Systems & Biomedical
Engineering Department

Bioinformatics

Course**: Database**

Report By:

Salma Hamza
Mohamed El Moatasem
Marwa Adel
Omar Abdelzaher
Rawan Sayed
Remon Alber

Submitted to Dr. Ahmed Hesham

ABSTRACT

Bioinformatics is an interdisciplinary field mainly involving molecular biology and genetics, computer science, mathematics, and statistics. Data intensive, large-scale biological problems are addressed from a computational point of view. The most common problems are modeling biological processes at the molecular level and making inferences from collected data. A bioinformatics solution usually involves the following steps: Collect statistics from biological data. Build a computational model. Solve a computational modeling problem. Test and evaluate a computational algorithm

Table of Contents

List of Figures

Introduction

Bioinformatics is a field of study that uses computation to extract knowledge from biological data. It includes the collection, storage, retrieval, manipulation and modeling of data for analysis, visualization or prediction through the development of algorithms and software.

This chapter gives a brief introduction to bioinformatics by first providing an introduction to biological terminology and then discussing some classical bioinformatics problems organized by the types of data sources.

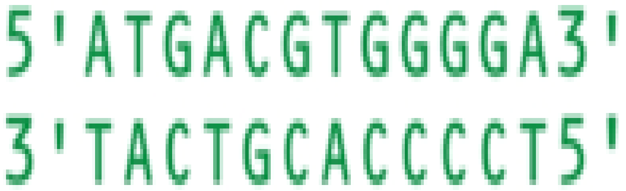
Sequence analysis is the analysis of DNA and protein sequences for clues regarding function and includes subproblems such as identification of homologs, multiple sequence alignment, searching sequence patterns, and evolutionary analyses. Protein structures are three-dimensional data and the associated problems are structure prediction (secondary and tertiary), analysis of protein structures for clues regarding function, and structural alignment.

Gene expression data is usually represented as matrices and analysis of microarray data mostly involves statistics analysis, classification, and clustering approaches. Biological networks such as gene regulatory networks, metabolic pathways, and protein-protein interaction networks are usually modeled as graphs and graph theoretic approaches are used to solve associated problems such as construction and analysis of large-scale networks. Systems biology involves modelling and simulating the complex dynamic interactions between genes, transcripts and proteins using mathematical and computational approaches. We will discuss a simple examples of systems biology model called The Repressilator.

SECTION II: Sequence Analysis

2.1 Sequence

Computers became essential in molecular biology when protein sequences became available after Frederick Sanger determined the sequence of insulin in the early 1950s. Comparing multiple sequences manually turned out to be impractical. A pioneer in the field was Margaret Oakley Dayhoff.[12] She compiled one of the first protein sequence databases, initially published as books[13] and pioneered methods of sequence alignment and molecular evolution.[14] Another early contributor to bioinformatics was Elvin A. Kabat, who pioneered biological sequence analysis in 1970 with his comprehensive volumes of antibody sequences released with Tai Te Wu between 1980 and 1991.



5'ATGACGTGGGA3'
3'TACTGCACCCT5'

Figure 1:Sequences of genetic material

2.2 Sequence Analysis

Since the Phage Φ-X174 was sequenced in 1977,the DNA sequences of thousands of organisms have been decoded and stored in databases. This sequence information is analyzed to determine genes that encode proteins, RNA genes, regulatory sequences, structural motifs, and repetitive sequences. A comparison of genes within a species or between different species can show similarities between protein functions, or relations between species (the use of molecular systematics to construct phylogenetic trees). With the growing amount of data, it long ago became impractical to analyze DNA sequences manually. Today[when?], computer programs such as BLAST are used daily to search sequences from more than 260 000 organisms, containing over 190 billion nucleotides. These programs can compensate for mutations (exchanged, deleted or inserted bases) in the DNA sequence, to identify sequences that are related, but not identical. A variant of this sequence alignment is used in the sequencing process itself. For the special task of taxonomic classification of sequence snippets, modern k-mer based software like Kraken achieves throughput unreachable by alignment methods..

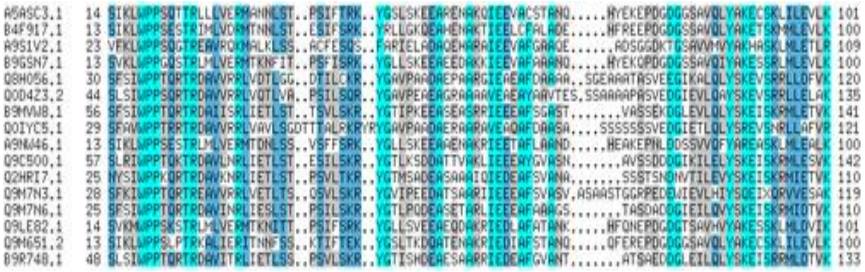


Figure 2 :The sequences of different genes or proteins may be aligned side-by-side to measure their similarity.

2.3 DNA Sequencing

` Before sequences can be analyzed they have to be obtained from the data storage bank example the Genbank. DNA sequencing is still a non-trivial problem as the raw data may be noisy or afflicted by weak signals. Algorithms have been developed for base calling for the various experimental approaches to DNA sequencing.

The DNA double helix model bears a resemblance to a spiral staircase, with two sugar-phosphate backbones and the paired bases in the middle of the helix. This structure demonstrates two of the most significant attributes of the molecule:

- First, it can be replicated, as each strand can act as a mould to produce the complementary strand.
- Second, it can store information in the nucleotides linear concatenation along each strand .

2.4 Sequence Assembly

Most DNA sequencing techniques produce short fragments of sequence that need to be assembled to obtain complete gene or genome sequences. The so-called shotgun sequencing technique (which was used, for example, by The Institute for Genomic Research (TIGR) to sequence the first bacterial genome, Haemophilus influenzae)generates the sequences of many thousands of small DNA fragments (ranging from 35 to 900 nucleotides long, depending on the sequencing technology). The ends of these fragments overlap and, when aligned properly by a genome assembly program, can be used to reconstruct the complete genome. Shotgun sequencing yields sequence data quickly, but the task of assembling the fragments can be quite complicated for larger genomes. For a genome as large as the human genome, it may take many days of CPU time on large-memory, multiprocessor computers to assemble the fragments, and the resulting assembly usually contains numerous gaps that must be filled in later. Shotgun sequencing is the method of choice for virtually all genomes sequenced today[when?], and genome assembly algorithms are a critical area of bioinformatics research.

2.5 Genome *A*nnotation

In the context of genomics, annotation is the process of marking the genes and other biological features in a DNA sequence. This process needs to be automated because most genomes are too large to annotate by hand, not to mention the desire to annotate as many genomes as possible, as the rate of sequencing has ceased to pose a bottleneck. Annotation is made possible by the fact that genes have recognisable start and stop regions, although the exact sequence found in these regions can vary between genes.

The first description of a comprehensive genome annotation system was published in 1995 by the team at The Institute for Genomic Research that performed the first complete sequencing and analysis of the genome of a free-living organism, the bacterium Haemophilus influenzae.Owen White designed and built a software system to identify the genes encoding all proteins, transfer RNAs, ribosomal RNAs (and other sites) and to make initial functional assignments. Most current genome annotation systems work similarly, but the programs available for analysis of genomic DNA, such as the GeneMark program trained and used to find protein-coding genes in Haemophilus influenzae, are constantly changing and improving.

Following the goals that the Human Genome Project left to achieve after its closure in 2003, a new project developed by the National Human Genome Research Institute in the U.S appeared. The so-called ENCODE project is a collaborative data collection of the functional elements of the human genome that uses next-generation DNA-sequencing technologies and genomic tiling arrays, technologies able to automatically generate large amounts of data at a dramatically reduced per-base cost but with the same accuracy (base call error) and fidelity (assembly error).

2.6 Computational Evolutionary Biology

Evolutionary biology is the study of the origin and descent of species , as well as their change over time. Informatics has assisted evolutionary biologists by enabling researchers to:

- trace the evolution of a large number of organisms by measuring changes in their DNA , rather than through physical taxonomy or physiological observations alone,
- compare entire genomes , which permits the study of more complex evolutionary events, such as gene duplication, horizontal gene transfer, and the prediction of factors important in bacterial speciation,
- build complex computational population genetics models to predict the outcome of the system over time
- track and share information on an increasingly large number of species and organisms

Future work endeavours to reconstruct the now more complex tree of life.

The area of research within computer science that uses genetic algorithms is sometimes confused with computational evolutionary biology, but the two areas are not necessarily related.

2.7 Comparative Genomics

The core of comparative genome analysis is the establishment of the correspondence between genes (orthology analysis) or other genomic features in different organisms. It is these intergenomic maps that make it possible to trace the evolutionary processes responsible for the divergence of two genomes. A multitude of evolutionary events acting at various organizational levels shape genome evolution. At the lowest level, point mutations affect individual nucleotides. At a higher level, large chromosomal segments undergo duplication, lateral transfer, inversion, transposition, deletion and insertion. Ultimately, whole genomes are involved in processes of hybridization, polyploidization and endosymbiosis, often leading to rapid speciation. The complexity of genome evolution poses many exciting challenges to developers of mathematical models and algorithms, who have recourse to a spectrum of algorithmic, statistical and mathematical techniques, ranging from exact, heuristics, fixed parameter and approximation algorithms for problems based on parsimony models to Markov chain Monte Carlo algorithms for Bayesian analysis of problems based on probabilistic models.

2.8 Pan Genomics

Pan genomics is a concept introduced in 2005 by Tettelin and Medini which eventually took root in bioinformatics. Pan genome is the complete gene repertoire of a particular taxonomic group: although initially applied to closely related strains of a species, it can be applied to a larger context like genus, phylum etc. It is divided in two parts- The Core genome: Set of genes common to all the genomes under study (These are often housekeeping genes vital for survival) and The Dispensable/Flexible Genome: Set of genes not present in all but one or some genomes under study. A bioinformatics tool BPGA can be used to characterize the Pan Genome of bacterial species.

2.8 Genetics of Disease

With the advent of next-generation sequencing we are obtaining enough sequence data to map the genes of complex diseases infertility, breast cancer or Alzheimer's disease. Genome-wide association studies are a useful approach to pinpoint the mutations responsible for such complex diseases. Through these studies, thousands of DNA variants have been identified that are associated with similar diseases and traits. Furthermore, the possibility for genes to be used at prognosis, diagnosis or treatment is one of the most essential applications. Many studies are discussing both the promising ways to choose the genes to be used and the problems and pitfalls of using genes to predict disease presence or prognosis.

2.9 Analysis of Mutations in Cancer

In cancer, the genomes of affected cells are rearranged in complex or even unpredictable ways. Massive sequencing efforts are used to identify previously unknown point mutations in a variety of genes in cancer. Bioinformaticians continue to produce specialized automated systems to manage the sheer volume of sequence data produced, and they create new algorithms and software to compare the sequencing results to the growing collection of human genome sequences and germline polymorphisms. New physical detection technologies are employed, such as oligonucleotide microarrays to identify chromosomal gains and losses (called comparative genomic hybridization), and single-nucleotide polymorphism arrays to detect known *point mutations*. These detection methods simultaneously measure several hundred thousand sites throughout the genome, and when used in high-throughput to measure thousands of samples, generate terabytes of data per experiment. Again the massive amounts and new types of data generate new opportunities for bioinformaticians. The data is often found to contain considerable variability, or noise, and thus Hidden Markov model and change-point analysis methods are being developed to infer real copy number changes.

` Two important principles can be used in the analysis of cancer genomes bioinformatically pertaining to the identification of mutations in the exome. First, cancer is a disease of accumulated somatic mutations in genes. Second cancer contains driver mutations which need to be distinguished from passengers.

With the breakthroughs that this next-generation sequencing technology is providing to the field of Bioinformatics, cancer genomics could drastically change. These new methods and software allow bioinformaticians to sequence many cancer genomes quickly and affordably. This could create a more flexible process for classifying types of cancer by analysis of cancer driven mutations in the genome. Furthermore, tracking of patients while the disease progresses may be possible in the future with the sequence of cancer samples.

Another type of data that requires novel informatics development is the analysis of lesions found to be recurrent among many tumors.

SECTION III: Gene and Protein expression

3.1 Analysis of Gene Expression

The expression of many genes can be determined by measuring mRNA levels with multiple techniques including microarrays, expressed cDNA sequence tag (EST) sequencing, serial analysis of gene expression (SAGE) tag sequencing, massively parallel signature sequencing (MPSS), RNA-Seq, also known as "Whole Transcriptome Shotgun Sequencing" (WTSS), or various applications of multiplexed in-situ hybridization. All of these techniques are extremely noise-prone and/or subject to bias in the biological measurement, and a major research area in computational biology involves developing statistical tools to separate signal from noise in high-throughput gene expression studies.[Such studies are often used to determine the genes implicated in a disorder: one might compare microarray data from cancerous epithelial cells to data from non-cancerous cells to determine the transcripts that are up-regulated and down-regulated in a particular population of cancer cells.

3.2 Analysis of Protein Expression

Protein microarrays and high throughput (HT) mass spectrometry (MS) can provide a snapshot of the proteins present in a biological sample. Bioinformatics is very much involved in making sense of protein microarray and HT MS data; the former approach faces similar problems as with microarrays targeted at mRNA, the latter involves the problem of matching large amounts of mass data against predicted masses from protein sequence databases, and the complicated statistical analysis of samples where multiple, but incomplete peptides from each protein are detected. Cellular protein localization in a tissue context can be achieved through affinity proteomics displayed as spatial data based on immunohistochemistry and tissue microarrays.

3.3 Analysis of Regulation

Gene regulation is the complex orchestration of events by which a signal, potentially an extracellular signal such as a hormone, eventually leads to an increase or decrease in the activity of one or more proteins. Bioinformatics techniques have been applied to explore various steps in this process.

For example, gene expression can be regulated by nearby elements in the genome. Promoter analysis involves the identification and study of sequence motifs in the DNA surrounding the coding region of a gene. These motifs influence the extent to which that region is transcribed into mRNA. Enhancer elements far away from the promoter can also regulate gene expression, through three-dimensional looping interactions. These interactions can be determined by bioinformatic analysis of chromosome conformation capture experiments.

Expression data can be used to infer gene regulation: one might compare microarray data from a wide variety of states of an organism to form hypotheses about the genes involved in each state. In a single-cell organism, one might compare stages of the cell cycle, along with various stress conditions (heat shock, starvation, etc.). One can then apply clustering algorithms to that expression data to determine which genes are co-expressed. For example, the upstream regions (promoters) of co-expressed genes can be searched for over-represented regulatory elements. Examples of clustering algorithms applied in gene clustering are k-means clustering, self-organizing maps (SOMs), hierarchical clustering, and consensus clustering methods.

SECTION IV: Analysis of cellular organization

Several approaches have been developed to analyze the location of organelles, genes, proteins, and other components within cells. This is relevant as the location of these components affects the events within a cell and thus helps us to predict the behavior of biological systems. A gene ontology category, *cellular compartment*, has been devised to capture subcellular localization in many biological databases.

4.1 Microscopy and image analysis

Microscopic pictures allow us to locate both organelles as well as molecules. It may also help us to distinguish between normal and abnormal cells, e.g. in cancer.

4.2 Protein localization

The localization of proteins helps us to evaluate the role of a protein. For instance, if a protein is found in the nucleus it may be involved in gene regulation or splicing. By contrast, if a protein is found in mitochondria, it may be involved in respiration or other metabolic processes. Protein localization is thus an important component of protein function prediction. There are well developed protein subcellular localization prediction resources available, including protein subcellular location databases, and prediction tools.

4.3 Nuclear organization of chromatin

Data from high-throughput chromosome conformation capture experiments, such as Hi-C (experiment) and ChIA-PET, can provide information on the spatial proximity of DNA loci. Analysis of these experiments can determine the three-dimensional structure and nuclear organization of chromatin. Bioinformatic challenges in this field include partitioning the genome into domains, such as Topologically Associating Domains (TADs), that are organised together in three-dimensional space.

SECTION V : Structural Bioinformatics

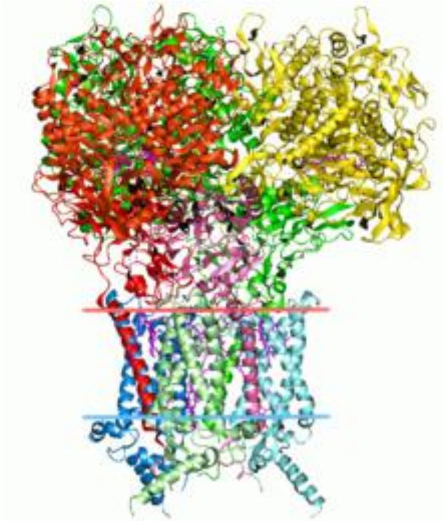


Figure 3 :3-dimensional protein structures such as this one are common subjects in bioinformatic analyses.

Protein structure prediction is another important application of bioinformatics. The amino acid sequence of a protein, the so-called primary structure, can be easily determined from the sequence on the gene that codes for it. In the vast majority of cases, this primary structure uniquely determines a structure in its native environment. (Of course, there are exceptions, such as the bovine spongiform encephalopathy (mad cow disease) prion.) Knowledge of this structure is vital in understanding the function of the protein. Structural information is usually classified as one of *secondary*, *tertiary* and *quaternary* structure. A viable general solution to such predictions remains an open problem. Most efforts have so far been directed towards heuristics that work most of the time.

One of the key ideas in bioinformatics is the notion of homology. In the genomic branch of bioinformatics, homology is used to predict the function of a gene: if the sequence of gene *A*, whose function is known, is homologous to the sequence of gene *B,* whose function is unknown, one could infer that B may share A's function. In the structural branch of bioinformatics, homology is used to determine which parts of a protein are important in structure formation and interaction with other proteins. In a technique called homology modeling, this information is used to predict the structure of a protein once the structure of a homologous protein is known. This currently remains the only way to predict protein structures reliably.

One example of this is hemoglobin in humans and the hemoglobin in legumes (leghemoglobin), which are distant relatives from the same protein superfamily. Both serve the same purpose of transporting oxygen in the organism. Although both of these proteins have completely different amino acid sequences, their protein structures are virtually identical, which reflects their near identical purposes and shared ancestor.

Other techniques for predicting protein structure include protein threading and *de novo* (from scratch) physics-based modeling.

Another aspect of structural bioinformatics include the use of protein structures for Virtual Screening models such as Quantitative Structure-Activity Relationship models and proteochemometric models (PCM). Furthermore, a protein's crystal structure can be used in simulation of for example ligand-binding studies and *in silico* mutagenesis studies.

SECTION VI : Network and Systems biology

Network analysis seeks to understand the relationships within biological networks such as metabolic or protein–protein interaction networks. Although biological networks can be constructed from a single type of molecule or entity (such as genes), network biology often attempts to integrate many different data types, such as proteins, small molecules, gene expression data, and others, which are all connected physically, functionally, or both.

Systems biology involves the use of computer simulations of cellular subsystems (such as the networks of metabolites and enzymes that comprise metabolism, signal transduction pathways and gene regulatory networks) to both analyze and visualize the complex connections of these cellular processes. Artificial life or virtual evolution attempts to understand evolutionary processes via the computer simulation of simple (artificial) life forms.

6.1 Molecular interaction networks

Figure 4 : Interactions between proteins are frequently visualized and analyzed using networks. This network is made up of protein–protein interactions from Treponema pallidum, the causative agent of syphilis and other diseases

. Tens of thousands of three-dimensional protein structures have been determined by X-ray crystallography and protein nuclear magnetic resonance spectroscopy (protein NMR) and a central question in structural bioinformatics is whether it is practical to predict possible protein–protein interactions only based on these 3D shapes, without performing protein–protein interaction experiments. A variety of methods have been developed to tackle the protein–protein docking problem, though it seems that there is still much work to be done in this field.

Other interactions encountered in the field include Protein–ligand (including drug) and protein–peptide. Molecular dynamic simulation of movement of atoms about rotatable bonds is the fundamental principle behind computational algorithms, termed docking algorithms, for studying molecular interactions.

SECTION VII : Bioinformatics with Machine Learning

Figure 5 : Explore the world of Bioinformatics with Machine Learning

Here is a brief introduction of Bioinformatics and how a machine learning classification algorithm can be used to classify the type of cancer in each patient by their gene expressions

Bioinformatics is a field of study that uses computation to extract knowledge from biological data. It includes the collection, storage, retrieval, manipulation and modeling of data for analysis, visualization or prediction through the development of algorithms and software.We can quote it in a simpler way **“Bioinformatics deals with computational and mathematical approaches for understanding and processing biological data”.**

It is an interdisciplinary field in which new computational methods are developed to analyze biological data and to make biological discoveries. For example, two typical tasks in genetics and genomics are the processes of sequencing and annotating an organism’s complete set of DNA. In neurosciences, neuroimaging techniques, such as computerized tomography (CT), positron emission tomography (PET), functional magnetic resonance imaging (fMRI), and diffusion tensor imaging (DTI), are used to study brains in vivo and to understand the inner workings of the nervous system.

7.1 Application of Machine Learning

The application of Machine Learning to biological and neuroimaging data opens new frontiers for biomedical engineering: improving our understanding of complex diseases such as cancer or neurodegenerative and psychiatric disorders. Advances in this field can ultimately lead to the development of automated diagnostic tools and of precision medicine, which consists of targeting custom medical treatments considering individual variability, lifestyle, and environment.

Prior to the emergence of machine learning algorithms, bioinformatics algorithms had to be explicitly programmed by hand which, for problems such as protein structure prediction, proves extremely difficult.

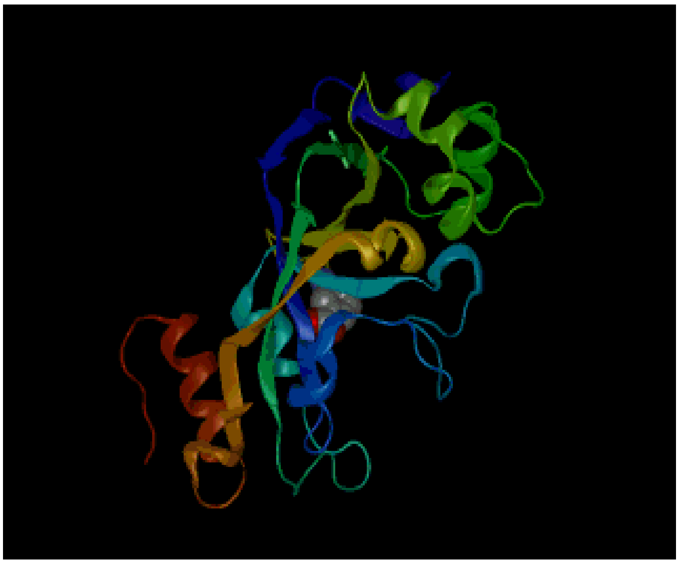


Figure 6 : 3-D structure of protein sequence

Machine learning techniques such as deep learning enable the algorithm to make use of automatic feature learning which means that based on the dataset alone, the algorithm can learn how to combine multiple features of the input data into a more abstract set of features from which to conduct further learning. This multi-layered approach to learning patterns in the input data allows such systems to make quite complex predictions when trained on large datasets. In recent years, the size and number of available biological datasets have skyrocketed, enabling bioinformatics researchers to make use of these machine learning algorithms.

Machine learning has been applied to six biological domains: Genomics, Proteomics, Microarrays, Systems biology, Stroke diagnosis, and Text mining.

7.1.1 Genomics

It is an interdisciplinary field of biology focusing on the structure, function, evolution, mapping, and editing of genomes. A Genome is an organism’s complete set of DNA, including all of its genes. There is an increasing need for the development of machine learning systems that can automatically determine the location of protein-encoding genes within a given DNA sequence and this problem in computational biology is known as gene prediction.

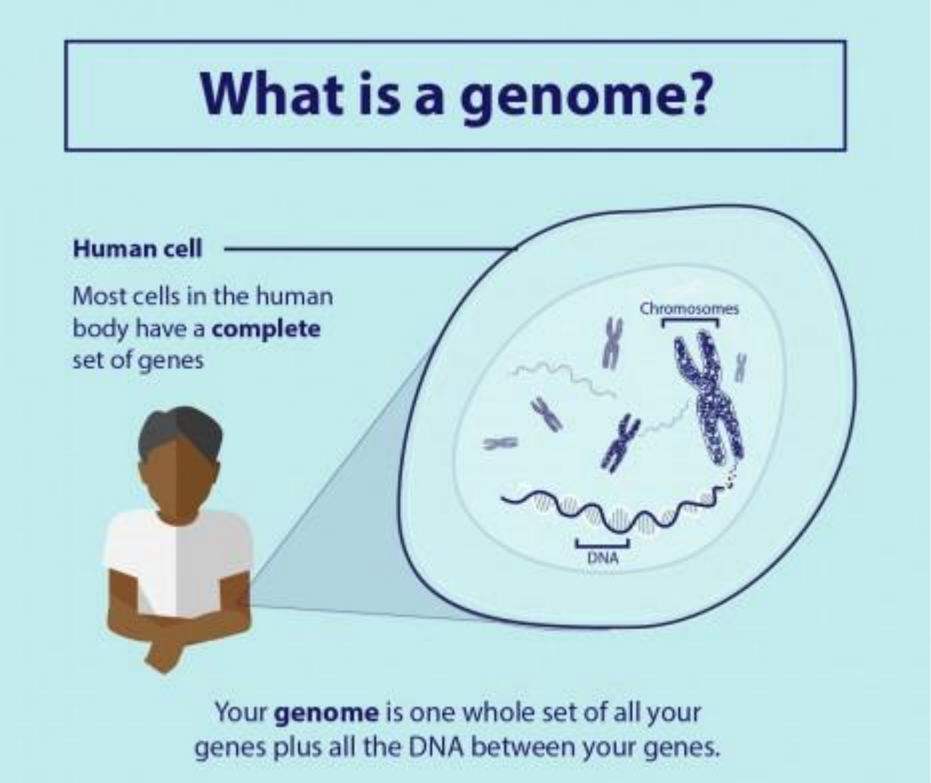

What is a genome?
Human cell
Most cells in the human body have a complete set of genes
Chromosomes
DNA
Your genome is one whole set of all your genes plus all the DNA between your genes.

Figure 7 :Genome

7.1.2 Proteomics

Proteomics is the large-scale study of proteomes. A proteome is a set of proteins produced in an organism, system, or biological context.

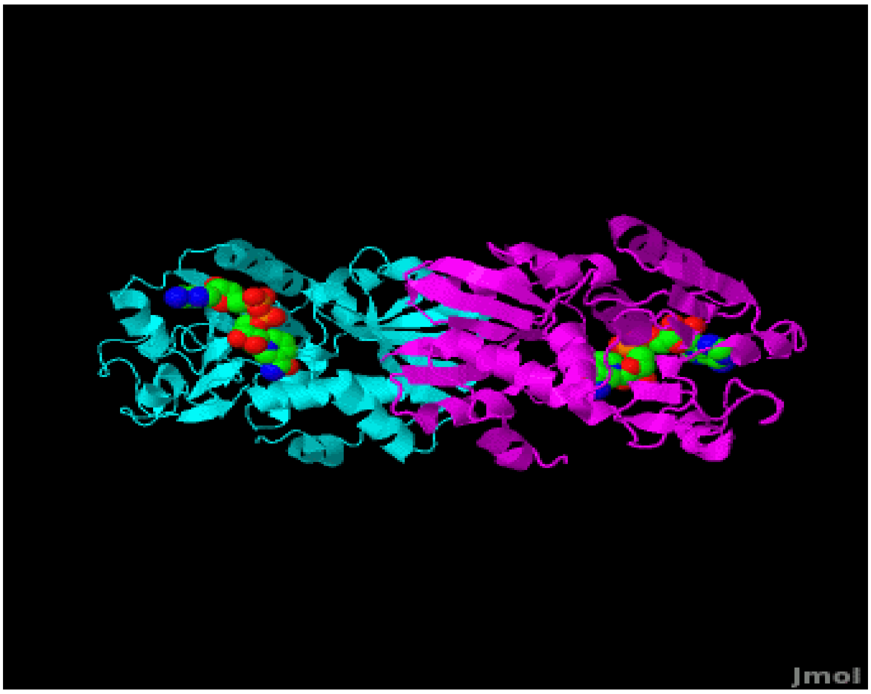


Figure 8 : Proteome

Proteins, strings of amino acids, gain much of their function from protein folding in which they conform into a three-dimensional structure. This structure is composed of a number of layers of folding, including the primary structure (i.e. the flat string of amino acids), the secondary structure (alpha helices and beta sheets), the tertiary structure, and the quaternary structure.

Protein secondary structure prediction is the main focus of this subfield as the further protein folding (tertiary and quaternary structures) are determined based on the secondary structure. Solving the true structure of a protein is an incredibly expensive and time-intensive process, furthering the need for systems that can accurately predict the structure of a protein by analyzing the amino acid sequence directly. Prior to machine learning, researchers needed to conduct this prediction manually.

The current state-of-the-art in secondary structure prediction uses a system called DeepCNF (deep convolutional neural fields) which relies on the machine learning model of artificial neural networks to achieve an accuracy of approximately 84% when tasked to classify the amino acids of a protein sequence into one of three structural classes (helix, sheet, or coil).

7.1.3 Microarrays

Microarrays, a type of lab-on-a-chip, are used for automatically collecting data about large amounts of biological material. Machine learning can aid in the analysis of this data, and it has been applied to expression pattern identification, classification, and genetic network induction.

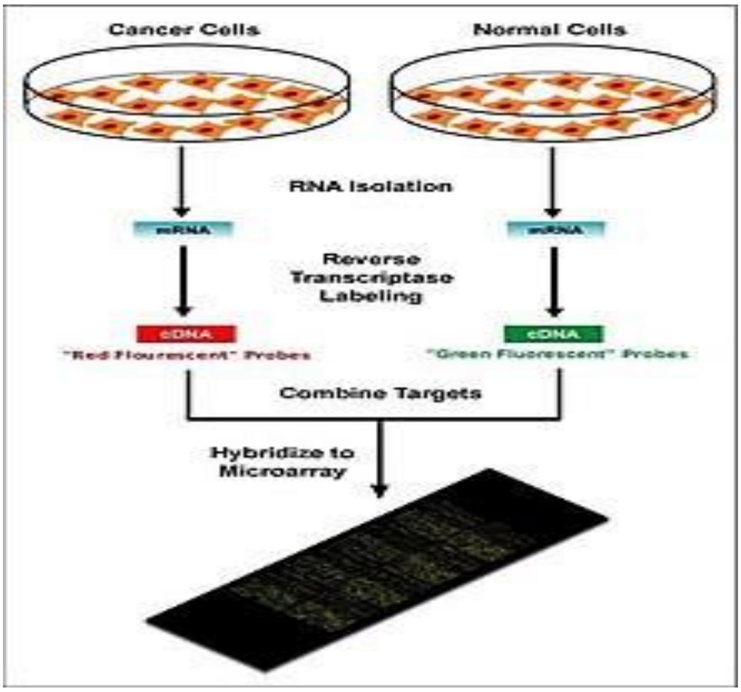


Figure 9 : DNA-microarray chip

This technology is especially useful for monitoring the expression of genes within a genome, aiding in diagnosing different types of cancer-based on which genes are expressed. One of the main problems in this field is identifying which genes are expressed based on the collected data.

Machine learning presents a potential solution to this problem as various classification methods can be used to perform this identification. The most commonly used methods are radial basis function networks, deep learning, Bayesian classification, decision trees, and random forest.

7.1.4 Systems biology

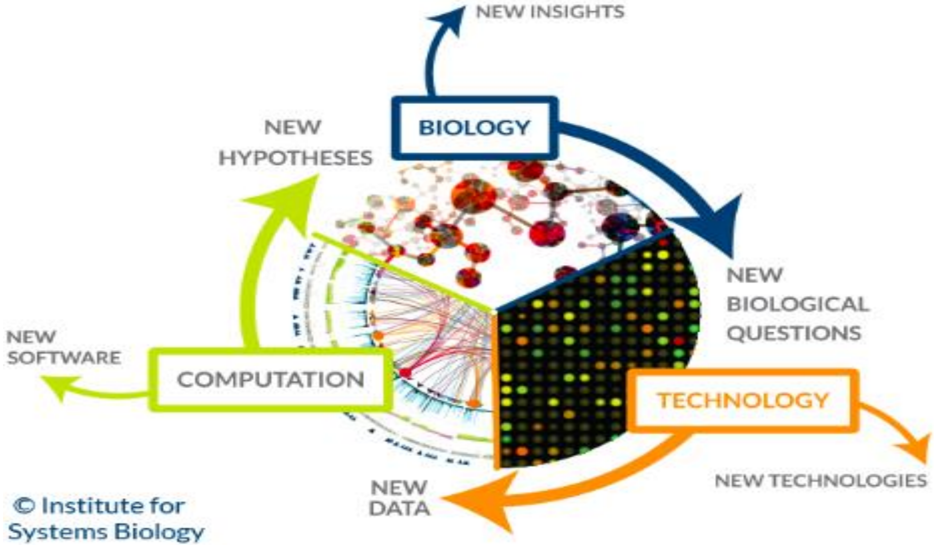


Figure 10 : System biology

Systems biology focuses on the study of the emergent behaviors from complex interactions of simple biological components in a system. Such components can include molecules such as DNA, RNA, proteins, and metabolites.

Machine learning has been used to aid in the modeling of these complex interactions in biological systems in domains such as genetic networks, signal transduction networks, and metabolic pathways. Probabilistic graphical models, a machine learning technique for determining the structure between different variables, are one of the most commonly used methods for modeling genetic networks. In addition, machine learning has been applied to systems biology problems such as identifying transcription factor binding sites using a technique known as Markov chain optimization. Genetic algorithms, machine learning techniques which are based on the natural process of evolution, have been used to model genetic networks and regulatory structures.

7.1.5 Stroke diagnosis

Machine learning methods for the analysis of neuroimaging data are used to help diagnose stroke. Three-dimensional Convolutional Neural Network(CNN) and Support Vector Machines(SVM) methods are often used.

7.1.6 Text mining

The increase in available biological publications led to the issue of the increase in difficulty in searching through and compiling all the relevant available information on a given topic across all sources. This task is known as knowledge extraction. This is necessary for biological data collection which can then, in turn, be fed into machine learning algorithms to generate new biological knowledge. Machine learning can be used for this knowledge extraction task using techniques such as Natural Language Processing(NLP) to extract useful information from human-generated reports in a database.

This technique has been applied to the search for novel drug targets, as this task requires the examination of information stored in biological databases and journals. Annotations of proteins in protein databases often do not reflect the complete known set of knowledge of each protein, so additional information must be extracted from biomedical literature. Machine learning has been applied to the automatic annotation of the function of genes and proteins, determination of the subcellular localization of a protein, analysis of DNA-expression arrays, large-scale protein interaction analysis, and molecule interaction analysis.

SECTION VIII : Molecular Classification of Cancer by Gene Expression Monitoring using Support Vector Machine(SVM)

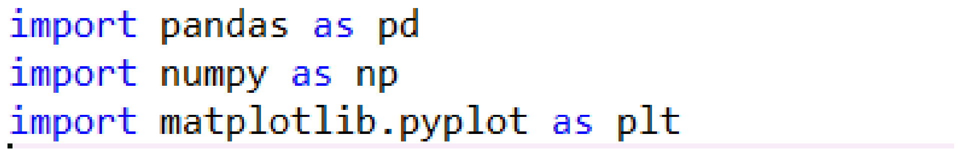
Lets us now implement the Support Vector Machine(SVM) algorithm in bioinformatics dataset and see how it works.

Although cancer classification has improved over the past 30 years, there has been no general approach for identifying new cancer classes (class discovery) or for assigning tumors to known classes (class prediction). The dataset comes from a proof-of-concept study published in 1999 by Golub et al. It showed how new cases of cancer could be classified by gene expression monitoring (via DNA microarray) and thereby provided a general approach for identifying new cancer classes and assigning tumors to known classes.The goal is to classify patients with acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) using the SVM algorithm.

The dataset can be downloaded from https://www.kaggle.com/crawford/gene-expression

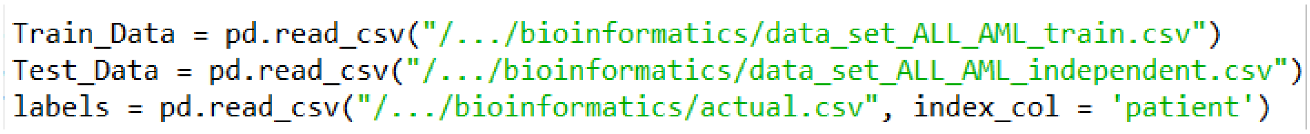
8.1 Coding

8.1.1 Loading Libraries



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

8.1.2 Load the Dataset

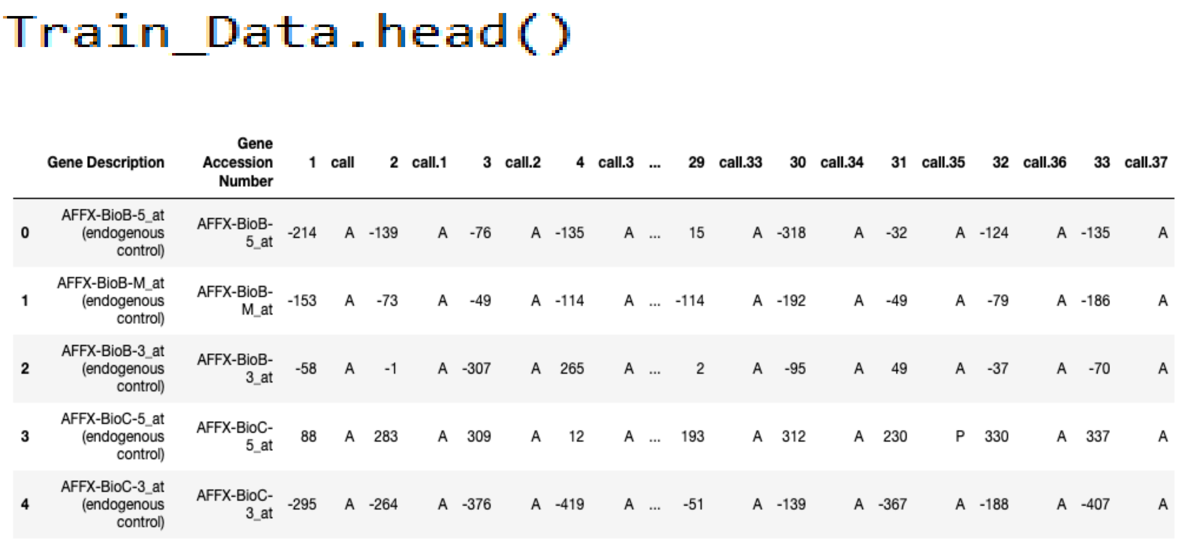


```
Train_Data = pd.read_csv("/.../bioinformatics/data_set_ALL_AML_train.csv")
Test_Data = pd.read_csv("/.../bioinformatics/data_set_ALL_AML_independent.csv")
labels = pd.read_csv("/.../bioinformatics/actual.csv", index_col = 'patient')
```

About the dataset:

1. Each row represents a different gene.
2. Columns 1 and 2 are descriptions about that gene.
3. Each numbered column is a patient in label data.
4. Each patient has 7129 gene expression values — i.e each patient has one value for each gene.
5. The training data contain gene expression values for patients 1 through 38.
6. The test data contain gene expression values for patients 39 through 72

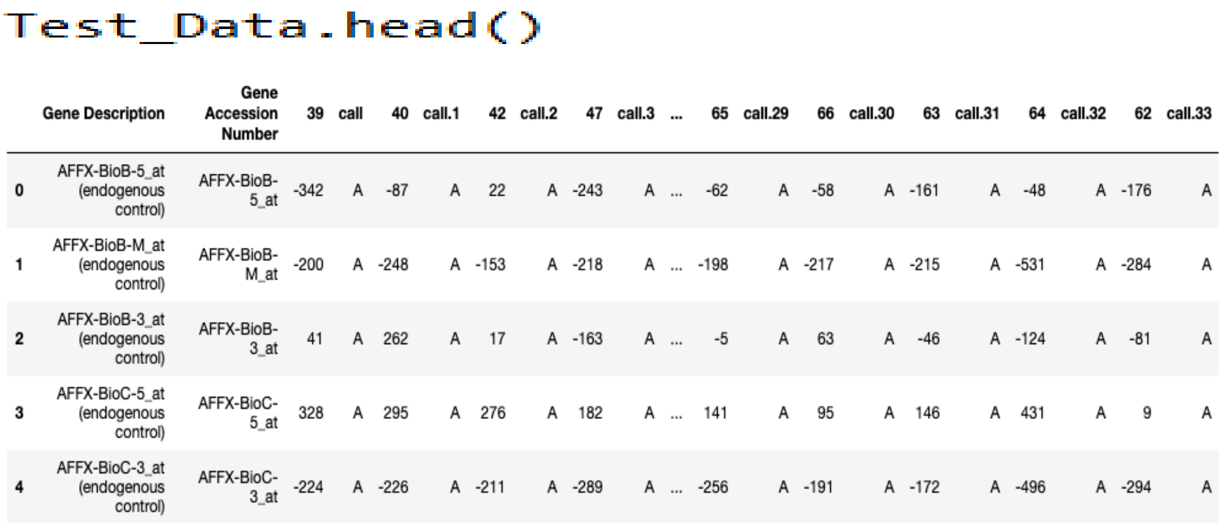
8.1.3 Training Set



```
Train_Data.head()
```

	Gene Description	Gene Accession Number	1	call	2	call.1	3	call.2	4	call.3	...	29	call.33	30	call.34	31	call.35	32	call.36	33	call.37
0	AFFX-BioB-5_at (endogenous control)	AFFX-BioB-5_at	-214	A	-139	A	-76	A	-135	A	...	15	A	-318	A	-32	A	-124	A	-135	A
1	AFFX-BioB-M_at (endogenous control)	AFFX-BioB-M_at	-153	A	-73	A	-49	A	-114	A	...	-114	A	-192	A	-49	A	-79	A	-186	A
2	AFFX-BioB-3_at (endogenous control)	AFFX-BioB-3_at	-58	A	-1	A	-307	A	265	A	...	2	A	-95	A	49	A	-37	A	-70	A
3	AFFX-BioC-5_at (endogenous control)	AFFX-BioC-5_at	88	A	283	A	309	A	12	A	...	193	A	312	A	230	P	330	A	337	A
4	AFFX-BioC-3_at (endogenous control)	AFFX-BioC-3_at	-295	A	-264	A	-376	A	-419	A	...	-51	A	-139	A	-367	A	-188	A	-407	A

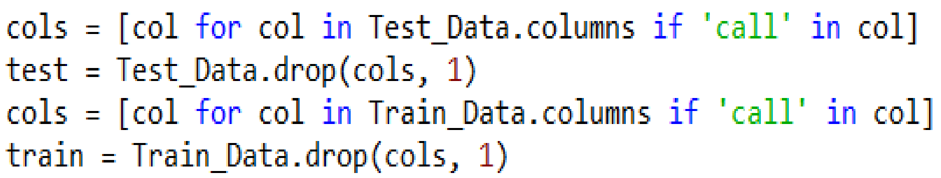
8.1.4 Testing Set



```
Test_Data.head()
```

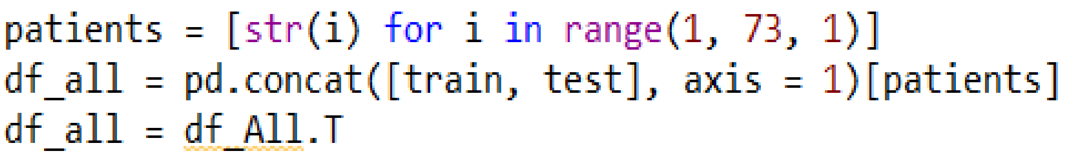
	Gene Description	Gene Accession Number	39	call	40	call.1	42	call.2	47	call.3	...	65	call.29	66	call.30	63	call.31	64	call.32	62	call.33
0	AFFX-BioB-5_at (endogenous control)	AFFX-BioB-5_at	-342	A	-87	A	22	A	-243	A	...	-62	A	-58	A	-161	A	-48	A	-176	A
1	AFFX-BioB-M_at (endogenous control)	AFFX-BioB-M_at	-200	A	-248	A	-153	A	-218	A	...	-198	A	-217	A	-215	A	-531	A	-284	A
2	AFFX-BioB-3_at (endogenous control)	AFFX-BioB-3_at	41	A	262	A	17	A	-163	A	...	-5	A	63	A	-46	A	-124	A	-81	A
3	AFFX-BioC-5_at (endogenous control)	AFFX-BioC-5_at	328	A	295	A	276	A	182	A	...	141	A	95	A	146	A	431	A	9	A
4	AFFX-BioC-3_at (endogenous control)	AFFX-BioC-3_at	-224	A	-226	A	-211	A	-289	A	...	-256	A	-191	A	-172	A	-496	A	-294	A

8.1.5 Drop column ‘call’ from both train and test data as it doesn't have any statistical relevance

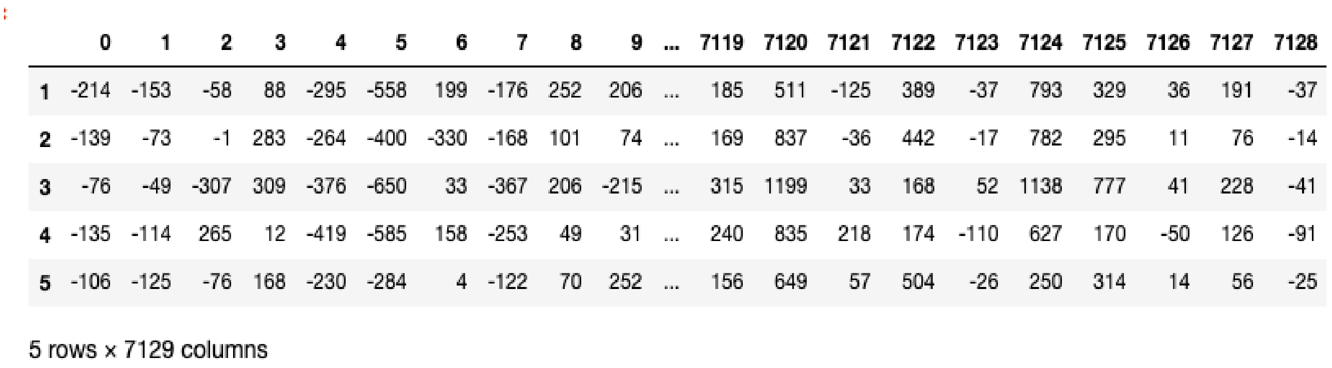


```
cols = [col for col in Test_Data.columns if 'call' in col]
test = Test_Data.drop(cols, 1)
cols = [col for col in Train_Data.columns if 'call' in col]
train = Train_Data.drop(cols, 1)
```

8.1.6 Join all the datasets and transpose the final joined data



```
patients = [str(i) for i in range(1, 73, 1)]
df_all = pd.concat([train, test], axis = 1)[patients]
df_all = df_All.T
```



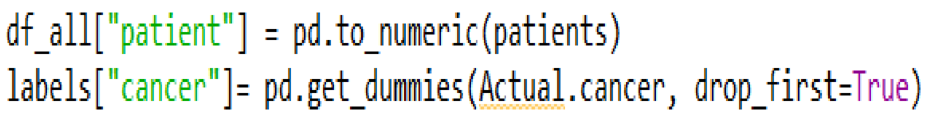
	0	1	2	3	4	5	6	7	8	9	...	7119	7120	7121	7122	7123	7124	7125	7126	7127	7128
1	-214	-153	-58	88	-295	-558	199	-176	252	206	...	185	511	-125	389	-37	793	329	36	191	-37
2	-139	-73	-1	283	-264	-400	-330	-168	101	74	...	169	837	-36	442	-17	782	295	11	76	-14
3	-76	-49	-307	309	-376	-650	33	-367	206	-215	...	315	1199	33	168	52	1138	777	41	228	-41
4	-135	-114	265	12	-419	-585	158	-253	49	31	...	240	835	218	174	-110	627	170	-50	126	-91
5	-106	-125	-76	168	-230	-284	4	-122	70	252	...	156	649	57	504	-26	250	314	14	56	-25

5 rows × 7129 columns

After transpose, the rows have been converted to columns(7129 columns/features)

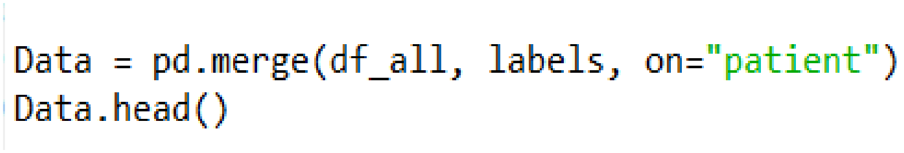
8.1.7 Convert patient column to a numeric value

create dummy variables(converts categories into numeric values) since ‘cancer’ is a cateogorical column having 2 categories(ALL, AML) .



```
df_all["patient"] = pd.to_numeric(patients)
labels["cancer"]= pd.get_dummies(Actual.cancer, drop_first=True)
```

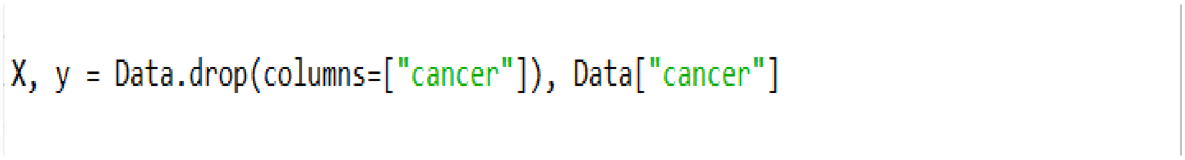
8.1.8 Join data frames df_all and labels on patient column.



```
Data = pd.merge(df_all, labels, on="patient")
Data.head()
```

8.1.9 Create two variables

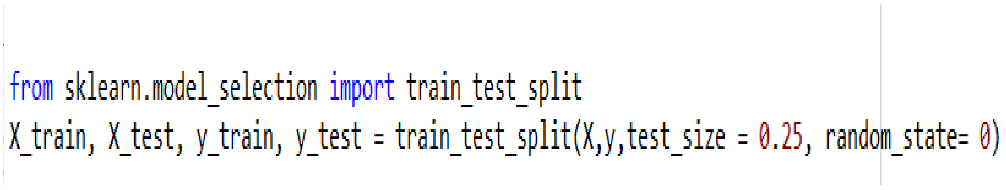
Two variables X(matrix of independent variables) and y(vector of the dependent variable)



```
X, y = Data.drop(columns=["cancer"]), Data["cancer"]
```

8.1.10 Splitting Dataset

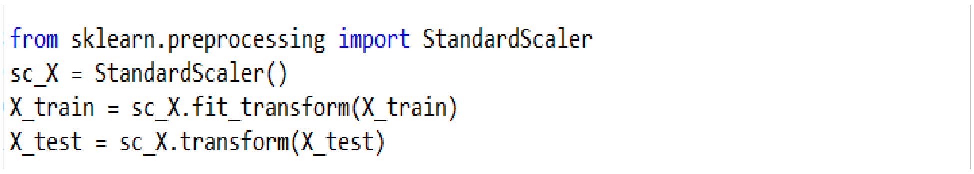
We split 75% of the data into training set while 25% of the data to test set. The test_size variable is where we actually specify the proportion of the test set.



```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.25, random_state= 0)
```

8.1.11 Normalize the Dataset

Normalize the data because if we closely look at the data the range of values of independent variables varies a lot. So when the values vary a lot in independent variables, we use feature scaling so that all the values remain in the comparable range.

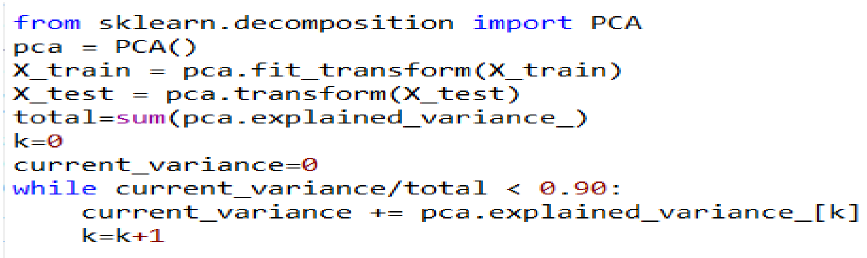


```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

The number of columns/features that we have been working with is huge. We have 72 rows and 7129 columns. Basically we need to decrease the number of features(Dimensionality Reduction) to remove the possibility of Curse of Dimensionality.

8.1.12 PCA

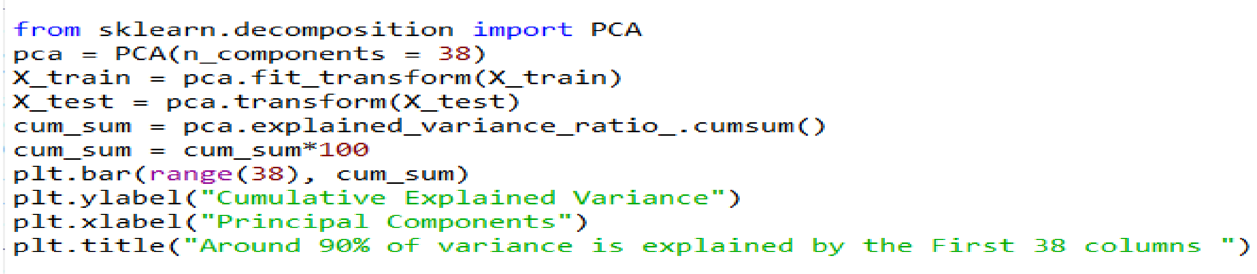
For reducing the number of dimensions/features we will use the most popular dimensionality reduction algorithm i.e. PCA(Principal Component Analysis).To perform PCA we have to choose the number of features/dimensions that we want in our data.



```
from sklearn.decomposition import PCA
pca = PCA()
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
total=sum(pca.explained_variance_)
k=0
current_variance=0
while current_variance/total < 0.90:
    current_variance += pca.explained_variance_[k]
    k=k+1
```

The above code gives k=38.

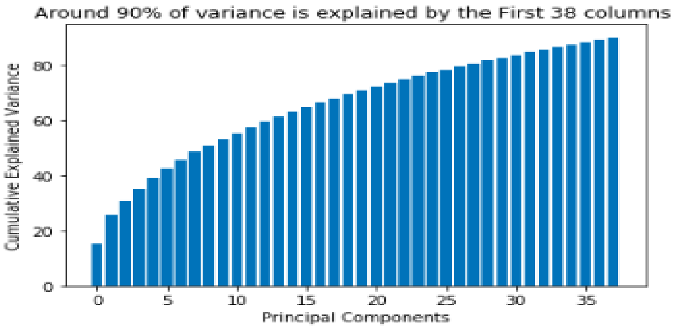
Now let us take k=38 and apply PCA on our independent variables.



```
from sklearn.decomposition import PCA
pca = PCA(n_components = 38)
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
cum_sum = pca.explained_variance_ratio_.cumsum()
cum_sum = cum_sum*100
plt.bar(range(38), cum_sum)
plt.ylabel("Cumulative Explained Variance")
plt.xlabel("Principal Components")
plt.title("Around 90% of variance is explained by the First 38 columns ")
```

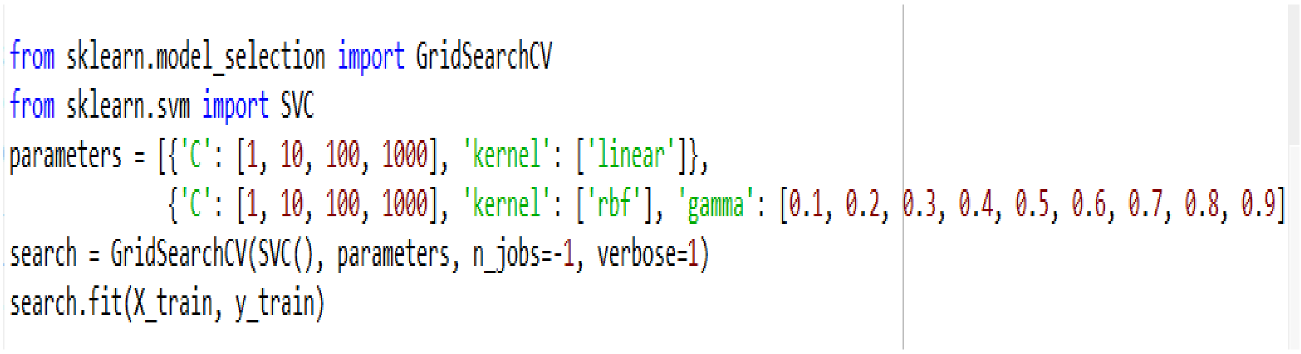
Note:- PCA can lead to a reduction in model performance on datasets with no or low feature correlation or does not meet the assumptions of linearity.

```
Text(0.5, 1.0, 'Around 90% of variance is explained by the First 38 columns ')
```



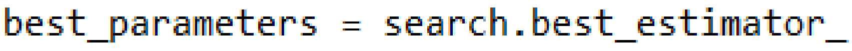
8.1.13 Fit Dataset into the Support Vector Machine(SVM) algorithm

Before doing this fit we will perform Hyperparameter optimization.Hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters are learned.We will use GridSearchCV from sklearn for choosing the best hyperparameters.



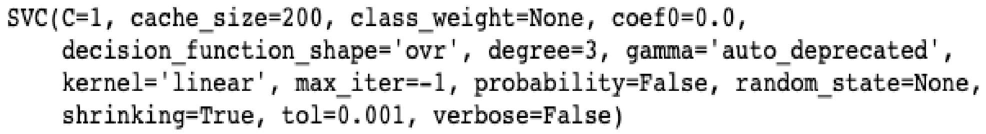
```
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC
parameters = [{'C': [1, 10, 100, 1000], 'kernel': ['linear']},
              {'C': [1, 10, 100, 1000], 'kernel': ['rbf'], 'gamma': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
search = GridSearchCV(SVC(), parameters, n_jobs=-1, verbose=1)
search.fit(X_train, y_train)
```

Now check what are the best parameters for our SVM algorithm



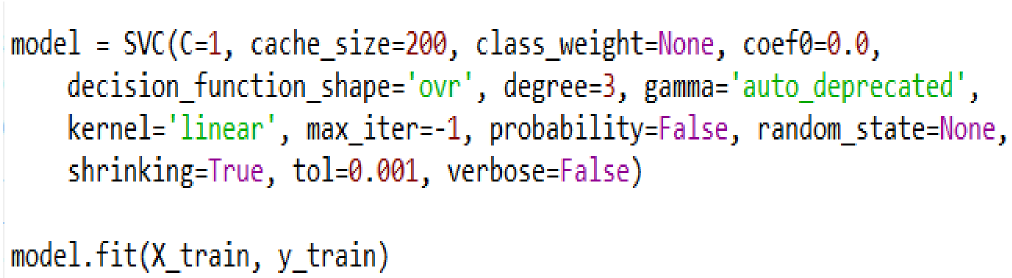
```
best_parameters = search.best_estimator_
```

Best Hyperparameters



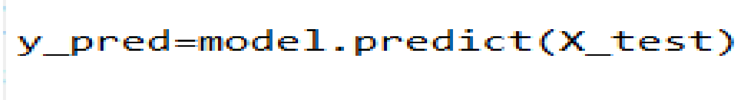
```
SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='linear', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

8.2 Train our SVM classification model



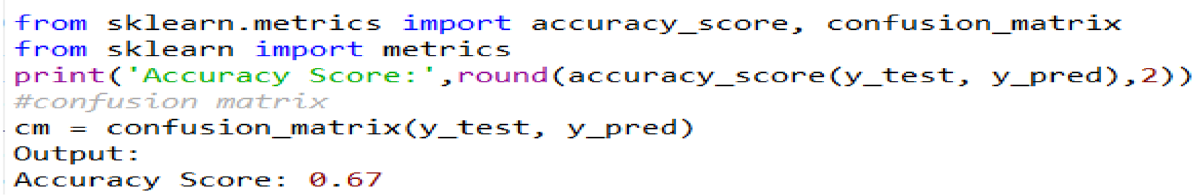
```
model = SVC(C=1, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='linear', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)

model.fit(X_train, y_train)
```

8.2.1 Predicted Values



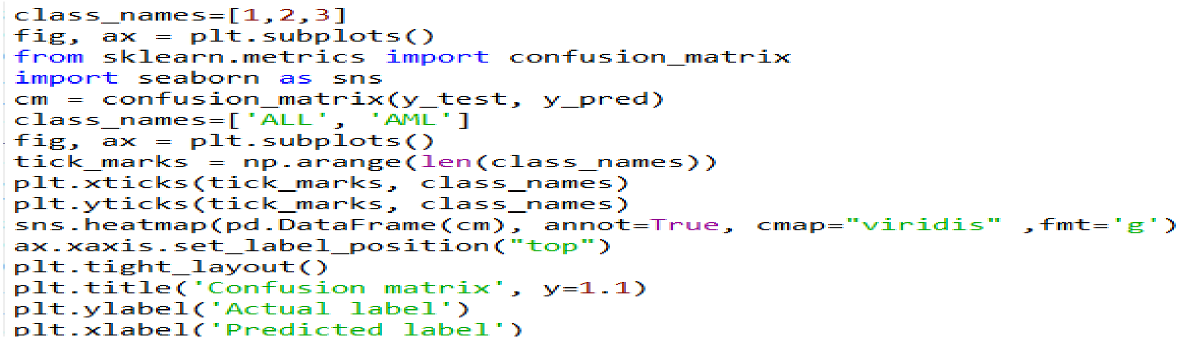
```
y_pred=model.predict(X_test)
```

8.2.2 Evaluate Model Performance

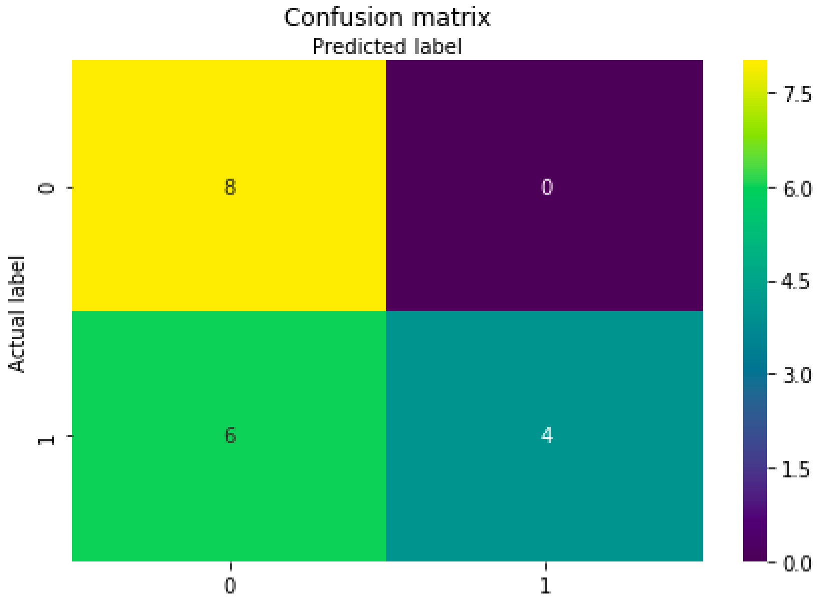


```
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn import metrics
print('Accuracy Score:',round(accuracy_score(y_test, y_pred),2))
#confusion matrix
cm = confusion_matrix(y_test, y_pred)
Output:
Accuracy Score: 0.67
```

8.2.3 Confusion Matrix and Visualize it using Heatmap



```
class_names=[1,2,3]
fig, ax = plt.subplots()
from sklearn.metrics import confusion_matrix
import seaborn as sns
cm = confusion_matrix(y_test, y_pred)
class_names=['ALL', 'AML']
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
sns.heatmap(pd.DataFrame(cm), annot=True, cmap="viridis" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```



Well, this example goes to show that if you just predict that every patient has AML, you’ll be correct more often than wrong.So our SVM classification model predicted the cancer patients with 67% accuracy which is of course not that good. What you can do is try different classifiers like Random forest, K-NN, Gradient Boosting, xgboost etc and compare the accuracies for each model.

8.3 Conclusion

We have seen how a classification ML algorithm can be used to predict cancer in a patient.Ultimately I think for machine learning to really flourish, it's going to come down to better bioinformatics data. Heath and bioinformatics data right now have pretty poor statistical power. Either they usually have poor signal (genomics), high noise/bias (electronic health records), or smallish sample sizes.