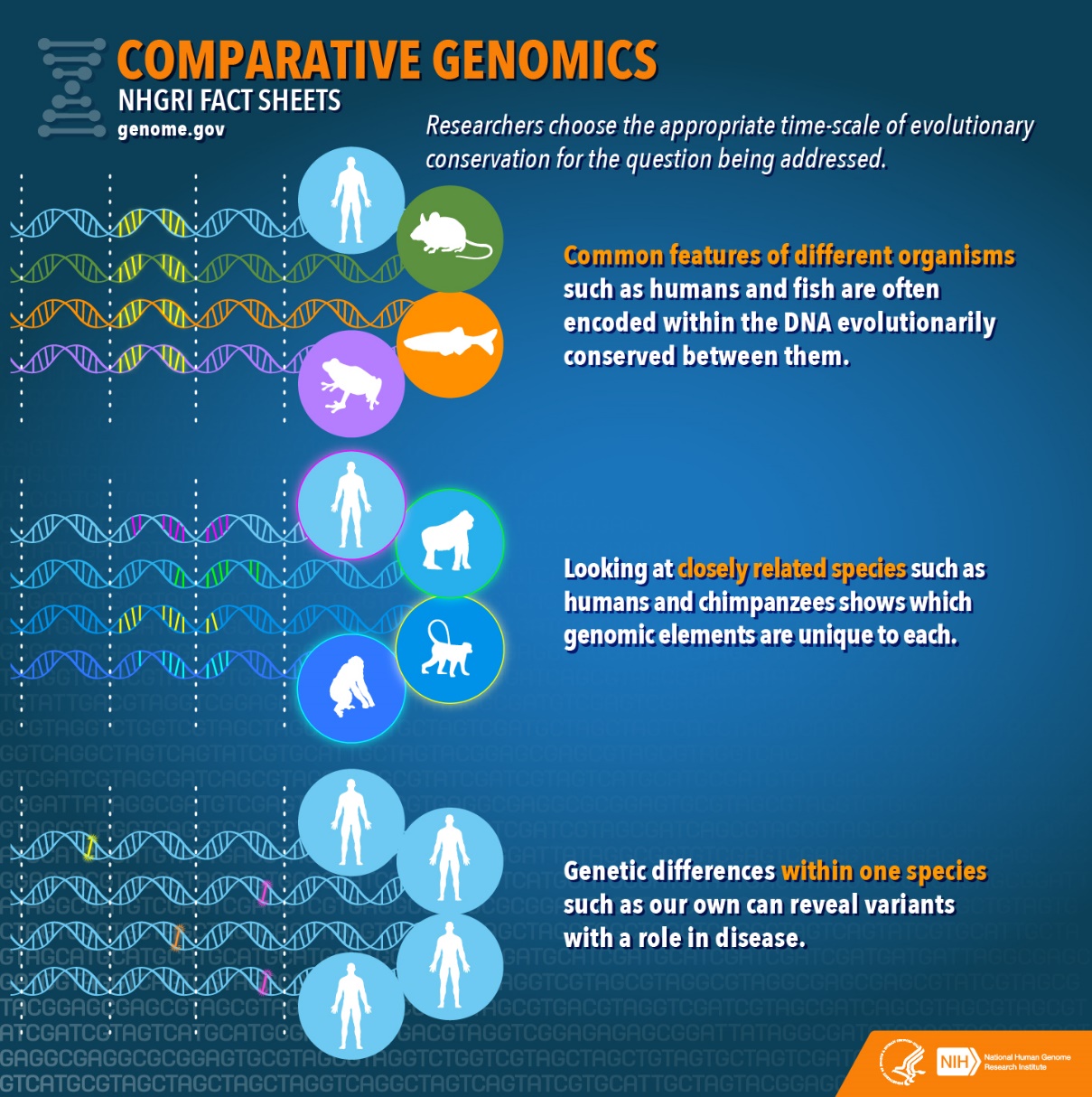
An Application for

Comparative Genetics using Biological Databases



**Team Members**:

Deigant

Ankith Prabhu

**Abstract:**

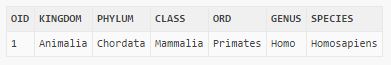
The research in the field of biology is growing at an exponential rate which is much more than the rate at which it is being processed or understood. The data obtained is not stored in an efficient manner and convenient manner on a large scale which greatly reduces the reach of these Biological samples to people throughout the world. Even though computer applications have been built to analyse this data, it would take lots of effort fetching individual data and then feeding it to the application for analysis. There is also the issue of redundancy in biological databases as biological databases tend to be complex to fit a simple definition of redundancy. Like two closely related organisms may have same locus. This restricts the concept like deletion due to the necessity of redundancy for correspondence. This is what our project aims to solve, to store and analyse genetic data. The genomic data of different organisms extracted from the biological samples taken from those organisms is stored in the database. The database is of a hierarchical tree structure with the root containing the taxonomical details about the organisms, which relates to a sequence of databases finally giving the genetic data of the organism like its DNA sequence. This is further processed using the Basic Alignment Search Tool to give the local and global alignment among the DNA samples. To conclude, our project allows efficient storage of genomic data and process it using alignment tools.

**Problem Statement:**

The aim of our project is to store and analyse genetic data of organisms. In order to obtain the data we have referred NCBI, which is a US government-funded national resource for molecular biology information. The dna and protein sequences of organisms are stored in FASTA format which we have stored in specific files with the file references in the database tables. The rest of the descriptive attributes like functions and taxonomical details were obtained through online resources. For alignment we have used API’s by bio java which use specific algorithms for local and global alignment.

**Sample Data:**

--For the keratin sample of a human being.



insert into organism values('1', 'Animalia', 'Chordata', 'Mammalia', 'Primates', 'Homo', 'Homosapiens');



insert into multicell values('1', 'Africa', '70yrs', 'Human', 'Heterotroph');



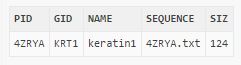
insert into sample values('2', '2', '28/May/2011', '2');



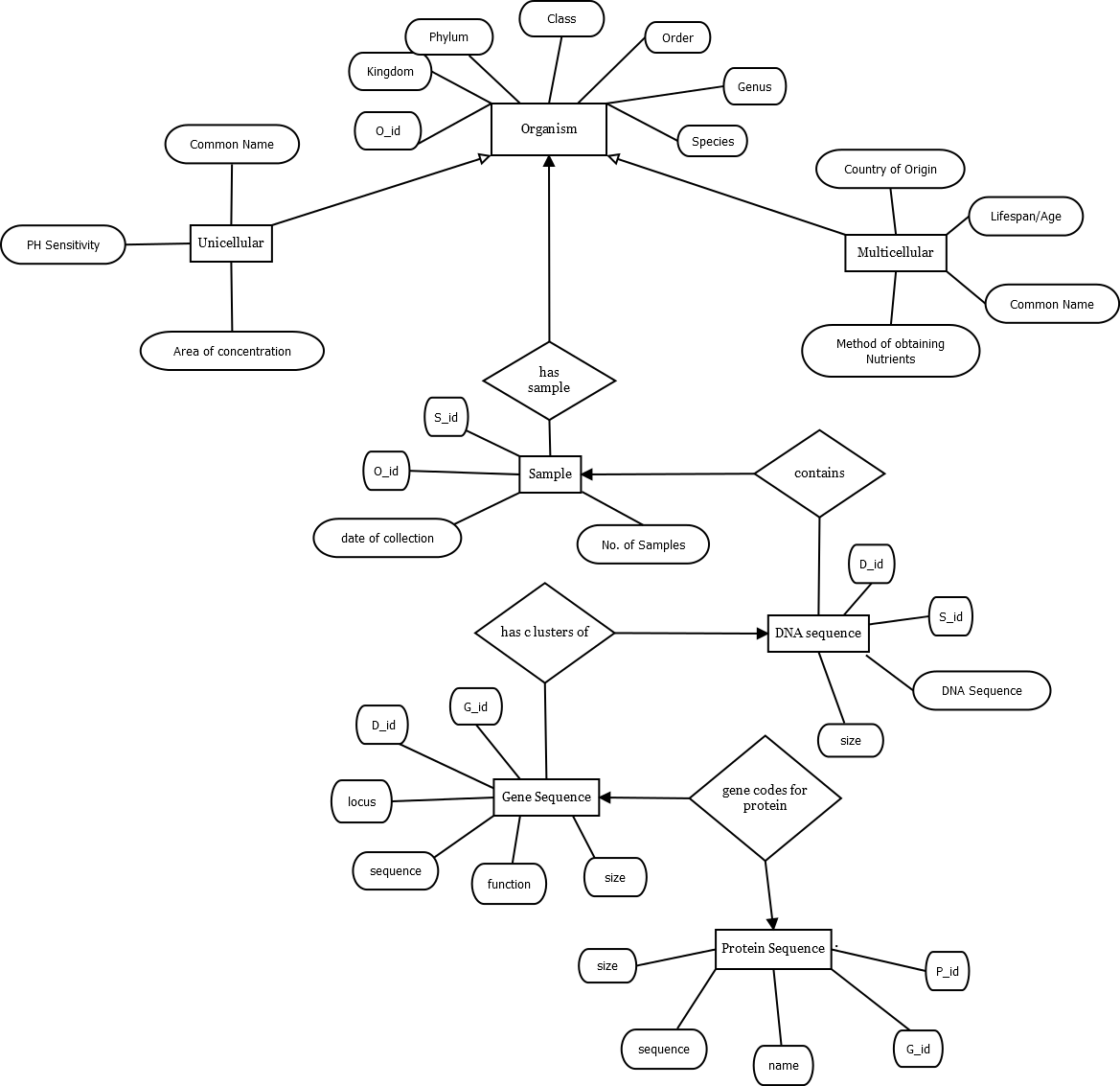
insert into dna\_seq values('cr12', '1', 'CR12.txt', '6806');



insert into gene\_seq values ('KRT1', 'cr12' , '12q13.13' , 'krt1.txt' , 'forms keratin protein', '5673');



insert into pro\_seq values('4ZRYA', 'KRT1', 'keratin1', '4ZRYA.txt', '124');

**ER Diagram:**

**Database Tables:**

* **Organism** (Oid, Kingdom, Phylum, Class, Ord, Genus, Species).
* **Multicell** (Oid, Country, Lifespan, Common, Food);
* **Unicell** (Oid, Common, Ph, Area);
* **Sample** (Sid, Oid, doc, num);
* **DNA\_seq** (Did, Sid, Sequence, Siz);
* **Gene\_seq** (Gid, Did, Locus, Sequence, Func, Siz);
* **Pro\_seq** (Pid, Gid, Name, Sequence, Siz);

**Create table queries:**

create table organism(oid varchar(25) primary key, kingdom varchar(25), phylum varchar(25), class varchar(25), ord varchar(25), genus varchar(25), species varchar(25));

create table multicell(oid varchar(25) primary key, country varchar(25), lifespan varchar(25), common varchar(25), food varchar(25), foreign key(oid) references organism on delete cascade);

create table uni(oid varchar(25) primary key, common varchar(25), ph varchar(25), area varchar(25), foreign key(oid) references organism on delete cascade);

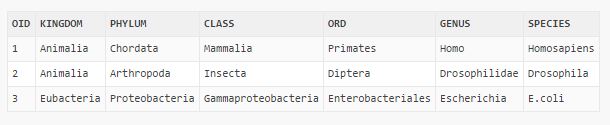
create table sample(sid varchar(25) primary key, oid varchar(25), doc varchar(25), num varchar(25), foreign key(oid) references organism(oid) on delete cascade);

create table dna\_seq(did varchar(25) primary key, sid varchar(25), sequence varchar(25), siz varchar(25), foreign key(sid) references sample(sid) on delete cascade);

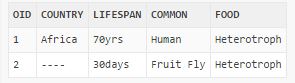
create table gene\_seq(gid varchar(25) primary key, did varchar(25), locus varchar(25), sequence varchar(25), func varchar(25), siz varchar(10), foreign key(did) references dna\_seq(did) on delete cascade);

create table pro\_seq(pid varchar(25) primary key, gid varchar(25), name varchar(25), sequence varchar(25), siz varchar(25), foreign key(gid) references gene\_seq(gid) on delete cascade);

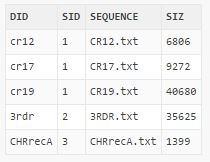
**Relational Tables:**

1) Organism****

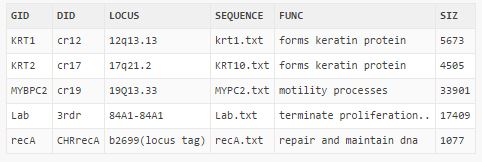
2) Multicell 3) Uni



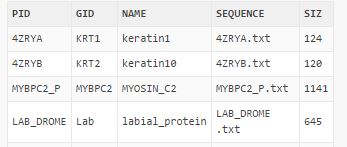
3) Sample 4) dna\_seq



5) gene\_seq

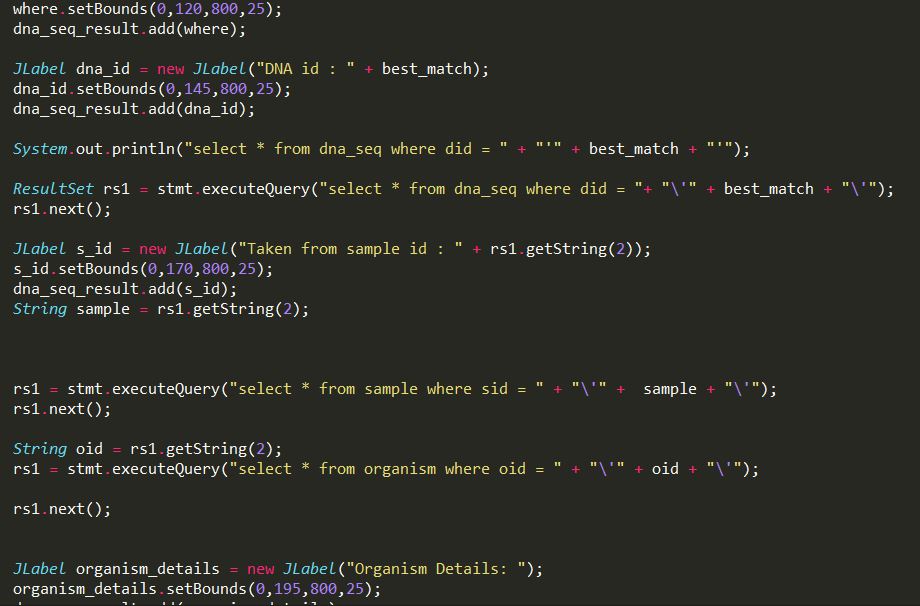


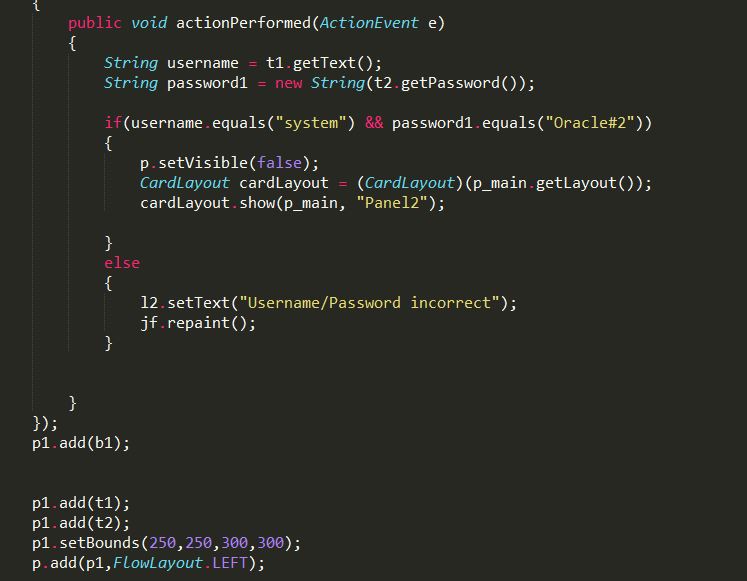
6) pro\_seq:



**Code Snapshots:**







**Code for Functional Design:**

**PsuedoCode:**

//step1 load the driver class

//step2 create the connection object

//step3 create the statement object

//step4 execute query

//step5 close the connection object

**Implementation:**

*try{*

*Class.forName("oracle.jdbc.driver.OracleDriver");*

*//step2 create the connection object*

*Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521/xe","system","Oracle#2");*

*//step3 create the statement object*

*Statement stmt=con.createStatement();*

*double largest\_score = 0;*

*String best\_match = "";*

*String best\_file = "";*

*int cat = 0;*

*int flag = 0;*

*//step4 execute query*

*ResultSet rs=stmt.executeQuery("select \* from DNA\_seq");*

*while(rs.next())*

*{*

*String target = "";*

*try*

*{*

*BufferedReader br = new BufferedReader(new FileReader("data/" + rs.getString(3)));*

*String s;*

*while((s = br.readLine())!=null)*

*{*

*target = target+s;*

*}*

*}*

*catch(Exception e)*

*{*

*e.printStackTrace();*

*}*

**Future Scope:**

The application that we propose to develop aims to combine the tasks of storing and analysing Biological data (Gene sequences, Protein and mRNA sequences) of different species and can be used as a tool to help in understanding the new biological data that is becoming available to us every day. A database of the biological data will be maintained which would enable us to store the available data in an efficient manner and also help us to store the relationships between the various types of data (for example the relationship between a protein sequence and the region in the DNA that codes for that sequence). All forms of detail, from the source from which the sequence was extracted to the date on which it was obtained, will be stored in the Database. Apart from the database the application will also provide tools to process the data and perform operations like searching the database for a given sequence, aligning two given sequences (global or local alignment) and finding out the closeness of the sequences depending on the results of the alignment etc.

**References:**

* <https://blast.ncbi.nlm.nih.gov/Blast.cgi>
* <https://github.com/biojava/biojava>
* <http://sepwww.stanford.edu/public/docs/sep112/bob2/paper_html/node3.html>