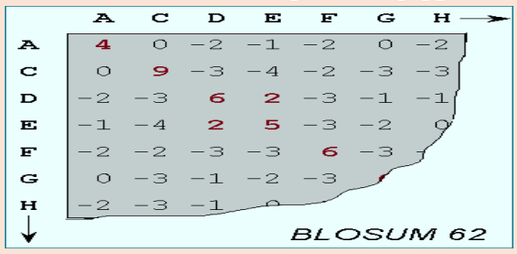
- PAM250 is scaled by 3/log2 to give scores in third-bits

BLOSUM matrices:

- Dayhoff matrices do not capture true difference between short time substitutions and long term ones.
- PAM matrices do not perform well in case of distantly related proteins.
- BLOSUM matrices are derived from set of aligned, ungapped regions from protein families called BLOCKS database (Henikoff&Henikoff 1992).
- Sequences from each block was clustered together with score >***L%.***
- Matrices with ***L= 62*** and ***L= 50*** known as **BLOSUM62** and **BLOSUM50** respectively.

MUHAMMAD IMRAN

- **BLOSUM62** is good for ungapped alignments and **BLOSUM50** is good for gapped alignments



	A	C	D	E	F	G	H
A	4	0	-2	-1	-2	0	-2
C	0	9	-3	-4	-2	-3	-3
D	-2	-3	6	2	-3	-1	-1
E	-1	-4	2	5	-3	-2	0
F	-2	-2	-3	-3	6	-3	[illegible]
G	0	-3	-1	-2	-3	[illegible]	[illegible]
H	-2	-3	-1	0	[illegible]	[illegible]	[illegible]

BLOSUM 62

Conclusions

- Substitution scores can be derived from probabilistic models
- PAM and BLOSUM are famous substitution matrices

98.Optimal Algos

Introduction:

Finding the path whose total score is maximal will give the best sequence alignment

- Two methods
  - Local alignment
  - Global alignment

Global Alignment:

It is an alignment that essentially spans the full extents of input sequences

- Hence it covers the entire length of sequences involved
- The Needleman-Wunsch algorithm finds best global alignment between two sequences

Local Alignment:

- It only covers parts of the sequences to be aligned
- Smith-Waterman algorithm finds the best local alignment between two sequences

Dynamic Programming:

- Dynamic programming is used to find an optimal alignment of two sequences and its scores
- It is a method by which a larger problem may be solved by first solving smaller, partial versions of the problem
- Three steps in dynamic programming:
  - Initialization
  - Matrix filling (scoring)

Traceback (alignment

- **Initialization:** Create matrix with M+1 columns and N+1 rows where M and N correspond to the size of sequences to be aligned
- **Matrix filling:**Fill the matrix with highest possible score

MUHAMMAD IMRAN

- **Trace back:** Move from the last corner and follow the arrow

Conclusion:

- Dynamic programming is used to find an optimal alignment of two sequences and its scores

99.Needleman_wunch Algos

Introduction:

It performs global alignment on two sequences

- **The algorithm was developed by Saul B. Needleman and Christian D. Wunsch and published in 1970**

Basic idea is

- to build up the best alignment by using optimal alignments of smaller subsequences
- It was the first application of dynamic programming to compare biological sequences

Steps:

Three steps

1. Initialization
2. Matrix filling
3. Traceback

Creating the Matrix:

Initial matrix is created with M+1 columns and N+1 rows

- Where M and N correspond to the length of sequences

Initialization:

- The cell of first row and first column of the matrix is initially filled with zero
- Add gap penalty for each shift to the right

Initialization:

a. $F(0, 0) = 0$
b. $F(0, j) = -j \times d$
c. $F(i, 0) = -i \times d$

Matrix Fill:

- Move through the cells row by row, calculating the score for each cell
- Compute three scores:
  - A match score
  - Vertical gap score
  - Horizontal gap score
- The *match score* is the sum of the diagonal cell score and the score for a match

MUHAMMAD IMRAN

- The *horizontal gap score* is the sum of the cell to the left and the gap score
- The *vertical gap score* is computed analogously

Matrix filling: Filling-in partial alignments

For each $i = 1 \ldots\ldots M$

For each $j = 1 \ldots\ldots N$

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i, y_j), & \text{[case 1]} \\ F(i-1,j) - d, & \text{[case 2]} \\ F(i,j-1) - d & \text{[case 3]} \end{cases}$$

$$Ptr(i,j) = \begin{cases} DIAG, & \text{if [case 1]} \\ LEFT, & \text{if [case 2]} \\ UP, & \text{if [case 3]} \end{cases}$$

Traceback:

- The final step in the algorithm is the trace back for the best alignment
- Start at the bottom-right corner
- Follow where maximum value comes from
- **F(M, N) is the optimal score, and from Ptr(M, N) can trace back optimal alignment**

Scoring Scheme:

- Scoring scheme introduced can be user defined
- It contains specific scores for match and mismatch residues as well as gap

Conclusions:

- Needleman and Wunsh Algorithm performs global alignment on two sequences using a dynamic programming approach

100.Smith_waterman Algo

Introduction:

- Finds the best local alignment between two subsequences

Steps:

- Initialization
- Matrix filling
- Traceback or alignment

Algorithm

1. **Initialization:**
   a. $F(0, 0) = 0$
   b. $F(0, j) = 0$
   c. $F(i, 0) = 0$

Matrix filling: Filling-in partial alignments

For each $i = 1 \ldots\ldots M$

For each $j = 1 \ldots\ldots N$

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i, y_j), & \text{[case 1]} \\ F(i-1,j) + d, & \text{[case 2]} \\ F(i,j-1) + d, & \text{[case 3]} \\ 0 \end{cases}$$

$$Ptr(i,j) = \begin{cases} DIAG, & \text{if [case 1]} \\ LEFT, & \text{if [case 2]} \\ UP, & \text{if [case 3]} \end{cases}$$

Trace-back:

MUHAMMAD IMRAN

F(M, N) is the optimal score, and from Ptr(M, N) can trace back optimal alignment

Creating the Matrix:

- Initial matrix is created with M+1 columns and N+1 rows
  - Where M and N correspond to the length of sequences

Initialization:

- First row and first column of the matrix is filled with zero

		T	G	G	T	G
	0	0	0	0	0	0
A	0					
T	0					
C	0					
G	0					
T	0					

Matrix Filling:

		T
	0	0
A	0	0

- Add the match or mismatch scores diagonally
- Add gap penalties vertically and horizontally
- Replace the negative values by zero

		T	G
	0	0	0
A	0	0	0
T	0	5	3

		T	G	G
	0	0	0	0
A	0	0	0	0
T	0	5	3	1
C	0	3	2	0

Match=+5
Mismatch=-3
Gap penalty=-2

		T	G	G	T	G
	0	0	0	0	0	0
A	0	0	0	0	0	0
T	0	5	3	1	0	0
C	0	3	2	0	0	0
G	0	1	0	7	5	5
T	0	5	3	5	12	10

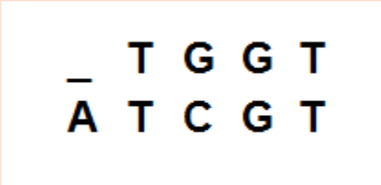
Traceback:

- Traceback starts with maximum value in the matrix and then go backwards

		T	G	G	T	G
	0	0	0	0	0	0
A	0	0	0	0	0	0
T	0	5	3	1	0	0
C	0	3	2	0	0	0
G	0	1	0	7	5	5
T	0	5	3	5	12	10

MUHAMMAD IMRAN

Alignment:

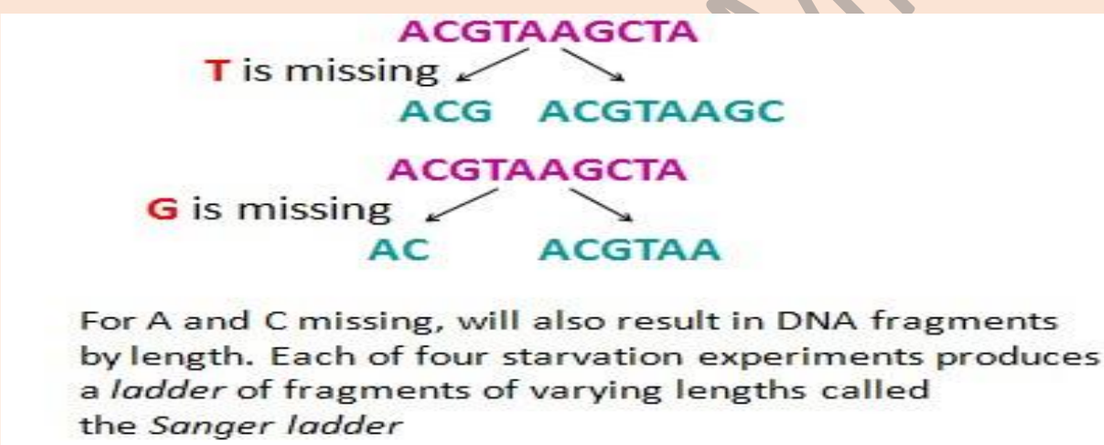


Conclusion:

- **Finds the best local alignment between two subsequences**

Topic 101-102 DNA Sequencing

Magazine cut into millions of pieces. Problem of fragment assembly in DNA sequencing Short 500 to 700 nucleotide sequences per experiment, Assembling entire genome reassembling the magazine Both problems are complicated by unavoidable experimental error. The data are frequently incomplete. DNA sequencing methods-Fred Sanger and Walter Gilbert. Cells make copies of DNA DNA fragments of different lengths- if one base is missing.



Sequencing of a 5386- nucleotide virus, DNA sequence data, Human Genome Project , 3 billionnucleotide sequence, DNA sequencing technology, Sequencing reads, Continuous genome, DNA reading process, DNA sequencing machines, Single DNA fragment.

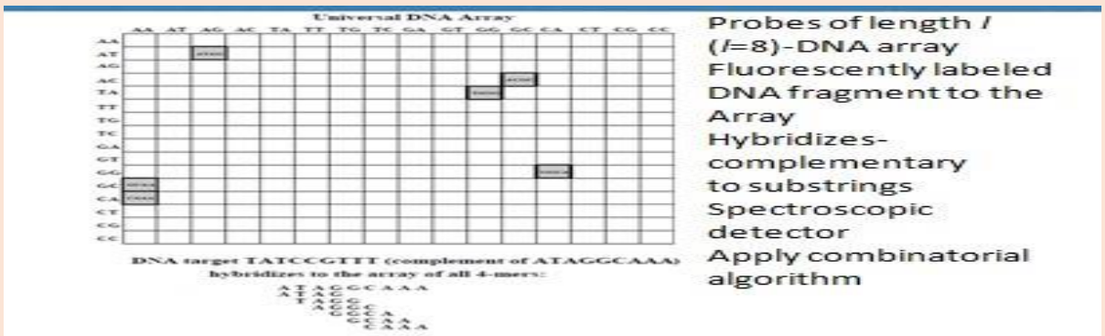
Shotgun sequencing

Sonicated, Inserts, Vector, Bacterial host, Cloning proce, DNA sequencing- inserts and computational

Topic-103 DNA Array

Human Genome Project, Sequencing ***by Hybridization,*** DNA array (DNA chip) , Probes, Hybridization, Weak chemical bond. DNA probes, Biochemical problem and the combinatorial problem ***Science***, SteveFodor and colleagues, Light-directed polyme synthesis –similarities to computer chip manufacturing.

In Fig. An array with all 4^l probes- $4 \cdot l$ separate reactions Affymetrix-64-kb DNA array in 1994 1-Mb or larger Probes- unknown target DNA-l-mer composition Universal DNA array contains all $4l$ probes of length l .



Topic-104 Sequencing by Hybridization

Unknown DNA sequence- length l , String s of length n, the l-mer composition is the multiset of $n - l + 1$ l-mers in s and is written *Spectrum*(s, l).

If $l = 3$ and s = **TATGGTGC**, then

Spectrum(s, l) = {**TAT, ATG, TGG, GGT, GTG, TGC**}

Lexicographic order, like **ATG,GGT,GTG, TAT, TGC, TGG**

Sequencing by Hybridization (SBH) Problem:

Reconstruct a string from its l-mer composition

Input: A set, S, representing all l-mers from an (unknown) string s

Output: String s such that *Spectrum(s, l) = S*

Topic-105- Fragment Assembly in DNA Sequencing

500- to 700-bp DNA reads Reconstruct the entire genomic DNA sequence Fragment assembly Error rate in DNA reads Sequencing errors, Assignments to read Major problem-repeats in DNA-

20% of human genome Alu sequence-million times Repeats occur at many scales Human T-cell receptor locus-Trypsinogen gene (4 kb).1 million Alu repeats (300 bp) and 200,000 LINE repeats (1000 bp) 3 billion-letter sequence 500-letter reads, large number of repeats.

106. Strategy for Sequencing

Strategy for Sequencing

Clone into BAC, Sequenced each one-minigenome, Simplifies the computational assembly problem, Human Genome project, Whole-genome assembly paradigm (James Weber and Gene Myers)Celera Genomics Mate-pair reads sequencing technique.

Inserts of length *L*- both ends are sequenced, Mates at distance *L*- length is larger than most repeats

Most fragment assembly algorithms consist of the following three steps

Overlap: Finding potentially overlapping reads

Layout: Finding the order of reads along DNA

Consensus: Deriving the DNA sequence from layout Best match between the suffix of one read and the prefix of another. Sequencing errors-Dynamic programming algorithm. constructing the layout is the hardest step in fragment assembly. Whole-genome shotgun sequencing. Large number of reads to ensure that experimental errors are reduced to minor noise.

MUHAMMAD IMRAN

Topic-107- Protein Sequencing and Identification

Routinely sequenced proteins-Frederick Sanger-Nobel prize, Computational problem ***Edman degradation reaction*** to chop off one terminal a.a, Sanger digested insulin with proteases
DNA sequencing "break-read the fragments-assemble"

Edman degradation reaction, 1960s protein sequencing machines were on the market DNA sequencing technology Protein-Obtaining reads-problem Assembly- easy

108. Computational Protein Sequencing

Computational Protein Sequencing

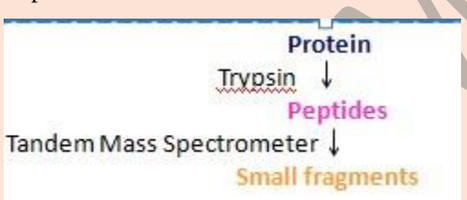
Proteins produced in cell Sequence of previously unknown proteins. Proteins -biological system and Range-Brain cells-Liver cells

De novo protein sequencing Protein identification Gedanken experiment Proteins constitute the DNA polymerase
Protein sequencing and identification , ***Spliceosome ,*** Matthias Mann and colleagues purified the spliceosome complex

109. Mass Spectrophotometry

Programmed cell death, Survival factors, Developing nematode DNA sequence data Nervous system- mutation in several genes, Mass spectrometry.

Mass spectrum of a peptide is a collection of masses of these fragments- Derive the sequence of a peptide given its mass spectrum. For an ideal fragmentation process the peptide sequencing problem is simple. The fragmentation process is not ideal, and mass spectrometers measure mass with some imprecision.



110.111 Peptide Sequencing Problem

Amass spectrometer breaks a peptide $p_1 p_2 \cdots p_n$

GPFNA GPFNA

G, GP, GPF, GPFN PFNA, FNA, NA, A

GPFNA into GP and FNA , lose some small parts of GP and FNA, fragments of a lower mass
GP might lose a water (H_2O), and the peptide FNA might lose an ammonia (NH_3). Mass of GP minus the mass of water (1 + 1 + 16 = 18 daltons) , and the mass of FNA minus the mass of ammonia (1 + 1 + 1 + 14 = 17 daltons). Two different ion types

Peptide fragmentation is characterized by a set of numbers $\Delta = \{\delta 1, \ldots, \delta k\}$ types of ions The set of ion types. A *δ-ion* of an N-terminal partial peptide P_i is a modification of P_i that has mass $m_i - \delta$, The most frequent N-terminal ions are called *b*-ions (ion b_i corresponds to P_i with $\delta = -1$) and the most frequent C-terminal ions are called y-ions (ion y_i corresponds to P_i^- with $\delta = 19$)

112.Protein identification via Database Search

De novo protein sequencing algorithms- invaluable for known, unknown proteins. Useful for complete spectra Spectra far from complete, De novo peptide sequencing algorithms. If we had access to a database of all proteins from a genome, then we would no longer need to consider all 20^l peptide sequences to interpret an MS/MS spectrum, but could instead limit our search to peptides present in this database . Database search "the back of a book", Experimental spectrum , Sequence of the experimental peptide, SEQUEST algorithm – John, Yates and colleagues.

Protein Identification Problem:

Find a protein from a database that best matches the experimental spectrum.

Input: A database of proteins, an experimental spectrum *S*, a set of ion types , and a parent mass *m*.
Output: A protein of mass m from the database with the best match to spectrum *S*

Topic-113 Modified Protein Identification Problem

SEQUEST Algorithm

Exhaustive search approach

Virtual database

Potential modifications

Combinatorial problem

Modified Protein Identification Problem:

Find a peptide from the database that best matches the experimental spectrum with up to k modifications.

Input: A database of proteins, an experimental spectrum *S*, a set of ion types Δ, a parent mass *m*, and a parameter *k* capping the number of modifications

Output: A protein of mass *m* with the best match to spectrum *S* that is at most *k* modifications away from an entry in the database

Modified Protein Identification problem P_1 and P_2- S_1 and S_2, Notion of spectral similarity
Shared peaks count, Limitations in detecting similarities by database search.

Topic 114-Protein Structures

Background

Complex protein structures enable proteins to perform complex functions. We know over a million protein sequences but only about 100,000 protein structures.

Why only 100,000 proteins for over million protein sequences

Estimating exact protein structures is very difficult. Its difficult to crystallize proteins. Even if we manage to get protein's X-Ray, to reconstruct the structure is extremely complex

Introduction

What if we could somehow predict protein structures?

MUHAMMAD IMRAN

- Since we know so many sequences, they can be used for predicting protein structures. This indeed is possible and helpful.

The Basic Idea

1. Amino acids determine the protein structure
2. We have a large protein sequence dataset (uniprot)

Hence, we can fold protein sequences and predict their structures!

Why predict and why not exact solutions?

A deterministic solution of protein folding is a major unsolved problem in molecular biology! Proteins fold spontaneously or with the help of enzymes or chaperones.

To predict we must first learn!

To computationally predict protein structures, we need to copy or mimic the natural folding! What are the steps in protein folding and structure formation?

To fold we must learn the steps

Step 1: "Collapse"- leading to burial of hydrophobic AA's

Step 2: Fluid globule - helices & sheets form, but are unorganized

Step 3: Compaction, and rearrangement of 2' structures

Conclusion

- Protein structure prediction involves learning how the amino acids in primary sequence fold.
  - Using this information, upon getting a protein sequence, we can try to predict how it folds!

115. Predicting Secondary Structures

Background

Since the first step in protein folding is the formation of secondary structures, we must evaluate which amino acids in the primary sequence prefer which secondary structures?

- By looking at the structures in PDB, we know that Alanine mostly found in Alpha Helices. So if we have several Alanines in the sequence, then we can anticipate that a helix may be formed by them

Introduction

What if we survey the entire PDB and check the presence of each amino in each type of secondary structure

- **If we know which amino acid is found in which specific secondary structure, then we can use it for prediction!**

Amino Acid	P_α	P_β	P_t
Glu	1.51	0.37	0.74
Met	1.45	1.05	0.60
Ala	1.42	0.83	0.66
Val	1.06	1.70	0.50
Ile	1.08	1.60	0.50
Tyr	0.69	1.47	1.14
Pro	0.57	0.55	1.52
Gly	0.57	0.75	1.56

Conclusion

MUHAMMAD IMRAN

- Several algorithms have been designed to predict 2’ given an amino acid sequence
- The first such algorithm was the Chou-Fasman Algorithm!
- We will see it in the upcoming modules!

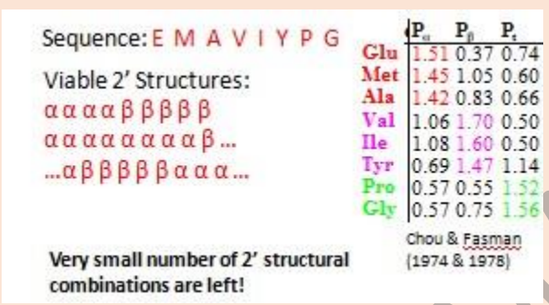
116. 2’ Structures in Chou Fasman Algorithm

Background

- For a primary sequence, and a tentative 2’ structure,propensity table can help us compute the overall propensity
- Product of propensity values is computed for overall propensity for each 2’ structure

Introduction

- An important point to note here is that 2’ structures are formed due to hydrogen bonding between amino acids
- So, we need to consider the neighboring amino acids as well!



	P_α	P_β	P_t
Glu	1.51	0.37	0.74
Met	1.45	1.05	0.60
Ala	1.42	0.83	0.66
Val	1.06	1.70	0.50
Ile	1.08	1.60	0.50
Tyr	0.69	1.47	1.14
Pro	0.57	0.55	1.52
Gly	0.57	0.75	1.56

Conclusion

- You only need to compute propensities for a small number 2’ structures
- The highest net propensity will be the most probably secondary structure that will be formed!

117.121 Chou Fasman Algorithm -

Only a small number of combinations of secondary structures are possible due to their individual properties. Such as 4 amino acids are needed to start an Alpha Helix and 5 amino acids for Beta Sheet **Note that besides the alpha helix and beta sheets, LOOPS are an other secondary structure.**

How can loops be integrated into predicting 2’ structures?

- Loops are small ~ 3-4 amino acids

MUHAMMAD IMRAN

1. Scan through the sequence : E M A V I Y P G

2. Identify sequence regions where:
   - 4 out of 6 contiguous residues give a P(α) > 1.0
   - That region is declared as alpha-helix
   - Extend helix to both sides until 4 out of 6 contiguous residues give a P(α) < 1.0
   - That is declared end of the helix

Conclusion

- For Alpha Helices, 4 contiguous amino acids are required
- Their Alpha-Helix propensity should be more than 1.0
- Once this propensity falls below 1.0, Alpha-Helix stops

Chou Fasman Algorithm - II

Background

Alpha Helices are formed from 4 contiguous amino acids having an Alpha-Helix propensity over 1.0. The Alpha-Helix stops if this propensity falls below 1.0! Introduction

- Once Alpha Helices are constructed, and concluded, the remaining amino acids can be evaluated for Beta sheets and turns etc
- Let's see how Beta sheets are evaluated using Chou Fasman Algorithm

1. Compute P(β) for contiguous regions of 5 Amino Acids

2. From these regions, identify regions where:
   - 5 contiguous residues have P(α) > P(β)
   - That region is finalized as alpha-helix

Repeat this step for the full amino acid sequence to finalize all possible alpha helical regions in the sequence.

Conclusion

- Alpha Helices can be finalized if their propensity is higher than the propensity for Beta Sheets in regions of 5 amino acids
- For those regions where that is not the case, further evaluation is required

Chou Fasman Algorithm - III

Alpha Helices were finalized if their propensity was higher than the propensity for Beta Sheets in regions of 5 amino acids**.** For those regions where that are not the case, what should be done?

We can evaluate such regions for Beta Sheets.

Let us see step by stop how to find a beta sheet and how to differentiate them from alpha helices

MUHAMMAD IMRAN

Scan the sequence to identify regions where:

- 3 out of 5 amino acids have P(β) > 1.0
- That region is declared as beta sheet
- Extend beta sheet to both sides until 4 contiguous residues average P(β) < 1.0
- That is declared end of the beta sheet
- Those regions are finalized as beta-sheets which have average P(β) > 1.05 and the average P(β) > P(α) for that region.

Regions where overlapping alpha-helices and beta-sheets occur are declared helices if

- the average P(a-helix) > P(b-sheet) for that region

Else, a beta sheet is declared if

- average P(b-sheet) > P(a-helix) for that region

Conclusion

- Using the strategy of higher propensity, alpha helices and beta sheets can be completely resolved
- Assignments for each beta sheet and alpha helix can be finalized
- But what about the loops?

Chou Fasman Algorithm - IV

Background

- After computing the propensity of alpha helices and beta sheets, we need to settle for loops
- Let's see how can we find out the loops using Chou Fasman Algorithm

- For any *jth* residue in sequence, we calculate f (Total) = f(j) f(j+1) f(j+2) f(j+3) (tetrapeptide)
- If
  1. f(Total) > 0.000075
  2. the average value for P(turn) > 1.00 in the tetrapeptide
  3. the averages for the tetrapeptide are such P(a-helix) < P(turn) > P(b-sheet),

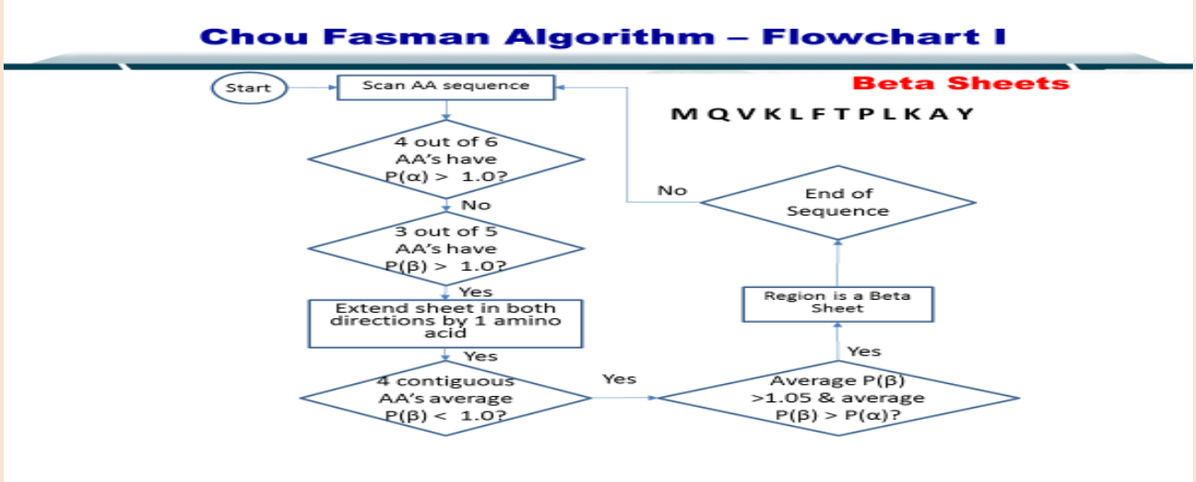
Conclusion

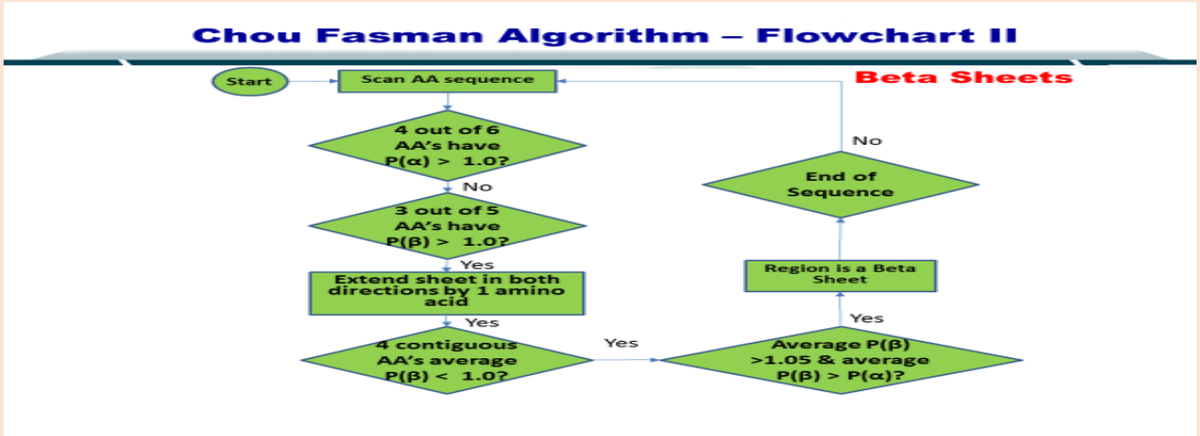
- Chou Fasman Algorithm helps predict Alpha Helices, Beta Sheets and Turns
- The algorithm is based on statistical occurrence of Amino Acids in known structures

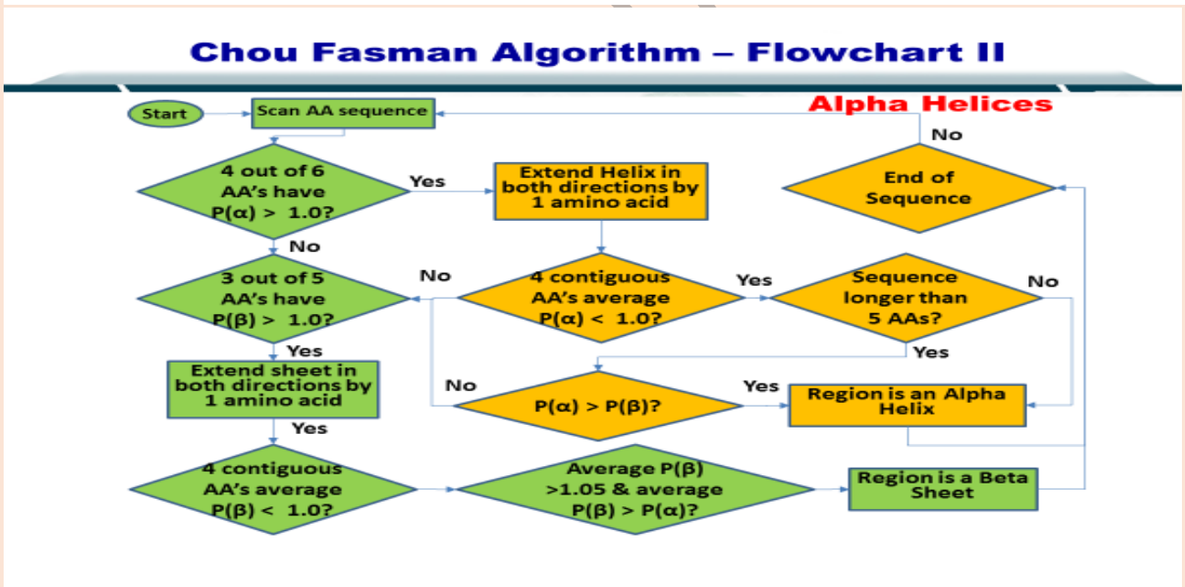
122.125. Chou Fasman Algorithm – Flowchart I, II & III.

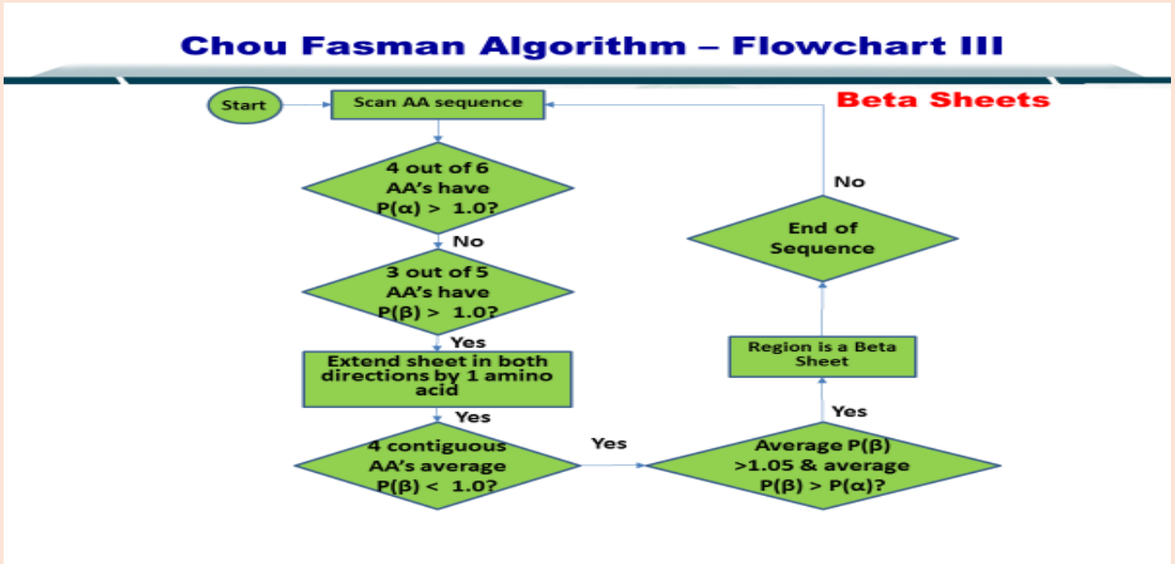
- Chou Fasman Algorithm helps predict secondary structures such as Alpha Helices, Beta Sheets and Turns. Step by step flowchart of the entire algorithm. Beta sheets can be predicted from primary amino acid sequences

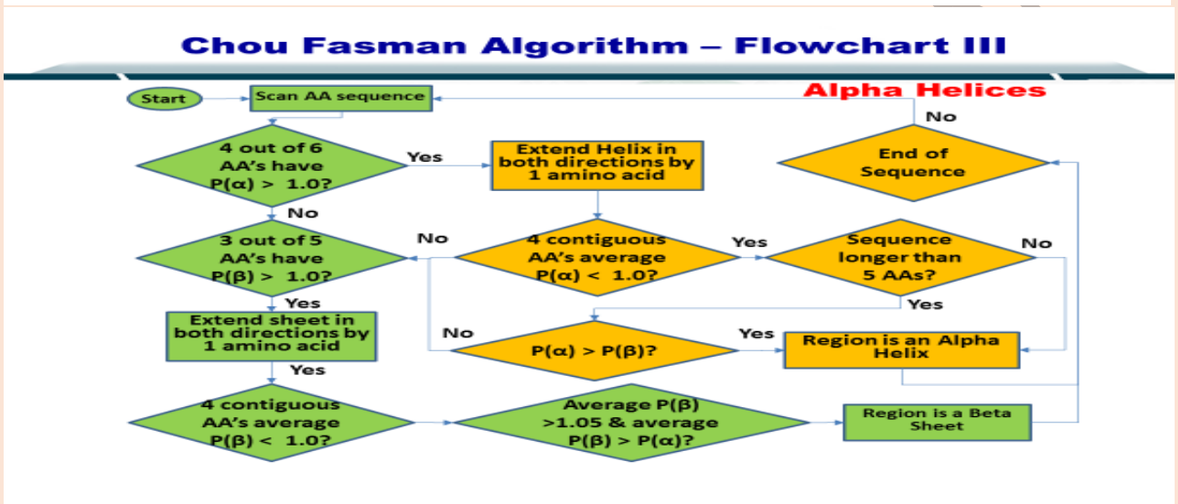
MUHAMMAD IMRAN

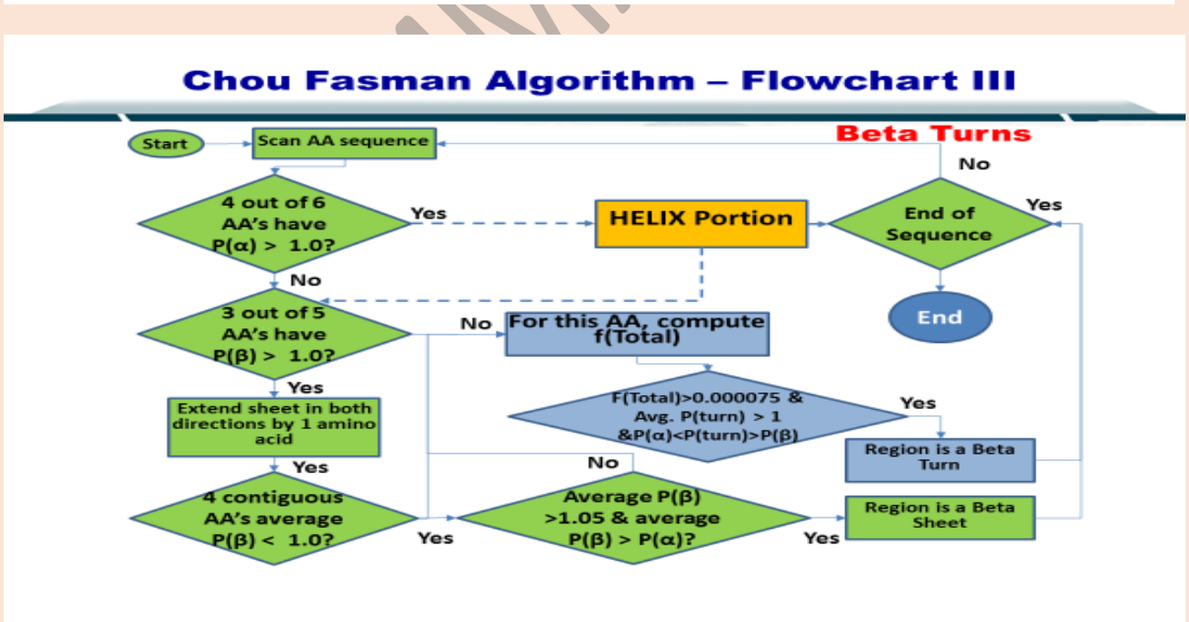

Chou Fasman Algorithm – Flowchart I
Beta Sheets
M Q V K L F T P L K A Y
Start
Scan AA sequence
4 out of 6 AA's have P(α) > 1.0?
No
3 out of 5 AA's have P(β) > 1.0?
Yes
Extend sheet in both directions by 1 amino acid
Yes
4 contiguous AA's average P(β) < 1.0?
Yes
Average P(β) >1.05 & average P(β) > P(α)?
Yes
Region is a Beta Sheet
End of Sequence
No


Chou Fasman Algorithm – Flowchart II
Beta Sheets
Start
Scan AA sequence
4 out of 6 AA's have P(α) > 1.0?
No
3 out of 5 AA's have P(β) > 1.0?
Yes
Extend sheet in both directions by 1 amino acid
Yes
4 contiguous AA's average P(β) < 1.0?
Yes
Average P(β) >1.05 & average P(β) > P(α)?
Yes
Region is a Beta Sheet
End of Sequence
No


Chou Fasman Algorithm – Flowchart II
Alpha Helices
Start
Scan AA sequence
4 out of 6 AA's have P(α) > 1.0?
Yes
Extend Helix in both directions by 1 amino acid
No
End of Sequence
3 out of 5 AA's have P(β) > 1.0?
No
4 contiguous AA's average P(α) < 1.0?
Yes
Sequence longer than 5 AAs?
No
Yes
Extend sheet in both directions by 1 amino acid
Yes
No
P(α) > P(β)?
Yes
Region is an Alpha Helix
4 contiguous AA's average P(β) < 1.0?
Average P(β) >1.05 & average P(β) > P(α)?
Region is a Beta Sheet







- Chou Fasman Algorithm helps predict secondary structures from amino acid sequences. Step by step flowchart of the algorithm for extracting Alpha Helices
- Alpha helices, beta sheets and turns can be predicted using Chou Fasman Algorithm. This algorithm is based on statistical analysis of amino acid occurrences in proteins.

Chou Fasman Algorithm – Improvements

- Secondary structure propensity values of alpha helix, beta sheet and turns should be recalculated with the latest protein data sets.

Special consideration for:

- Nucleation regions
- Membrane proteins
- Hydrophobic domains
- Consider variable coil and loop sizes besides the from tetra peptide turns
- Consider local protein folding environments
- Solvent accessibility of residues
- Protein structural class
- Protein's organism

Conclusion

Chou Fasman can be improved to better predict secondary structures by incorporating biochemical factors and updated statistics!

126. Summary of Visualization, Classification & Prediction

A. Why do we need to visualize proteins?
B. Which atoms are used to reconstruct proteins?
C. Where are the positions of these atoms stored?

Structure Classification

A. What is the relationship between protein structure and function?
B. What is the need to classify proteins
C. Hierarchy of classification

Structure Prediction

A. Why structure of proteins are important?
B. Why are so few structures reported till date?
C. Benefits of predicting structures

Conclusion

Structure visualization, classification and prediction equip us to perform functional evaluation of proteins! This is important for understanding disease and designing drugs for treating them

Topic-127- Introduction to Homology Modelling

Proteins are 3D molecules with their own unique structures. Protein structure is reflective of the protein function. Protein structure includes 1', 2', 3' and 4' structures. 1' structure of proteins is the sequence of proteins and can be obtained by mass spectrometry. 2' structures formed by proteins are the helices, beta sheets, loops and coils. 3' structure of proteins is the combination of 2' structures such that the overall protein structure is formed. 4' protein structure is formed when two or more proteins complex together. **X**-Ray Crystallography and NMR Spectroscopy are used to find the structures of proteins. However, these methods are difficult and expensive. Solution: Prediction of structures. Protein sequence gives rise to its structure**.** If another protein which has a similar sequence also has its structure known, the structure of an unknown protein can be predicted based on that similar protein **.** So, it is then possible to identify unknown protein structures by just examining the homologous protein sequences.

Conclusions

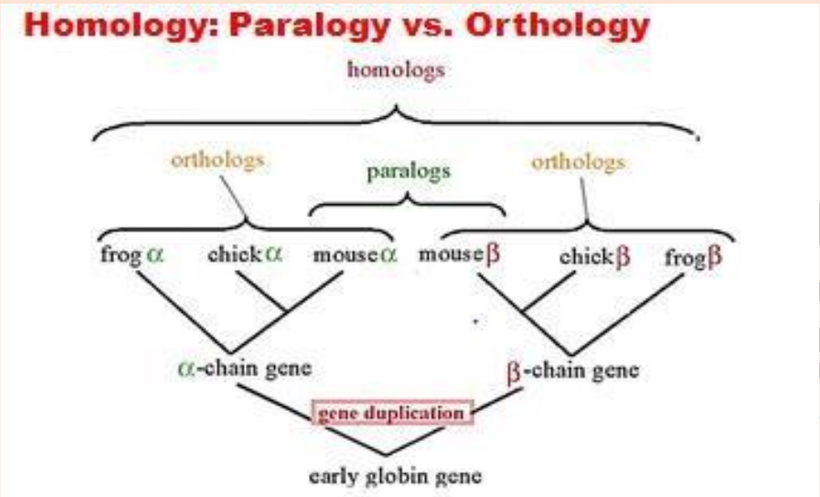
MUHAMMAD IMRAN

- Sequence Identity
- Alignment Length

Which combination of identity and alignment length is suitable for best for structure prediction?

128.Homology, Paralogy and Orthology

In homology modelling, proteins with similar 1' sequences are considered. Given that one of them has its 3' structure known, then the 3' structure of other protein can be predicted



Conclusions

- Good sequence alignment and identity ensures that homology modelling will give accurate results

129. Workflow of Structural Modelling

Homology modelling is used to predict structures of proteins having high sequence similarity with other proteins with known structures! **Overall, there are three different strategies for structure prediction**

1. Homology Modelling
2. Threading/Fold Recognition
3. Ab Initio Modelling

130.135 Seven Steps to Homology Modelling – I

Protein structure can be predicted by 3 methods:

1. Homology Modelling

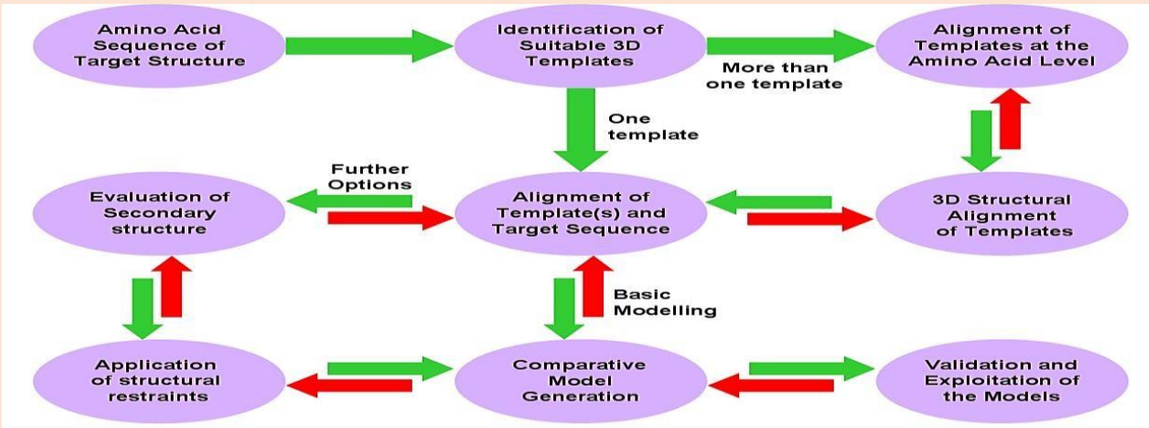
2. Fold Recognition / Threading

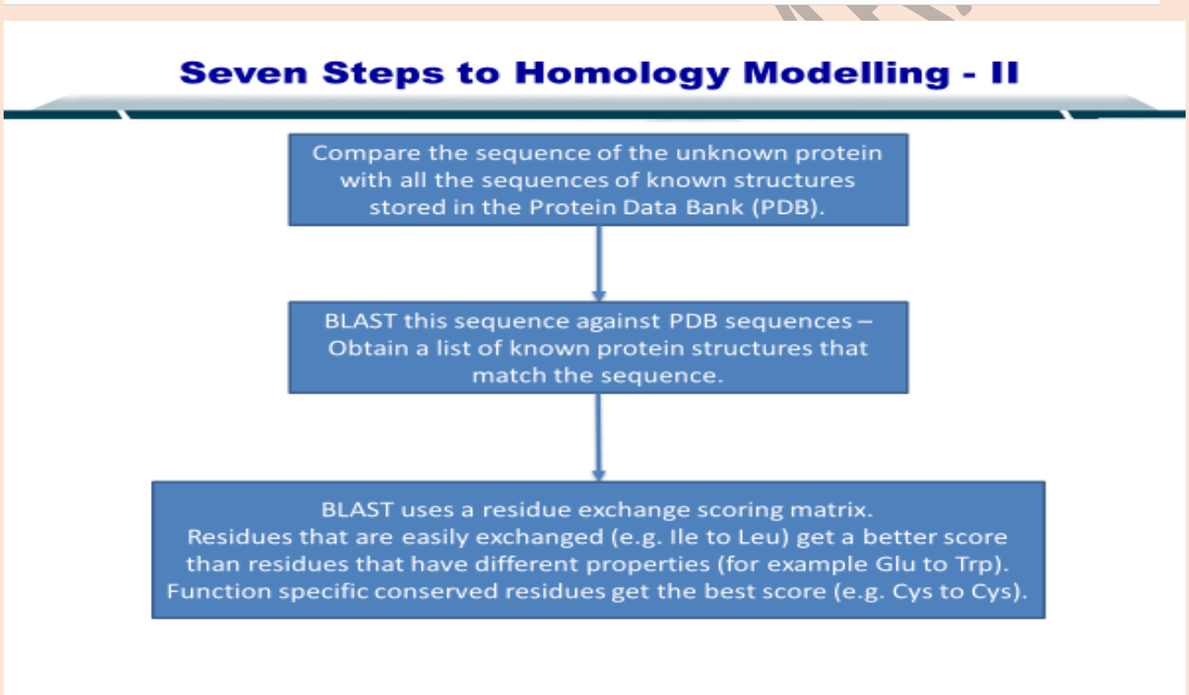
3. Ab Initio Modelling

Let's start by looking at Homology Modelling. There are seven salient steps in any Homology Modelling pipeline. Definition of Template (known) & Target (unknown). Homology modeling of the target structure can be done as follows:

MUHAMMAD IMRAN

1. Template recognition and initial alignment
2. Alignment correction
3. Backbone generation
4. Loop modeling
5. Side-chain modeling
6. Model optimization
7. Model validation





Seven Steps to Homology Modelling - II

BLAST will provide a list of possible templates for the unknown structure
To make the best initial alignment, BLAST uses an alignment-matrix based on the residue exchange matrix and adds extra penalties for opening and extension of a gap between residues.

The target-sequence is sent to a BLAST server, which searches the PDB to obtain a list of possible templates and their alignments.
The best hit has to be chosen, which is not necessarily the first one.

Seven Steps to Homology Modelling - III

Fine tune and adjust the BLAST alignments

Example: Ala -> Glu is possible but unlikely in a hydrophobic core, so these residues should not be aligned.

Use MSA tools (e.g. ClustalW), to find the residues and properties that need to be conserved.

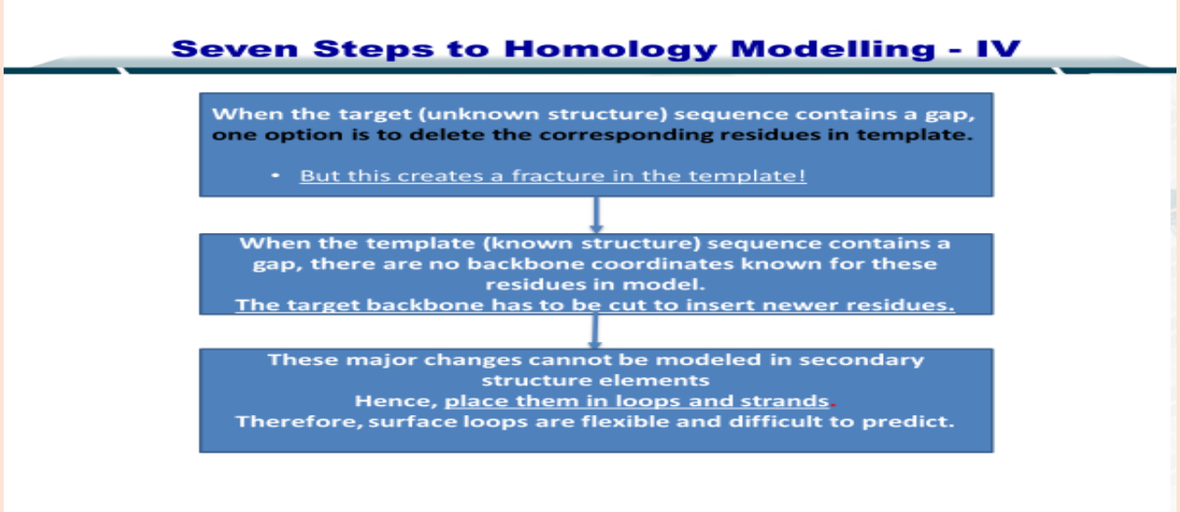
Examine the template structure to check which residues are in the core hence less likely to change than the residues at the outside.

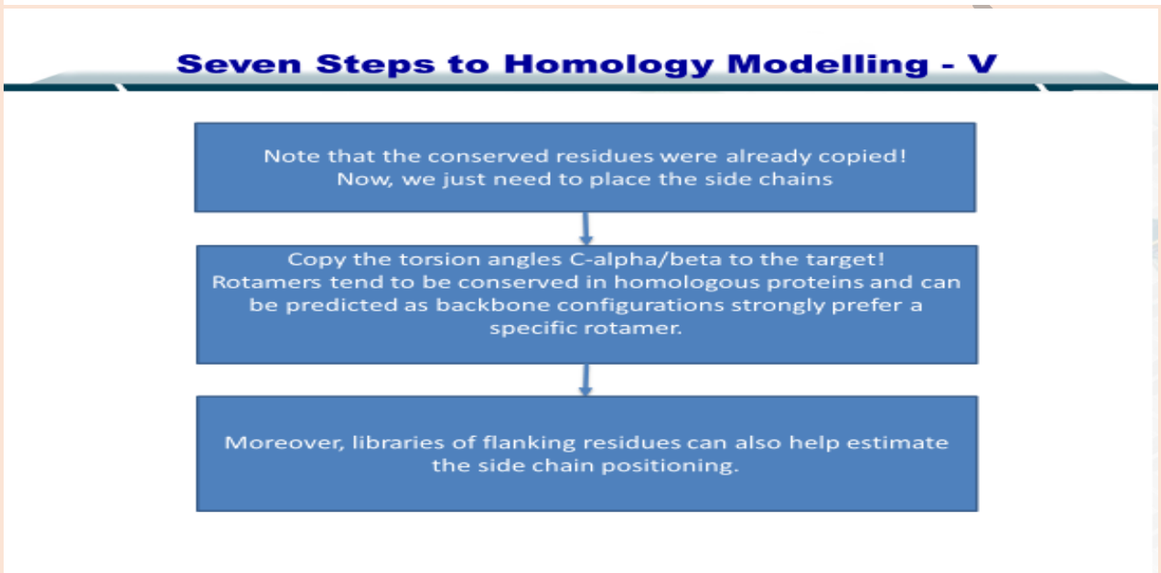
Seven Steps to Homology Modelling - III

Insertions and deletions can be made in those parts of the sequence which are highly variable .

Note: MSA can be helpful to find these places.

Gaps have to be shifted around until they are as small as possible.





So, Homology modelling works in seven steps. It is a repetitive process

136.MODELLER for Homology Modelling

- Template recognition and initial alignment
- Alignment correction
- Backbone generation
- Loop modeling
- Side-chain modeling
- Model optimization
- Model validation

Modeller is a software for homology modelling

salilab.org/modeller

Inputs:
Python script file, Sequence alignment & Template (PDB)

.log : log output from the run.

.B* : model generated in the PDB format.

.D* : progress of optimisation.

.V* : violation profile.

.ini : initial model that is generated.

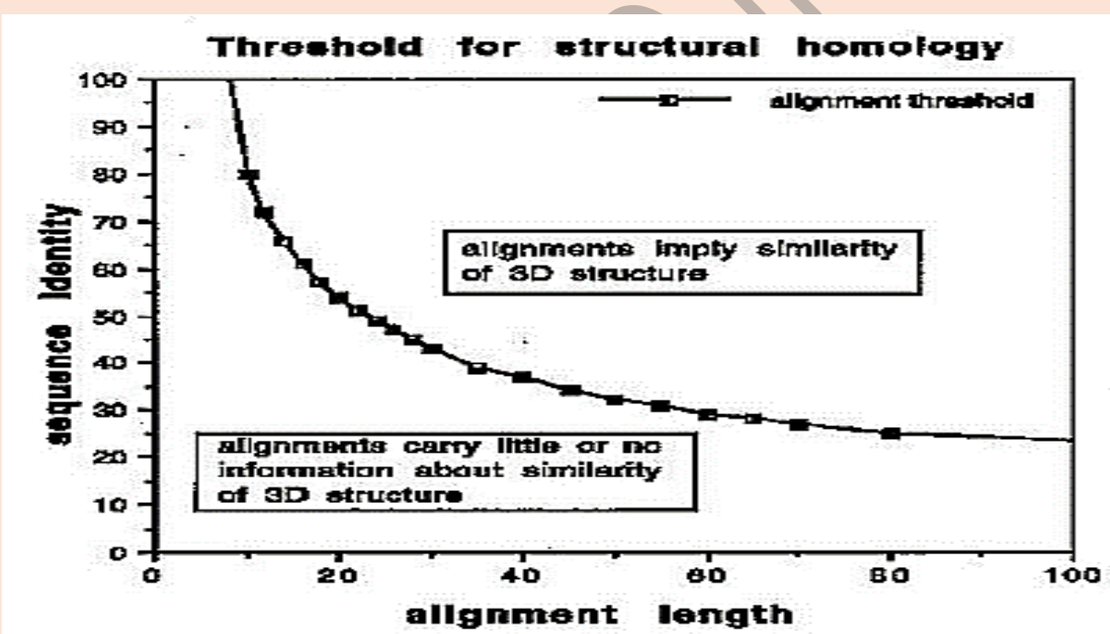
.rsr : restraints in user format.

.sch : schedule file for the optimisation process.

Homology modelling helps predict protein structures by using prior structural informationSeveral tools are available to perform homology modelling in a programmatic or automated way!

137.139.Fold Recognition – Threading

When should we use Fold Recognition?



Introduction

- **A protein fold is defined by the way the secondary structure elements of the structure are arranged relative to each other in space.**
- **Common folds include 4-helix bundle and the TIM barrel.**
- **5,000 stable folds in nature**
- **Fold recognition: Finding the best fit of a sequence to a set of candidate folds**

Fold recognition is also called Threading. Technique for predicting protein structures. Employed when homology modelling cannot predict quality structures. A protein fold is defined by the way the secondary structure elements of the structure are arranged relative to each other in space. Common folds include 4-helix bundle and the TIM barrel. 5,000 stable folds in nature. Fold recognition: Finding the best fit of a sequence to a set of candidate folds.

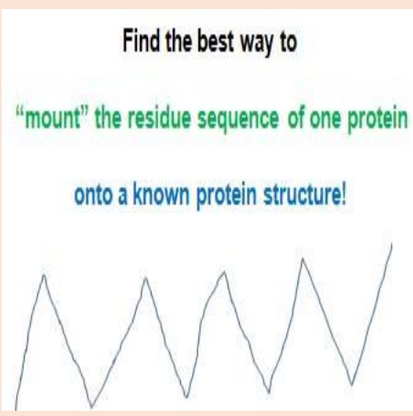
Fold recognition or Threading is a technique for predicting protein structures. It is useful in cases where homology modelling fails to predict quality structures

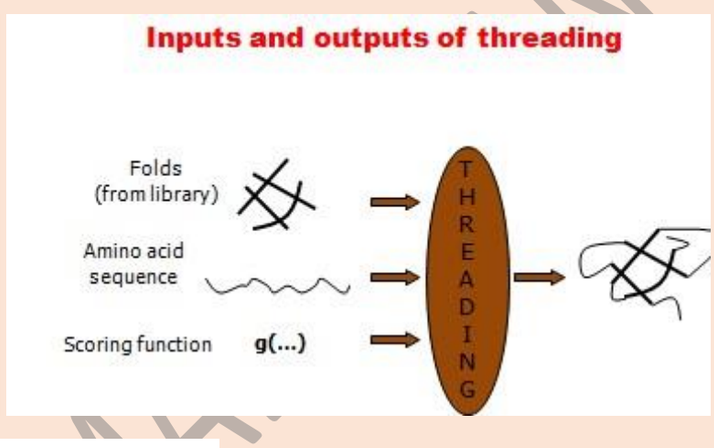
The process of threading

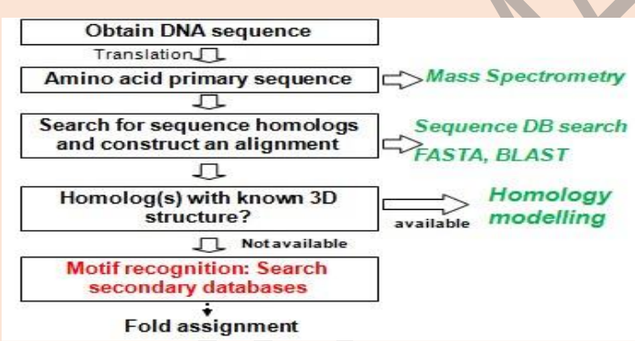
- In the process of "Threading", we mount an amino acid sequence on to the backbone of template structures in a folds library
- Each step is "drag" along the sequence (MQVKLFTY...) through each location of each template fold
- Then, for each fold, we must compute the fitness of sequence matching that fold!

-

Threading involves "passing" the amino acid sequence through each fold in the database. The best match is computed using a scoring function. Combinations of secondary structures come together to form the best prediction. Scoring typically involves using a Z-Score function based on energy of a molecule.







140 Online Tools for Threading – iTasser

iTASSER helps thread amino acid sequences on fold and secondary structure databases. It also helps predict function of structures output.

Iterative threading assembly refinement (I-TASSER) server

- Software for automated protein structure & function prediction based on the sequence-to-structure-to-function.
- Steps:
  - Starts from amino acid sequence
  - i-TASSER first generates 3D atomic models from multiple threading alignments and iterative structural assembly simulations.
  - The function of the protein is then inferred by structurally matching the 3D models with other known proteins.
  - Outputs full-length secondary & tertiary structures and functional annotations on ligand-binding sites
  - An estimate of accuracy of the predictions is provided based on the confidence score of the modeling

141.Advantages and Disadvantages of Threading

Fold recognition or Threading is a technique for predicting protein structures. It is useful in cases where homology modelling fails to predict quality structures.

Advantages

Threading helps predict secondary structures of proteins towards tertiary structure prediction. For the "Twilight Zone" with low alignment quality and identity, threading is use.

Disadvantages

Novel proteins cannot be predicted using threading. Fewer than 30% of the predicted first hits are true remote homologues. Validation of each result is necessary.

142. 3D-1D Bowie Algorithm

Homology employed high alignment scores. Threading worked by creating combinations of primary sequences and corresponding secondary structures. Proposed **by Bowie et al in 1991**. It converts 3D structure into a 1-D string profile for each structure in the fold library. Align the target sequence to these profiles. 3D-1D methods convert structure and environment information into "profiles". Score for each amino acid is computed for each profile. **Inputs and outputs of 3D-1D**

- Identify amino acids based on: protein core, side chain positioning, solubility etc. (6 in all)
- Part of secondary structure including □-helix, □-sheet etc (3 in all)
- Total of 3 x 6 = 18 distinct states
- $P_{a:j}$= prob. of finding amino acid (a) in environment (j)
- P_a=probability of finding (a) anywhere
- Maximize sum of scores for the fold:

$$s_{aj} = \log\left(\frac{P_{a:j}}{P_a}\right)$$

143. Introduction to Ab Initio Modelling

Ab initio methods have Anfinsen's thermodynamic hypothesis at the center. These methods attempt to identify the structure with minimum free energy. Ab initio methods rely on computing the energies of folded proteins. The protein structures with the lowest energy are deemed as plausible predictions.

Need for Ab Initio Modelling

- Applicable to any sequence
- Not very accurate biologically
- Accuracy and applicability are limited by our understanding of the protein folding problem

Limitation

- Computationally expensive

MUHAMMAD IMRAN

- Suitable for proteins with less than 100 residues

144.Rationale of Ab Initio Modelling

Ab initio methods rely on computing the energies of folded proteins**.** The protein structures with the lowest energy are declared as plausible predictions

Rationale

Sometimes it so happens that even slightly homologous proteins may not be available. This renders homology modelling and threading/fold recognition as futile . Also, newer protein structures continue to be discovered every day. These could not have been identified by methods which only rely on matching with available structures. Lastly, homology / fold recognition predict protein structures without computing fundamental physical/chemical properties of the mechanisms and driving forces in structure formation. Ab initio methods, in contrast, base their predictions on physical models for these mechanisms. Energy released during the folding process is computed for predicting structure.

145.Strategies for Ab Initio Modelling

Ab initio methods base their predictions on physical models of folding mechanisms. Stabilization is measured by energy released during the folding process. Start with an energy function. Fold structures in order to obtain the most stable structure. This structure will have the minimum energy.

Energy Optimization in Ab Initio Modelling

1. Start with a rough initial model.
2. Define an energy function mapping structures to energy values. We have to minimize this later!!
3. Solve the computational problem of finding the global minimum.

Simulation of the Folding Process

1. Build an accurate initial model (including energy and forces).
2. Accurately simulate the dynamics of the protein folding process.
3. The native structure will steadily emerge.

146.Energy States of Folded Proteins

Ab initio methods predict protein structures by folding proteins based on each constituent atom's volume, charge, mass etc. The protein structure reporting lowest energy is selected to be the optimal structure.

Energies of Bonded Atoms vs. Nonbonded Atoms

$$V(R) = E_{bonded} + E_{non-bonded}$$

MUHAMMAD IMRAN

147.Local versus Global Minima

The protein structure reporting lowest energy is selected to be the optimal structure

Best Case Energy Function

- Clear energy minimum in the native structure
- Viable path towards this minimum
- Global optimization finds the most stable structure

Optimal Energy Function

- Easier to design and compute
- Native structure not always at the global minimum
- No clear way of choosing among alternative structures that are generated

148. Pros and Cons of Ab Initio Modelling

Native structure not always at the global minimum**.** No clear way of choosing among alternative structures that are generated **Advantages**

- Ab Initio methods can fold any target sequence using only physical atomic properties
- Predictions are mostly accurate and correctly describe the natural folding process

Disadvantages

- Ab initio methods are the very difficult to design (energy function)
- These methods are slow due to the huge possibilities

149.151 Summary of Structural Modelling

Strategies for Structural Modelling

- Homology Modelling
- Fold Recognition
- Ab Initio Modelling

Homology modeling of the target structure can be done as follows:

1. Template recognition and initial alignment
2. Alignment correction
3. Backbone generation
4. Loop modeling
5. Side-chain modeling
6. Model optimization

MUHAMMAD IMRAN

7. Model validation

Energy Optimization in Ab Initio Modelling

1. Start with a rough initial model.
2. Define an energy function mapping structures to energy values. We have to minimize this later!!
3. Solve the computational problem of finding the global minimum.

Simulation of the Folding Process

1. Build an accurate initial model (including energy and forces).
2. Accurately simulate the dynamics of the protein folding process.
3. The native structure will steadily emerge.

Conclusion

- Homology modelling is performed in cases of high identity and alignment score
- For the "Twilight zone", other strategies are employed
- For low identity and alignment scores, a "Twilight zone" for structure prediction exists
- Fold recognition / threading is useful in such cases
- For cases where even the fold libraries do not give any high scoring matches, Ab Initio strategies can help model the structure
- However, this is a complex and computationally expensive process

152.Review of Sequence Analysis

Important Concepts

How do we sequence:

Genomes, Proteomes

How do we compare sequences:

Pair-wise Sequence Alignment, Multiple Sequence Alignment

Types of Alignments:

Global Alignment , (Needle Wunsch), Local Alignment, (Smith Waterman)

- **Advanced Tools:**

Fast Alignment (FASTA), Basic Local Alignment Search Tool,

(BLAST) **Databases:** GenBank, UniProt **Online Portals:**

Ensemble, Expasy, UniProtKB

MUHAMMAD IMRAN

153.Review of Phylogenetics

- **Important Concepts**

Molecular Evolution

Insertions, Deletions, Substitutions

Phylogenetic Trees

Scaled Trees, Unscaled Trees

Phylogenetic Trees

Rooted Trees, Unrooted Trees

UPGMA: Unweighted Pair – Group Method using arithmetic Averages

Two sequences with with the shortest evolutionary distance between them are considered

These sequences will be the last to diverge, and represented by the most recent internal node.

Clustering Vs. Non-clustering Methods:

UPGMA is a clustering method

Maximum Parsimony etc are non-clustering methods (not included in this course).

154.Review of Protein Sequencing

Important Concepts

Techniques of protein sequencing, Edman Degradation, Mass Spectrometry, Protein Ionization, Mass Analysis, Protein Fragmentation, MS1, MS2, Estimating and scoring whole protein mass, Extracting & Scoring Peptide Sequence Tags, Searching Post-translational Modifications

Composite Scoring Schemes

Online tools:

Mascot, Sequest, Prosight PC

155 Review of RNA Structure Prediction

156.Review of Protein Structures

Important Concepts

Protein Structures are generally of four types:

Primary, Secondary, Tertiary, Quaternary

Techniques for determining protein structures

X-Ray Crystallography, NMR Spectroscopy

Types of Protein Secondary Structures

Helices, Beta Sheets, Coils, Loops

- **Foundation of structure prediction algorithms**
  Propensities of certain amino acids to form specific secondary structures

MUHAMMAD IMRAN

- **Algorithm for predicting protein structures**

  Chou Fasman Algorithm

- Protein Structure Database – PDB, Online tools for predicting structures by using proteins sequences

157.Review of Homology Modelling

Four Strata of Protein Structures

Primary, Secondary, Tertiary, Quaternary

- **Justification for homology modelling**

  Number of known protein sequences is much larger as compared to known proteins structures

Three Strategies for Structure Prediction

Homology Modelling, Fold Recognition, Ab Initio Modelling, Protein Structure Database – PDB, Online tools for predicting structures such as MODELLER and iTASSER

158.Conclusions from this Course

Definition of Bioinformatics, Need for Bioinformatics, Areas within Bioinformatics, Bioinformatics as an interdisciplinary area, Need to store, process and analyze biological data, Requirement of newer faster algorithms **Specific areas focused were:**

Comparing sequences, Comparing structures, Predicting structures

We looked at:

Algorithms, Databases, Online Tools for each topic.

- We studied the basic algorithms for each topic, With evolution and growth of Bioinformatics, newer and better algorithms are now also available!

159.Advanced Follow-up Courses

We looked into the foundations of Bioinformatics, However, each topic that was studied has a undergone a lot of development.

- For advanced study in Genomics, you may take "Computational Genomics" course **Topics:**

Genome Assembly, Gene Finding, Annotation, GWAS etc

- For advanced study in Proteomics, you may take "Computational Proteomics" course. **Topics:** Protein Sequencing, PTM search, Structure Modelling and PPI studies

- **For advanced study in Integrative Biology, you may take "Systems Biology" course. Topics:** Metabolomics, Transcriptomics, Network Biology etc

Also, now there are cutting edge courses on:

Nano-Bio-IT, Computational Drug Design, Personalized Medicine

MUHAMMAD IMRAN

160.Careers in Bioinformatics

Pakistan as an infrastructure-limited country. The onset of digital revolution. Emergence of data as the most precious commodity, globally. Specifically, health data as a key commodity of the future. Health and disease as the primordial challenge of mankind

- **Unique opportunity for us in Pakistan**

Bioinformatics requires two things

1. Smart mind

2. Internet connected computer

One man company

You can take public databases and design drugs. One man vs. Roche?

BIGDATA

You can make a startup company which manages and process health BIGDATA. All it needs is basic software development skills coupled with Bioinformatics

The next disruption

The next Google, Facebook and Uber is going to emerge from Health and Bioinformatics. Pharmaceutical companies are investing into bioinformatics human resource development

Jobs Market

Pharmaceutical Giants**,** Research Centers & Universities**,** Hospital & Diagnostic IT departments **,** Your own startup company

Topic-161 RNA Structure

Outline

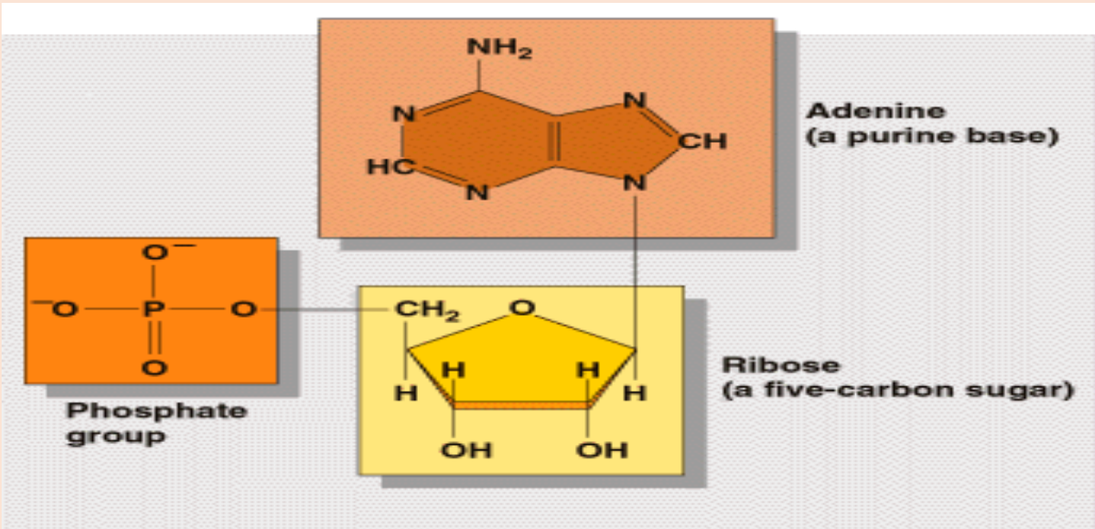
- **RNA folding**
- **Dynamic programming for RNA secondary structure prediction**
- **Covariance model for RNA structure prediction**

Base Pairing

- **RNA bases A,C,G,U**
- **Bases can only pair with one other base**
- **Canonical Base Pairs**
    - **A-U**
    - **G-C**
- **"wobble" pairing**
    - **G-U**
    - **I-U**

MUHAMMAD IMRAN

- **I-A & I-C**



Canonical Base Pairs

A-U, G-C

- **"wobble" pairing**

G-U, I-U, I-A & I-C

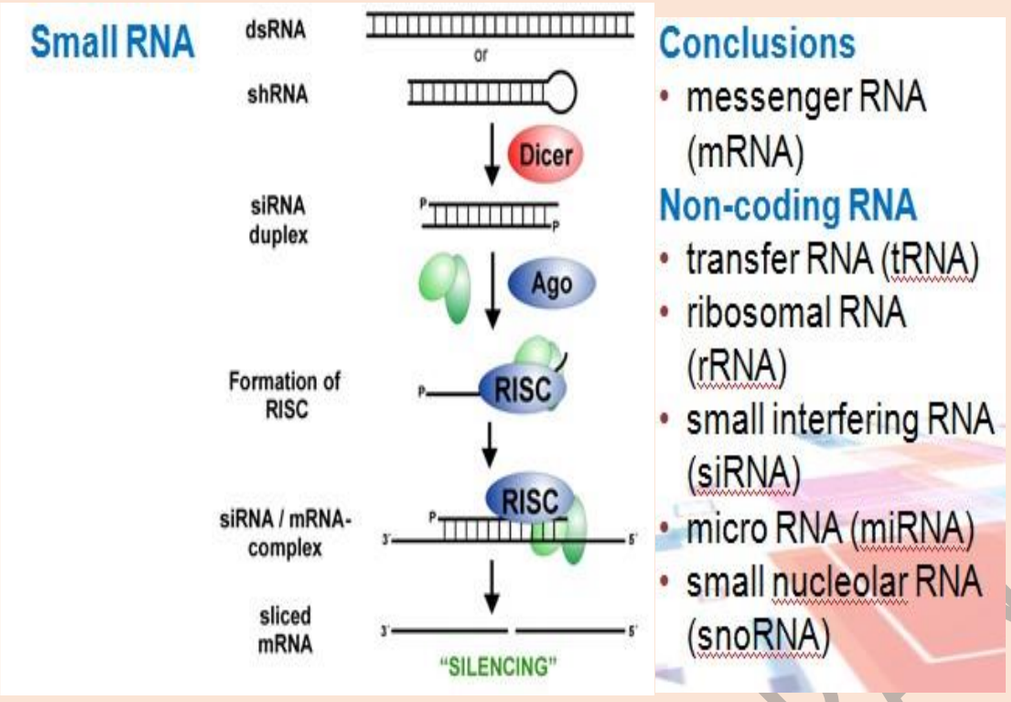
RNA Types messenger RNA

(mRNA) **Non-coding RNA**

transfer RNA (tRNA), ribosomal RNA (rRNA), small interfering RNA (siRNA), micro RNA (miRNA), small nucleolar RNA (snoRNA)

Types of RNAs	Primary Function(s)
mRNA - messenger	translation (protein synthesis), Coding, regulatory
rRNA - ribosomal	translation (protein synthesis)
t-RNA - transfer	translation (protein synthesis)
hnRNA - heterogeneous nuclear	precursors & intermediates of mature mRNAs & other RNAs

Types of RNAs	Primary Function(s)
scRNA - small cytoplasmic	signal recognition particle (SRP), tRNA processing
snRNA - small nuclear snoRNA - small nucleolar	mRNA processing, poly A addition, rRNA processing/maturation/methylation
regulatory RNAs (siRNA, miRNA, etc.)	regulation of transcription and translation, other??



162.RNA Secondary Structure

Some form of RNA can form secondary structures by "pairing up" with itself. This can change its properties dramatically.

Base Pairing

Pairing of bases helps in determining the secondary structure

Aligning bases, based on pairing with each other gives an algorithmic approach to determining the optimal structure

RNA Folding

RNA is produced as a single stranded molecule (unlike DNA)

- Strand folds upon itself to form base pairs & secondary structures
- RNA sequence analysis is different from DNA sequence

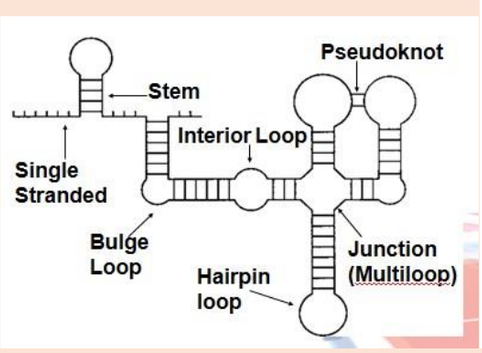
RNA Structure

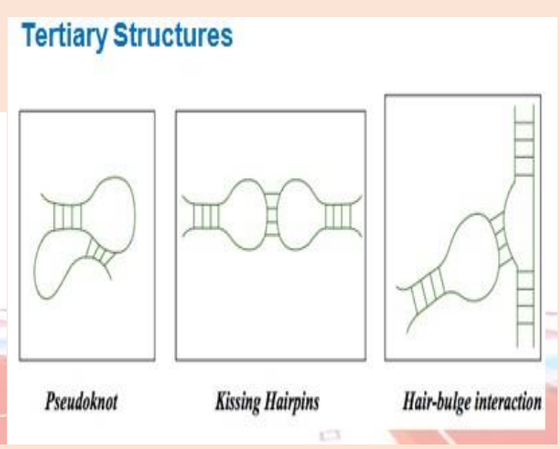
Structures are more conserved than sequences •

Covariation

Secondary Structure representation

2D, Circle plot, Dot plot, Mountain, Parentheses, Tree model





Pseudoknots

Pseudoknots cause a breakdown in the Dynamic Programming Algorithm. In order to form a pseudoknot, checks must be made to ensure base is not already paired – this breaks down the recurrence relations

Conclusions

Some forms of RNA can form secondary structures by "pairing up" with itself. This can change its properties dramatically.

163.RNA Secondary Structure Prediction

There are different approaches to predict secondary structure of RNA

- Energy minimization

Comparative sequence analysis

- Folding and alignment
- Base-Pair Maximization

1. Energy minimization

Dynamic programming approach

Does not require prior sequence alignment**.** Require estimation of energy terms contributing to secondary structure

Assumptions;

Energetically most stable structure is more likely structure. Energy associated with any position is only influenced by local sequence and structure. Neglect pseudoknots

Approach

Energy minimization algorithm predicts secondary structure by minimizing the free energy (□G). □G calculated as sum of individual contributions of:

- Loops
- stacking

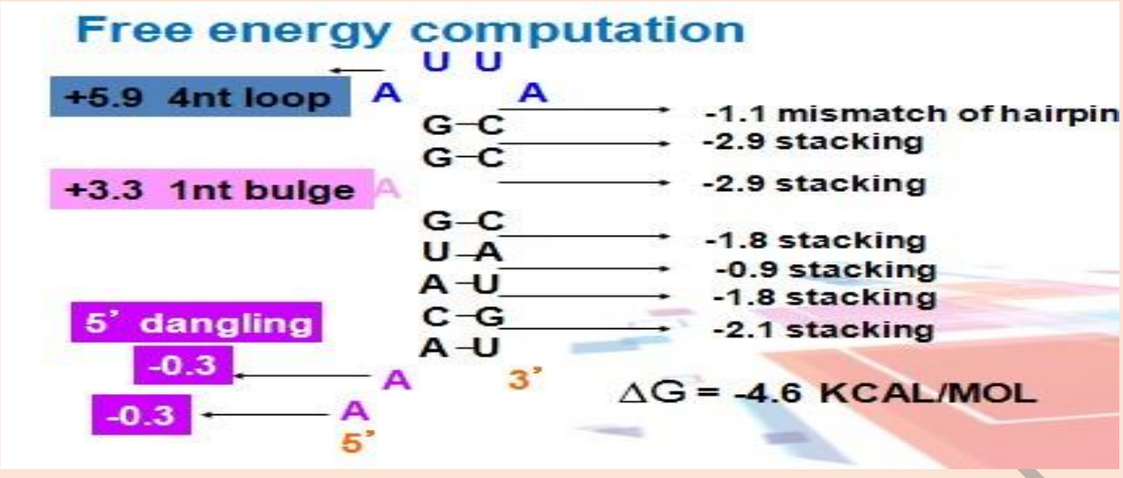
Energy minimization

- Thermodynamic Stability
- Estimated using experimental techniques
- Theory : Most Stable is the Most likely
- No Pseudknots due to algorithm limitations
- Uses Dynamic Programming alignment technique
- Attempts to maximize the score taking into account thermodynamics
- MFOLD and ViennaRNA

Drawbacks

MUHAMMAD IMRAN

Compute only one optimal structure. Usual drawbacks of purely mathematical approaches



MUHAMMAD IMRAN

2. Comparative sequence Analysis

Need a multiple sequence alignment as input.

Requires sequences be similar enough (so that they can be initially aligned), Sequences should be dissimilar enough for covarying substitutions to be detected, comparative analysis produces accurate structure predictions.

164.RNASeq

Calculating transcript abundance and prevalence by Ultra high throughput cDNA sequencing (Mortazwi et al, 2008)

The sequence reads are individually mapped to the source genome and counted to obtain the number and density of reads corresponding to RNA from each

- **known exon**
- **splice event**
- **new candidate gene**

Procedures

Isolation of all mRNA, Convert to cDNA using reverse transcriptase, Sequence the cDNA, Map sequences to the genome.

The more times a given sequence is detected, the more abundantly transcribed it is.

If enough sequences are generated (> 40 Million), *a comprehensive and quantitative view* of the entire transcriptome of an organism or tissue can be obtained (Mortazvi et al, 2008)

Data analysis

Mapping reads, Visualization Genome browser , De novo assembly , Quantification, Differential Gene Expression, Functional Analysis, Gene Networks

In RNASeq, transcript abundance and prevalence is calculated using Ultra high throughput cDNA sequencing

165.RNASeq Normalization

Sequencing reactions may vary across different sequencing plate-forms as well as within different lanes of the same sequencer. Transcript lengths also vary

Raw read counts may vary

RNASeq challenges

Uniformity of sequence coverage, Quantity of sequence required to reliably detect RNAs of lower abundance classes, Quantification and conversion of relative quantification to absolute RNA concentrations, Transcriptomes of organisms with large genomes, containing genes with more complicated structure, present some special challenges

Mapping Biases

Read counts will be higher if sequencer produces more reads. Longer genes will have the probability of mapping more reads than smaller ones

RPKM (reads per kilobase of transcript (or exon model) per million mapped reads)

Method for quantification of transcript levels. **RPKM** measure of read density reflects the molar concentration of a transcript in the starting sample by normalizing for RNA length and for the total read

MUHAMMAD IMRAN

number in the measurement. This facilitates transparent comparison of transcript levels both within and between samples

RPKM

Number of reads mapped per gene length in KB per total reads in that sample in millions

C = Count of Mapped Reads

L= Length of transcript

M = Mapped reads of sample

$$RPKM = \frac{C}{L/1000 * M/1000000}$$

RPKM

$$RPKM = \frac{C}{L/1000 \times M/1000000}$$

$$RPKM = \frac{C \times 10^9}{L\ M}$$

Example: What is the RPKM for a transcript of length 2500KB, with 900 alignments in a sample of 10000000 reads out of which 8000000 reads mapped?

$$RPKM = \frac{900 \times 10^9}{2500 \times 8 \times 10^6} = 4.5$$

RPKM

How many reads are required to map at 1 RPKM with a transcript of 2Kb length from a total of 40 Million Mapped reads?

$$RPKM = \frac{C \times 10^9}{L \times M}$$

$$1 = \frac{C \times 10^9}{2000 \times 40 \times 10^6}$$

$$C = \frac{2000 \times 40 \times 10^6}{10^9} = 80$$

FPKM (Fragments per kilobase of transcript per million mapped reads)

- Paired end RNASeq experiments produce two reads per fragment
- FPKM counts fragments not the reads
- Both reads might not map
- Counting reads might doubble count some fragmnets

(Trapnell et al 2010)

RPM

While comparing the same genes expression across different samples (treatments), normalizing for gene may not be necessary

$$RPM = \frac{C}{M/1000000}$$

Can compare relatively bigger numbers and get more DEG

RPKM reflects the molar concentration of transcript in starting sample normalized for

- **Length of RNA**

- **Total reads in the sample**

It facilitates transcript comparison within and across samples **Topic**

166.Neural Network

The human brain can be described as a biological neural network an interconnected web of neurons transmitting elaborate patterns of electrical signals A neural network is a "connectionist" computational system. Information is processed collectively in parallel throughout a network of nodes. Complex adaptive system.

- Learning processes in biological systems.
- Learning as an optimization process.
- Learning by modification of synaptic strength.

167.Association Rule Mining

It is an important data mining model studied extensively by the database and data mining community. Assume all data are categorical. No good algorithm for numeric data. An association rule has two parts

- an antecedent (if)
- a consequent (then)

Antecedent is an item found in the data. A consequent is an item that is found in combination with the antecedent **Market basket transactions:** t1: {bread, cheese, milk} t2: {apple, eggs, salt, yogurt}

… …

tn: {biscuit, eggs, milk} **Concepts:**

An item: an item/article in a basket **I:**

The set of all items sold in the store **A**

transaction:

Items purchased in a basket; it may have TID (transaction ID) **A**

transactional dataset:

A set of transactions

168.Clustering

Clustering is "a process of organizing objects into groups whose members are similar in some way" A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters.

- Simplifications
- Pattern detection
- Useful in data concept construction
- Unsupervised learning process

  Hierarchical agglomerative general algorithm

- Find the 2 closest objects and merge them into a cluster

MUHAMMAD IMRAN

- Find and merge the next two closest points, where a point is either an individual object or a cluster of objects
- If more than one cluster remains, return to step 2

Applications

- For administrative purposes
- Hospital activity and performance
- Used for researchers • Data used by physician
- For laboratory use etc.

169.Machine Learning

Programming computers to optimize performance criterion using example data or past experience

When to learn? Calculate payroll , Solution needs to be adapted to particular cases (user biometrics)

When To Learn

Human expertise does not exist (navigating on Mars), Humans are unable to explain their expertise (speech recognition), Solution changes in time (routing on a computer network)

Model

Build a model that is a good and useful approximation to the Data

KDD is the non-trivial process of identifying valid, novel, potentially useful, & ultimately understandable patterns in data

Applications

Retail , Finance , Manufacturing , Medicine , Telecommunications , Bioinformatics, Web mining

Retail: Market basket analysis, Customer relationship management (CRM)

Finance: Credit scoring, fraud detection

Manufacturing: Optimization, troubleshooting

Medicine: Medical diagnosis

Telecommunications: Quality of service optimization

Bioinformatics: Motifs, alignment

Web mining: Search engines

Machine Learning

Study of algorithms that Improve performance at some task with exp. Role of Statistics, Role of CS

Applications of ML

Speech recognition, NLP, Computer vision, Medical outcomes analysis/Computational biology, Robot control

170.ML Concepts

- Association Analysis
- Supervised Learning
  - Classification
  - Regression/Prediction
- Unsupervised Learning

MUHAMMAD IMRAN

❖ Reinforcement Learning

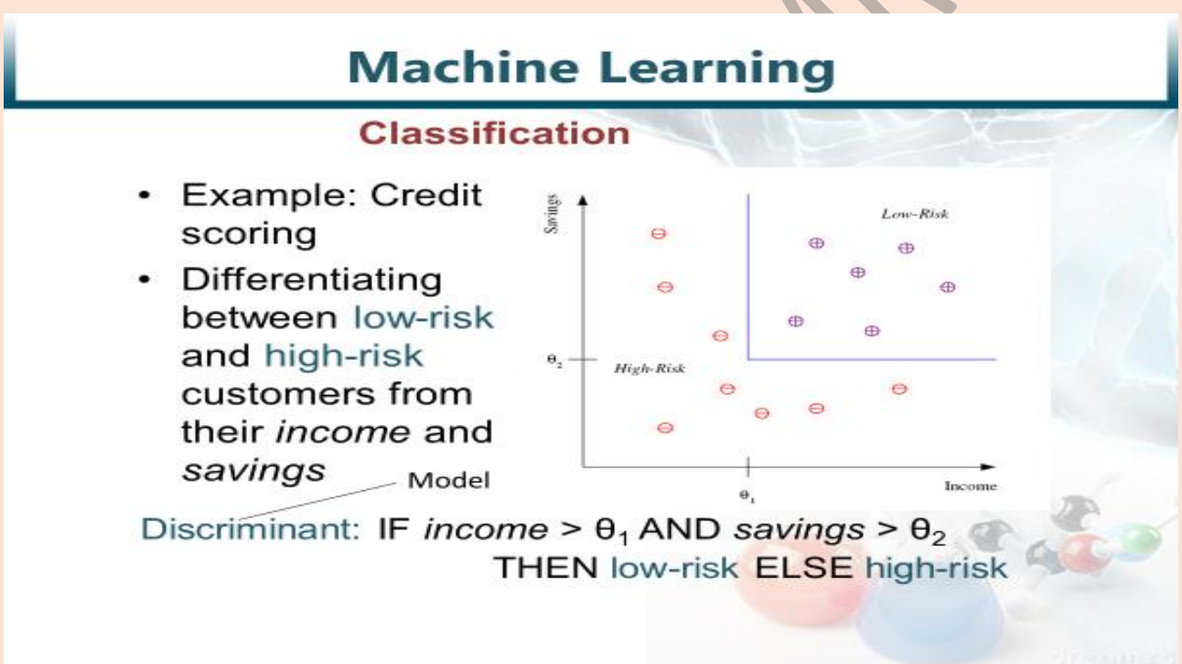
Learning Associations

Basket analysis:

P (*Y* | *X*) probability that somebody who buys *X* also buys *Y* where *X* and *Y* are products/services.

Example: *P* (chips | Beer) = 0.7

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke



Classification Apps

FR: Pose,lighting,occlusion(glasses,beard), make-up, hair style

Character recognition:

Speech recognition:

Medical diagnosis: From symptoms to illnesses

Web Advertizing

Retail: Market basket analysis, Customer relationship management (CRM)

Finance: Credit scoring, fraud detection

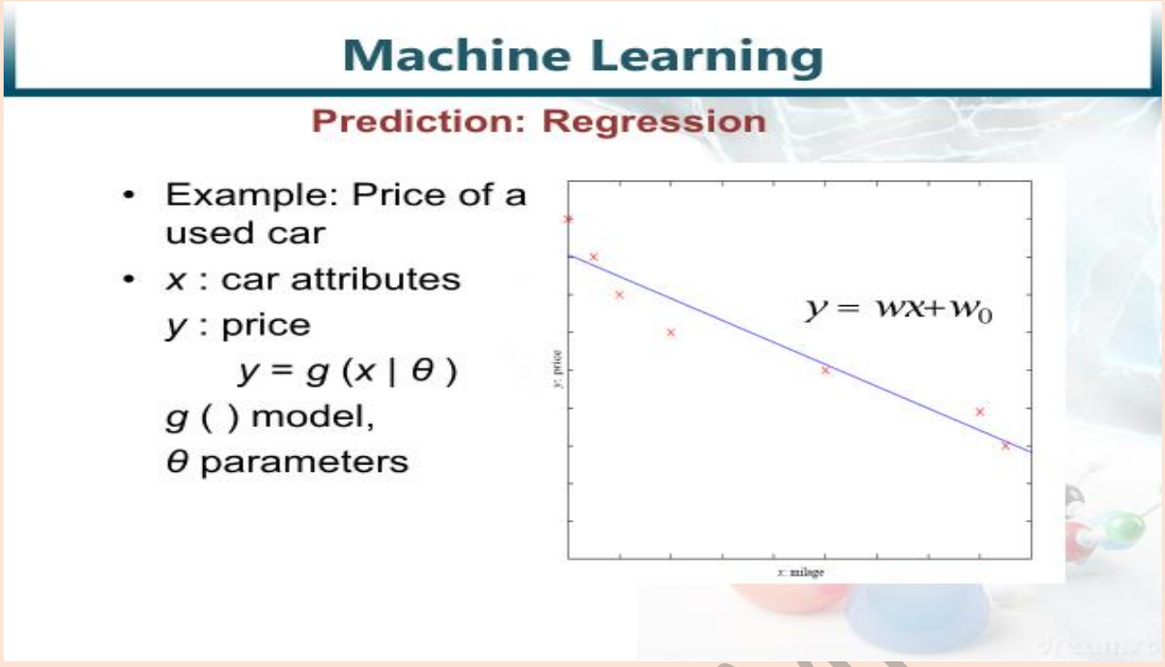
Manufacturing: Optimization, troubleshooting

Medicine: Medical diagnosis

Telecommunications: Quality of service optimization

Bioinformatics: Motifs, alignment

Web mining: Search engines



171.ML Applications

Supervised Learning

- Prediction of future cases
- Knowledge extraction
- Compression
- Outlier detection

Un-supervised Learning

Learning "what normally happens"

No output

Clustering: Grouping

similar instances

Applications

- Customer segmentation in CRM
- Image compression
- Bioinformatics

Reinforcement Learning

Policies: what actions should an agent take in a particular situation

Utility estimation: how good is a state (→used by policy)

No supervised output

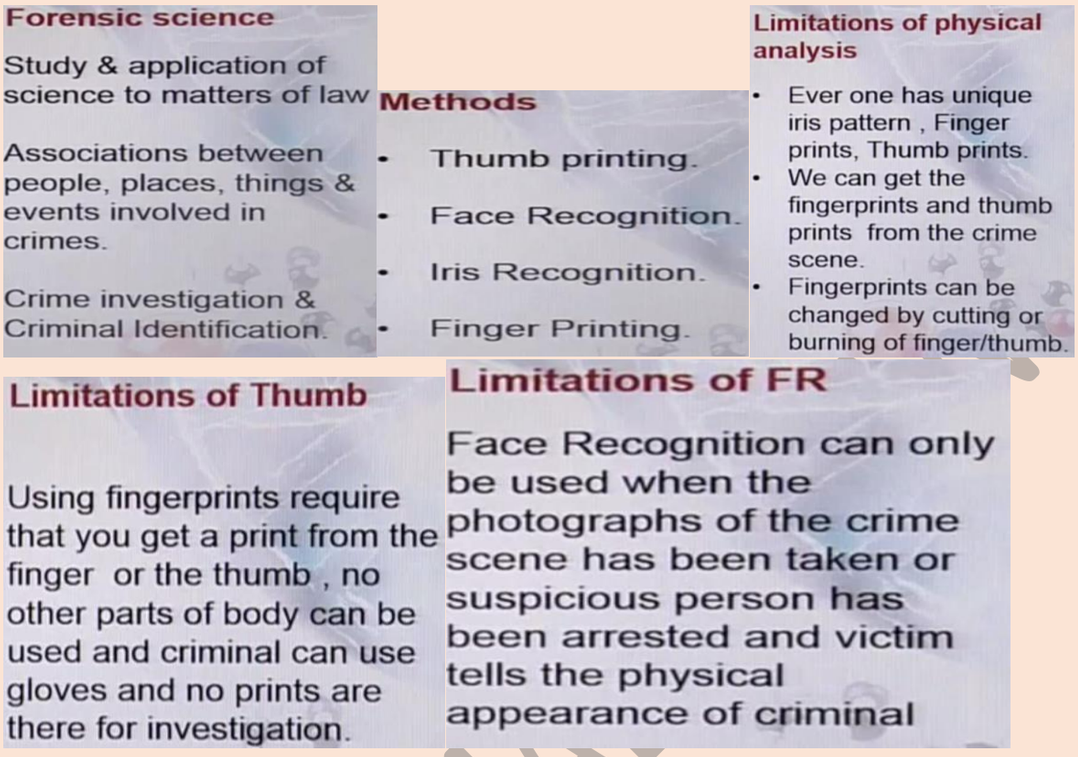
Delayed reward

Credit assignment problem (what was responsible for the outcome)

Applications:

Game playing Robot in a maze Multiple agents, partial observability, ...

172.Forensic Science



173.Advantages of bioinformatics in forensic

DNA Finger Printing / DNA Profiling:

It is a form of forensic identification that is used primarily to identify people.

Advantages

Even though all humans share 99.9% of their DNA sequences, the remaining 0.01% of sequences isunique enough to differentiate people.Because DNA is in every cell of a person's body is present and same , any part of a human's body including dead skin skills, hair, saliva, and more contain DNA sequences.

174. .Methods used in DNA profiling.

DNA finger printing

A forensic scientist will need two pieces of DNA to be compared.For example, DNA discovered at crime scene should be compared to a DNA sample taken from a suspect.

Steps

- Comparison of repeating DNA sequences
- Match b/w two samples

DNA extraction from cell

1. Collecting cell from sample:

Two meters of DNA in cellCollect cells from the sample with buccal swab.Place the swab into Eppendorf tube.

2. Burst cells open to release DNA:

Add the lysis solution to the tube to separate the cells.LS breaks Cell membrane& nuclear envelope causing cells to burst open& release DNA . It also removes histones proteins from DNA.

3. Separate DNA from proteins and Debris:

Cells have stayed in warm to for such a time that DNA is freed from cells.Salt causes proteins & other cellular debris to clump together.Place tube into micro centrifuge.Inside the centrifuge tube spin around and debris &heavy proteins sink in bottom of tube and DNA strands remains distributed throughout liquid.

4. Isolate concentrated DNA:

Add the liquid containing DNA in separate tube Now add isopropyl alcohol to tube.DNA is not soluble in this alcohol so it comes out and it can be seen with naked eye.DNA is collected at bottom tube after placing in centrifuge.

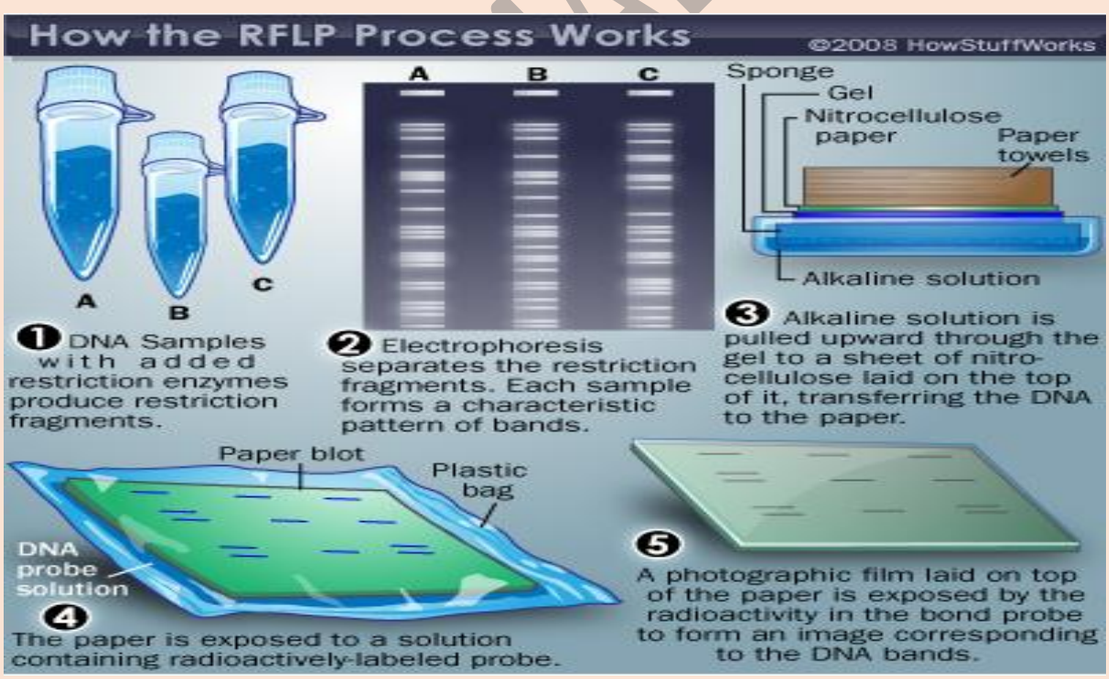
175. DNA Profile

- Encrypted sets of numbers that reflect a person's DNA makeup,
- Variable number tandem repeats(VNTRS).
- Short term tandem repeats (STR) in making the DNA profile of a person.
- These DNA profiles are the basis of a national DNA databases.

DNA profiling processes

- RFLP analysis.
- PCR analysis.
- STR analysis.
- Amp FLP.
- Y-chromosome analysis.

- **Restriction Fragment length Polymorphism (RFLP)**



RFLP

- It analyzes the lengthof strands of DNA that include repeating base pairs (VNTRs).
- Repeated sequence of human genome can be same but the number of times it is repeated is unique to everyone.

RFLP analysis requires investigators to dissolve DNA in an enzyme that breaks the strand at specific points.The number of repeats affects the length of each resulting strand of DNA.Investigators compare samples by comparing the lengths of the strands.

Example: CAT is repeated continuously 13 times in a row. In somebody else, it might be 12 times or 14 or whatever.

Limitation: RFLP analysis requires a fairly large sample of DNA that hasn't been contaminated with dirt.

176. DNA Profile Methods

Polymerase Chain

Reaction (PCR)

Replicate a small amount of DNA to create a larger sample for analysis. First, a heat-stable DNA polymerase -- a special enzyme that binds to the DNA and allows it to replicate -- is added. Next, the DNA sample is heated it to 200 degrees F (93 degrees C) to separate the threads. Then the sample is cooled and reheated. Reheating doubles the number of copies. Process is repeated about 30 times, there is enough DNA for further analysis.

Analyzing STRs

PCR is the first step in analyzing STRs (Short Tandem Repeats), which are very small, specific alleles in a variable number tandem repeat (VNTR).

Short Tandem repeats

Analyzing STRs is more accurate than the RFLP technique because their small size makes them easier to separate. If you want to create a fingerprint, you might look at 20 different STRs at different places in order to create a profile.

STRs

- It is impossible for two persons to have same number of STR repeated in a given sequence.

Y-chromosome Analysis

STRs in Y-chromosome Useful if the sample has mixed DNA. Gender analysis cases. It is processed just like simple STR analysis.

AmpFLP

Amplified fragment length polymorphism, is another technique that uses PCR to replicate DNA. Like RFLP, it first uses a restriction enzyme. Then, the fragments are amplified using PCR and sorted using gel electrophoresis. Can be automated , Doesn't cost very much., DNA sample must be high quality otherwise errors may result, which is the case with most DNA analysis techniques.

Topic 177- Introduction to Drug Discovery

Drug Discovery

Primary objective—

design & discovery of new compounds that are suitable for use as drugs

A team of workers—

chemistry, biology, biochemistry, pharmacology, mathematics, medicine & computing …

requirement

(i) Synthesis of the drug
(ii) Administration method
(iii) Development of tests
(iv) Procedures to establish how it operates in the body
(v) safety assessment
(vi) research into the biological and chemical nature of diseased state.

Drugs: Definition

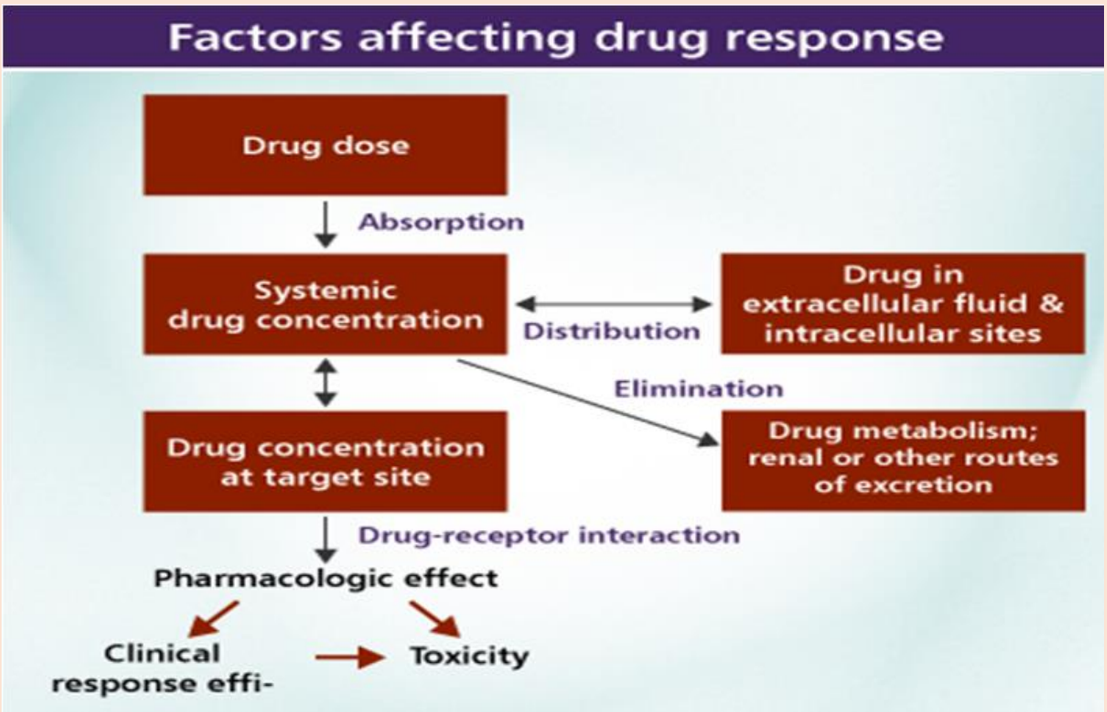
Chemical substances that are used to prevent or cure diseases in humans,
animals and plants

Activity: Pharmaceutical/pharmacological effect on the subject, e.g. Analgesic or β-blocker

MUHAMMAD IMRAN

Potency: quantitative nature of the effect

Drugs Properties: ADMET



Drug: agent used for the psychotic effect by the media or general public. Even the drugs abused have their **activity**. No drug is completely safe. Suitable quantity to cure or excess to be poisonous! E.g. aspirin, paracetamol can be toxic if excesses.

178. Drug Discovery Applications:-

Areas Influencing DD

- Molecular Biology on Drug Discovery
- High-Throughput Screening
- Combinatorial Chemistry

Molecular Biology Influence

Genetic information

Biochemical and chemical terms.Cloning and expressing genes that encode therapeuticallyuseful protein

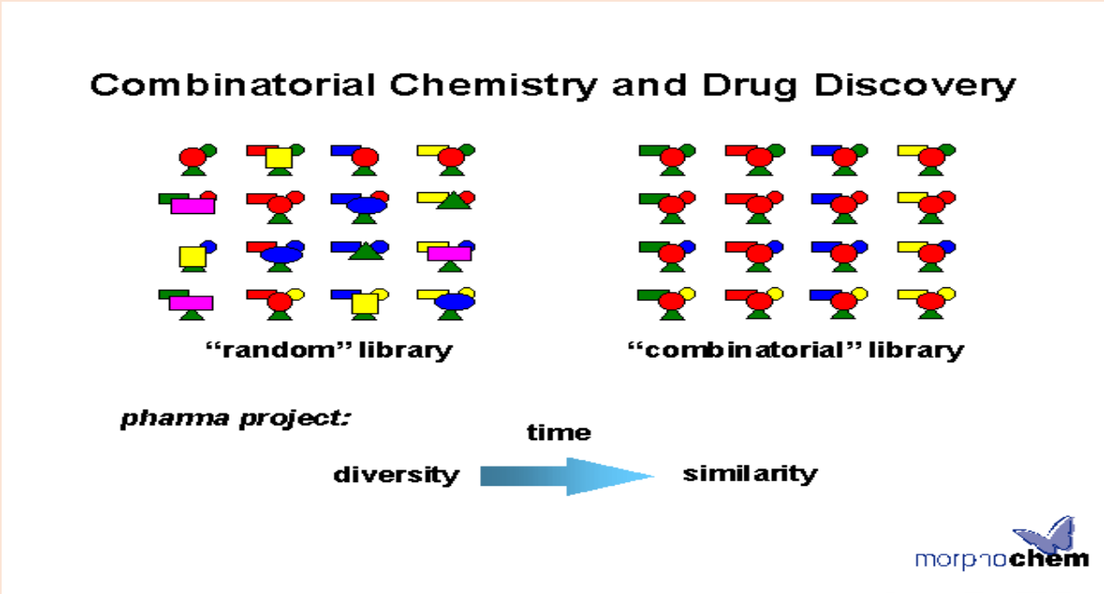
High Throughput Screening

Widely used in the pharmaceutical industry. Automation to quickly assaythe biological or biochemical activity of a large number of drug-like compounds.

Combinatorial Chemistry

Laboratory technique in which millions of molecular constructions can be synthesized and tested for biological activity.

MUHAMMAD IMRAN



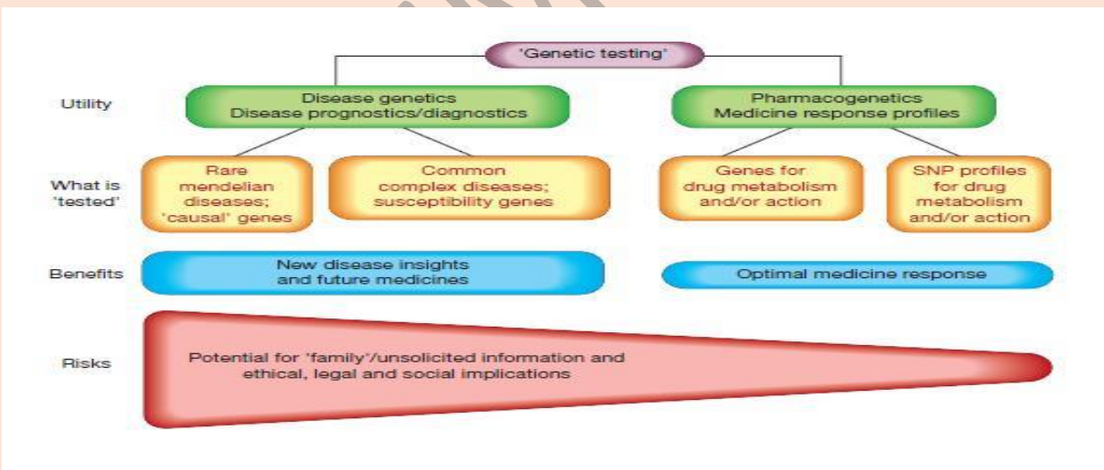
179.Pharmacogenetics

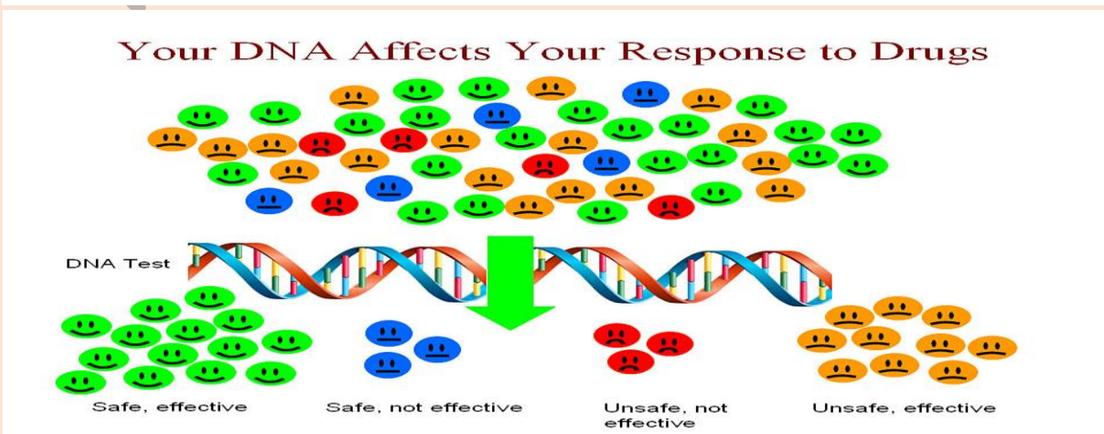
It is the branch of pharmacology concerned with the effect of genetic factors on reactions to drugs.

How people respond to medicines, Correlating heritable genetic variation to drug response.

Defination:-

Biotechnological science combines techniques of medicine, pharmacology & genomics which developing drug therapies to compensate for genetic differences in patients which cause varied responses to a single therapeutic regimen.



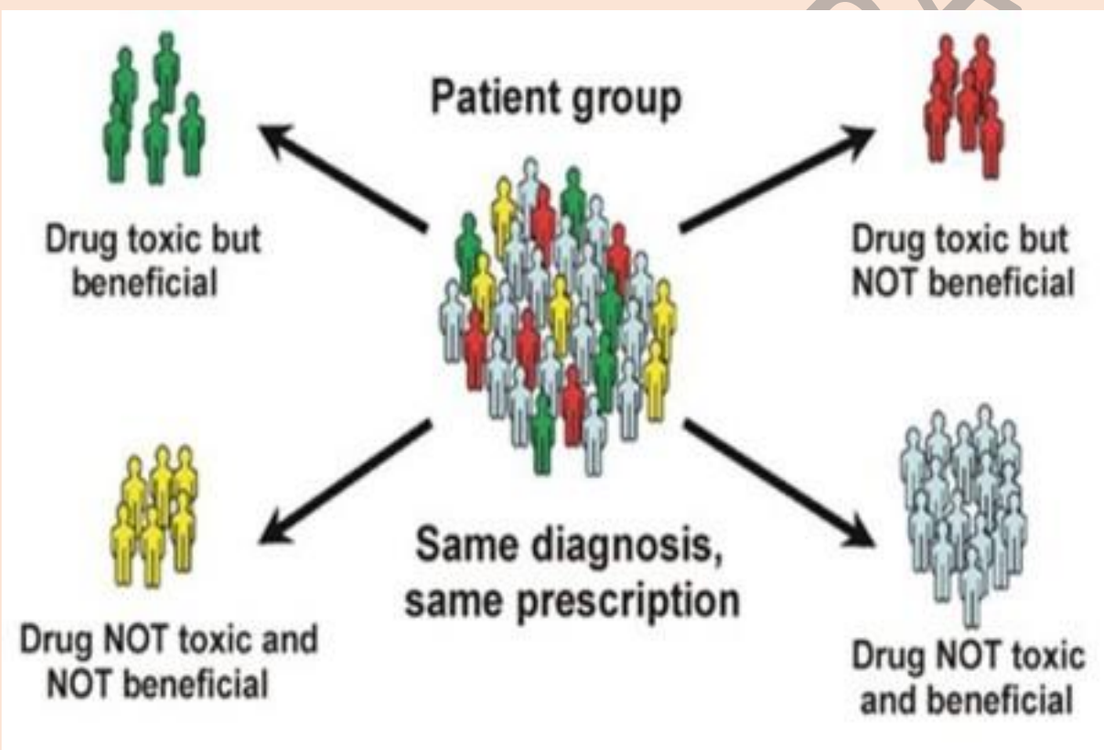


Applications:-

1. Detection of genetic variability of drug effects on the genome level
2. Agent selection
3. Analysis of drug
4. reactions and drug toxicity on gene expression
5. Development of new indications for already approved drugs
6. Discovery of new drug targets
7. Identification of (non) responders in clinical trials of phase I-IV
8. Identification of genotype dependent adverse drug reactions
9. Identification of individuals at risk for severe adverse drug effects

180.Pharmacogenomic applications

How genes affect persons response to drugs. Pharmacology (science of drugs). Genomics (the study of genes and their functions). Develop effective, safe medications & doses tailored to a person's genetic makeup.



Applications:-

Improve drug safety, Reduce ADRs, Tailor treatments to meet patients unique genetic predisposition, Optimal dosing, Improve drug discovery and Improve proof of principle for efficacy trials.

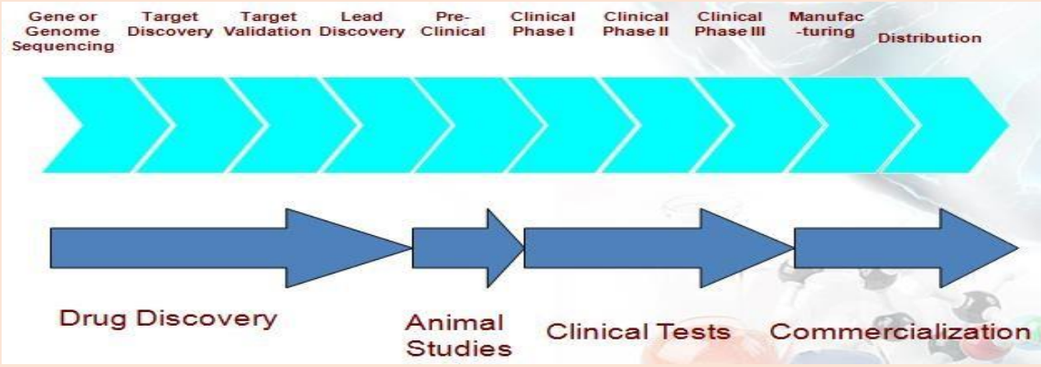
Future

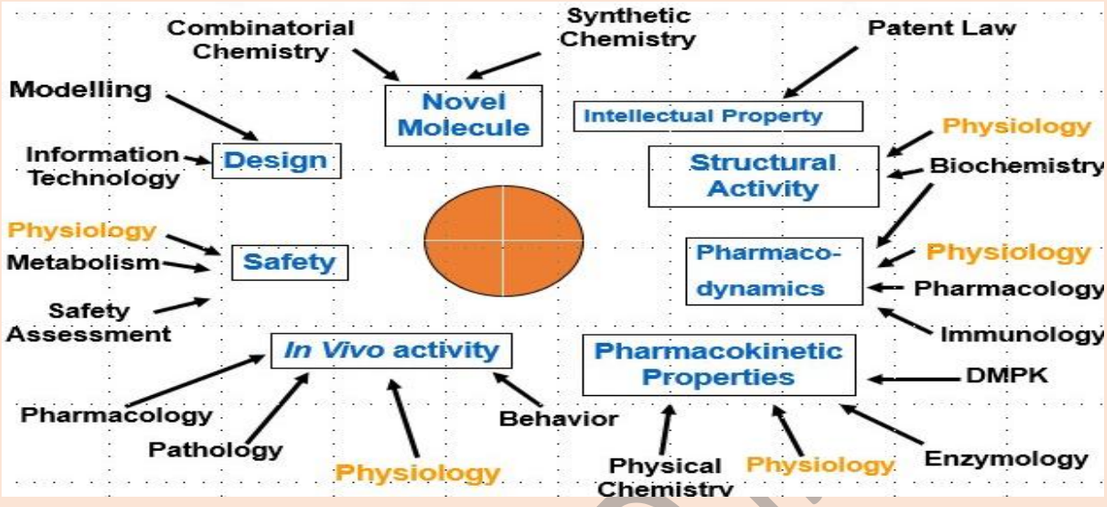
Blessing in research. As a simple example, for nearly a decade the ability to store more information on a hard drive has enabled us to investigate a human genome sequence cheaper.

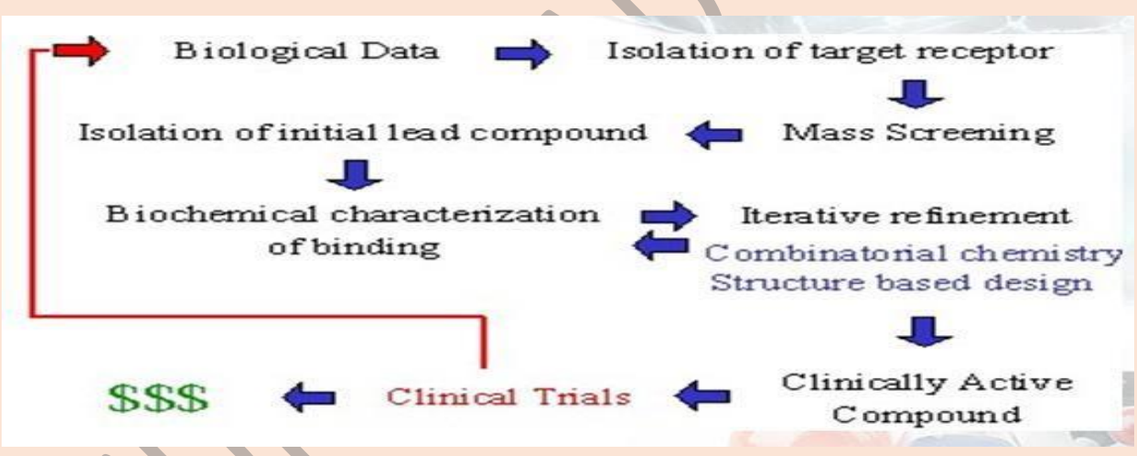
181.182. Drug Discovery – Pipeline

Target Identification, Target Validation, Lead Identification, Lead Optimization, Pre-Clinical

- Pharmacology &Toxicology







183.Drug Discovery Methods

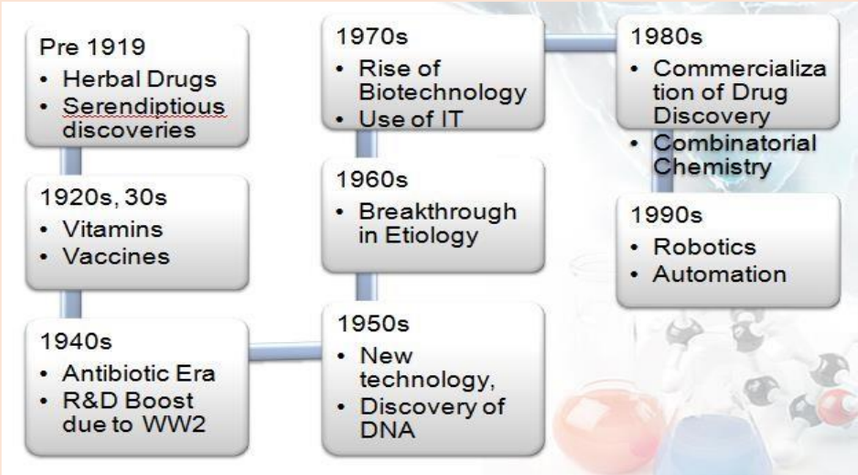
Past:

(i) Identification of active ingredient from traditional remedies

(2) serendipitous discovery.

Current:

Diseases are controlled at molecular & physiological level. Information of Human Genome



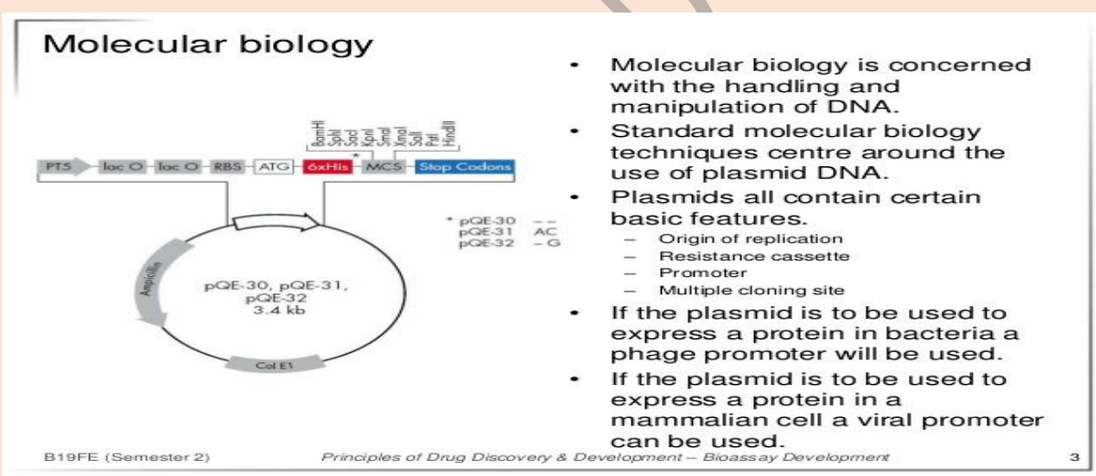
Methods for DD

- ✓ Random Screening
- ✓ Molecular Manipulation
- ✓ Molecular Designing
- ✓ Drug Metabolites • Serendipity

Random Screening

Higher/crude plants, opium, senna, reserpine, etc. Penicillin microorganism. Antibacterials with improved therapeutic profiles.

Molecular Manipulation



Drug Metabolism

Xenobiotic metabolism Biochemical modification of pharmaceutical substances or xenobiotics by living organisms, usually throughspecialized enzymatic systemsLipophilic chemical compounds into more

readily excreted hydrophilic products.Rate of metabolism determines duration & intensity of a drug's pharmacological action

Serendipity

Prototype psychotropic

drugs

Development of psychiatry

Finding of one thing while looking for something else

184.Biomedical annotated corpora

Biomedical researchers are interested

1. Understanding data
2. context information
3. background knowledge
4. curated databases &
5. Literature extensive

What is Annotated Corpora?

Dataset for extraction of disease/treatment entities relations. Corpora are usually constructed for training or evaluation purposes during the development of particular system

Annotation Consumers

The linguistic community typically uses annotation as training data or for specific tasks. An abundance of tools that can *produce* annotations in the specific format of those resources. Biomedical annotation typically used for gene set enrichment analysis

Information deluge

Bio-databases, controlled vocabularies and bio-ontologies encode small fraction of information

Linking text to dbs and ontologies

Curators struggling to process scientific literature. Discovery of facts & events crucial for gaining insights in biosciences: need for text mining

185.Steps for Creation of Biomedical Corpora

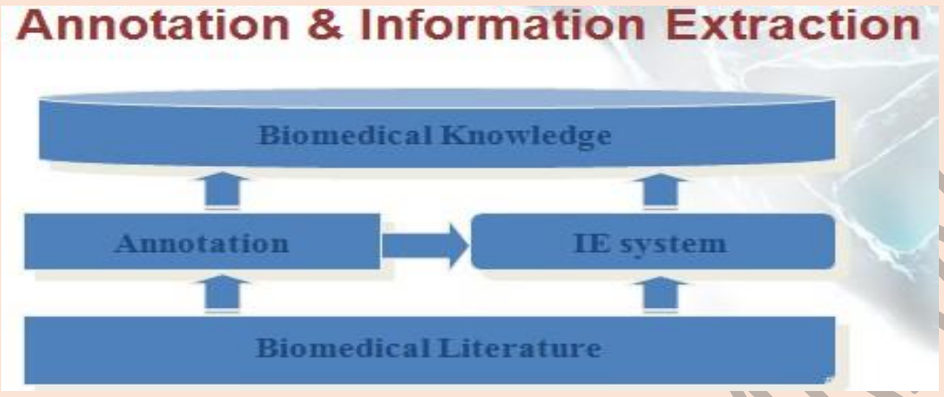
- ✓ Nature of data
- ✓ Standard datasets
- ✓ Formalism
- ✓ Users of dataset
- ✓ Evaluation measures

Data Mining

- ✓ Discovers unsuspected associations
- ✓ Combines & links facts
- ✓ and events
- ✓ Discovers *new* knowledge, finds new associations

IR: yields all relevant . Corpora; gathers, selects, filters documents that may prove useful , Finds what is known

IE: extracts facts & events of interest to user, Finds relevant concepts, facts about concepts. Finds only what we are looking for



- **IE systems can be developed by referencing annotated corpus.**
- **The performance of IE systems can be evaluated by being compared to the annotated corpus.**

Text Mining Pipelines

- ✓ Text documents
- ✓ Retrieval/storage Indexaccess relevant storage
- ✓ Text Processing: word Filters, Pattern filters,Lexicon matching,Ontology, NLP parsingetc, …

Feature Extractions:

Statistical: Word counts, pattern extraction & counts, etc

Domain-specific: Gene Name counts, etc

NLP-specific: Phrase counts, etc

Data Mining: Classification, Clustering, Association, Statistical Analysis,Visual Analysis,

etc …

186.Target Discovery Strategy

Target discovery to clinical application saga

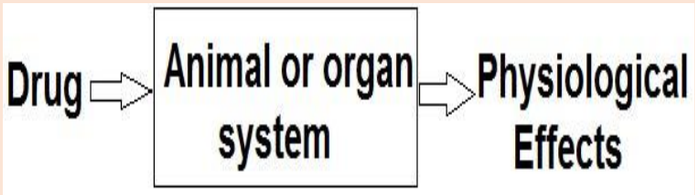
- Physiology-based approach
- Target-based approach

1. **Physiology-based approach**

Is a disease-centric approach in which target is not identified, multiple targets are involved. In vivo screening is done by using drugs, siRNAor antisense oligonucleotides.

MUHAMMAD IMRAN

This process relies on disease, not on the target.



2. **Target-centric approach**

Target based discovery starts with the identification of genes and their protein products. Aim to develop drugs affecting one gene or a molecular mechanism. The identification of diseaserelevant genes in vitro cellular models has been possible due to several tools. Gene-suppression tool used to linked the genes with disease.

Target has two types

1. Genetic
2. Mechanistic

Genetic targets are represented by genes and genes products. Mechanistic targets include mechanism based targets such as receptors, enzymes or genes, identified on the basis of the disease state.

187.Strategies To Identify Possible Drug Targets

First step of drug discovery is to identification of disease-associated targets. Genome sequencing and screening have enhanced opportunities for target identification and lead optimization.

Structure-based Target Discovery

It helped in defining the contours of the cognate surfaces of ligands and their protein targets, permitting optimizationof their potency and selectivity. Some drugs that are originated from structure-based approach:

- Dorzolamide
- Captopril
- Imatinib
- Zanamivir

The protein structure contributes in the following fields:

- Target identification from sequence structure homolog recognition.
- Structural genomics and drug targets.
- Identification of ligand binding region
- Identification of hits and leads
- Structure-guided design and screening

Kinase drug discovery & Kinome

Target Discovery through Cell-based Genetics

MUHAMMAD IMRAN

- New drugs are based on target identification.
- It wants the thorough knowledge of the disease processes and characterization of genes.
- Combination of three target discovery will provide the desired result.

Target Discovery Strategies

Target discovery strategies based on

- Expression profiling Proteomic approach to identify disease related genes based on differential EP, homology and post translational modification
- Biochemical and cell biological assays - To identify genes and proteins linked with disease pathways
- Cell-based genetics - Leads to the discovery of targets by disturbing gene function in whole organisms, corrleation with phenotypes.

Cell-based Genetics

Cell-based assays may lead to the identification of genes involve in cellular transformation, activation, migration and a host of biological processes relevant to a human disease.

Genetic-based Target Identification It

has some methods:

- Positional Cloning- Laboratory technique used to locate the position of a disease associated gene along the chromosomes.
- **Candidate gene approach**

To identify complex disease-linked genes through SNP markers. 10 million in HGP and 3 million identified.

Target class genetic approach

- Is applied to drug target gene families such as proteases, ion channels and GPCRs.
- 24000 protein coding genes & 2400 DTs
- Best candidate are selected from gene family for genetic analysis.

188. Target Validation

- A potential target is identified in the context of a specific disease.
- A validated target is the one that can be manipulated with drugs to produce positive clinical effects in humans.

Requirements for TV

✈ **Genetic Approach**

Gene to disease correlation in animal model

- Forward Genetics
- Reverse Genetics

Target Validation Tools

Helps in assessing their genetic association with disease. The following tools are:

- Antisense agents
- Ribozymes

MUHAMMAD IMRAN

- Peptide nucleic acids
- Transcription factors
- Gene knockouts

Antisense Technology

- Drugs as molecules action cellular proteins (enzymes, receptors).
- Downstream: Block events at nuclear/ribosomal levels.
- Prevent expression of aberrant proteins at earlier stage.
- Oligonucleotide reagents

Ribozymes

- Small RNA mol. cleave other RNA mol. at sequence specific sites.
- Hairpin Ribozyme: GUC

  HammerHead: NUH, H is AUC

- Mode of Delivery: Exogenously , Endogenously
- Peptide NAs: block protein translation
- Transcription Factors: Zinc finder proteins – high Affinity to bind to correct region of DNA.
  Gene Knockouts:

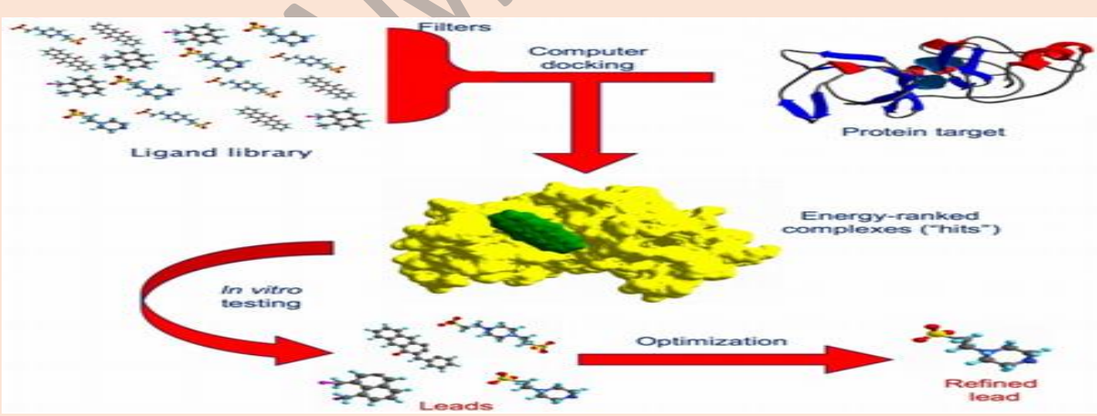
189.Lead Identification

- Compound for biological activity on target.
- Potency threshold
- Libraries of molecules

Lead Identification – Technologies

Virtual Screening:

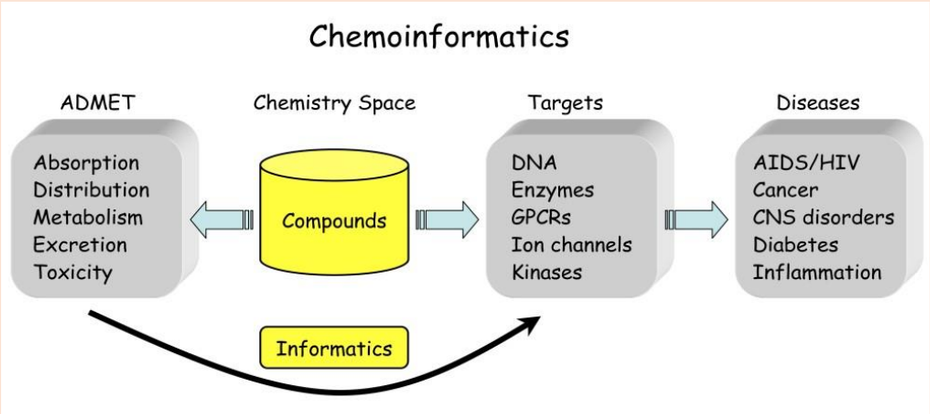
Protein structure , docking Chemical similarity search, Knowledge of compounds against receptor, receptor structure & receptor ligand interactions



Visual Screening

MLCC: Multilevel chemical Compatibility scoring

- Top selling drugs
- Compounds under biological scrutiny
- Anticancer drugs
- Compounds with poor drug like characters



Pharmacophore Mapping

Identify lead compounds against a desired target

Definition: 3-D arrange… **Usage:**

interaction of receptor & legend

DB concept

- **QSAR**

Quantitative structure activity relationship

- **SAR:**

synthesizing & testing a series of structurally related compounds

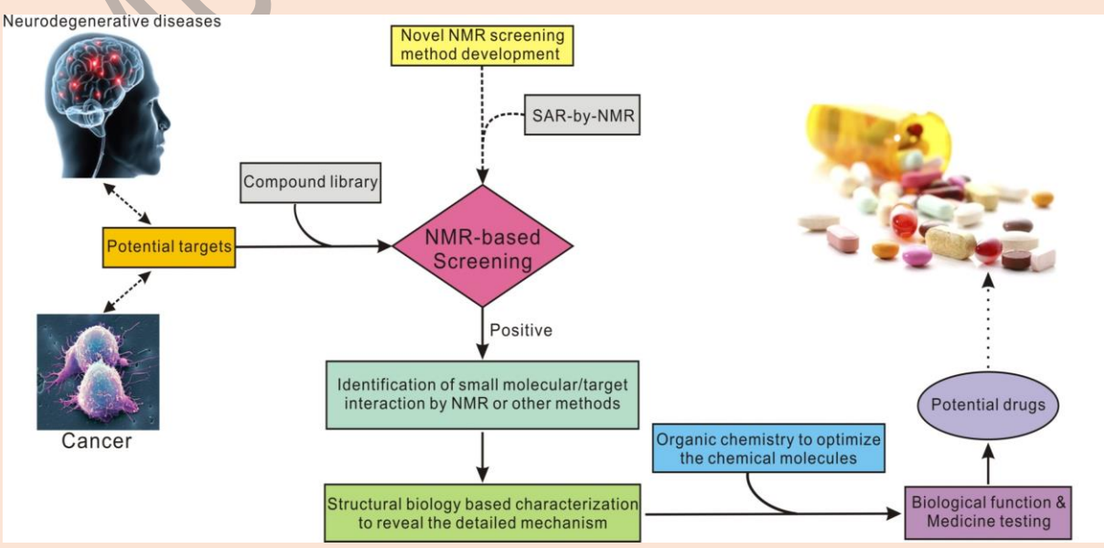
Least squares & KNNs

High Throughput Docking

- Ligand & protein
- Docking algos
- Force fields, knowledge based & empirical

NMR based screening

- Nuclear Magnetic
- Resonance
- 3-D potential DC & tertiary structure of Proteins
- Need of prior information
- SHAPES
- WaterLogsy



Chemical Genetics

- Gene-product function in cellular or organismal context using exogenous ligands

- Knockouts
- Cell cycle - arresting agents

190. Lead Optimization

- Optimize the desirable traits of the lead
- Lead should be amenable for chemistry optimization
- Methods from Lead Identification

Lead Optimization – Technologies

- Medicinal chemists conduct extensive SARs to improve potency and selectivity.
- Improve physicochemical and drug-like properties
- Best molecules are advanced to animal Models & preliminary toxicology

LO Methods

De novo drug desing

Charge distribution, liphophilicity or pk_a of side chains and H-bonds donors and acceptors

SBDD

Structure based drug Design. Effective: 3-D structure of inhibitor with target known. Large no of medicines. Molecular recognition in protein ligand complexes

Drug Like properties

- ADMET in phase 1
- Filters
- Bioavailability, PK, CNS

Pre- Clinical Pharma-cology & Toxicity

- Animals testing
- Xenograft models
  - ADME/T testing and validation

MUHAMMAD IMRAN