Volume 1, No.1, March – April 2012
International Journal of Science and Applied Information Technology
Available Online at www.warse.ijatcse.current

Design and Development of Software for managing data for Protein-Ligand Docking Studies

Supriya Chakraborty,Subrata Sinha,Surabhi Johari , Pallavi Dutta*
Center for Bioinformatics Studies
Dibrugarh University
Dibrugarh-786004, Assam
pallavidutta@rocketmail*

ABSTRACT

The recent advancement in computer aided drug designing different databases are required to retrieve information about a particular protein-ligand docking. The information on protein-ligand and diseases cannot be retrieved from one database alone. The study deals in designing and development of a publicly available protein-ligand database where the information on protein, ligands, protein ligand docking and the diseases it prevents can be stored. The database was designed by using DBMS software MS-SQL SERVER 7.0 as back end and Visual Basic as frond end. All the information and related data were arranged in a systemic manner in the database.

1

Keywords: Protein, Docking, Diseases, Database, Ligand

1. INTRODUCTION

Molecular docking is defined as a process in which small molecule binds to a biological target involving efficient sampling of possible poses in the specified binding pocket in order to identify the optimal binding geometry [6]. This is basically known as computational docking [6].

In the field of Molecular modeling, molecular docking is a method which predicts the preferred orientation of one molecule to a second when bound to each other to form a stable complex [4]. In order to carry out docking different databases are required to retrieve information about a protein, ligand and docking.

A database is an organized collection of information arranged and presented to serve an assigned purpose [3]. Protein database like Protein Data Bank (PDB) [8], Ligand databases like Chem database[2] and KEGG Ligand database[5] etc are available but protein ligand docking database which could store information about proteins, ligands, docking information and diseases it caused is yet to develop.

Our study deals with the design of relational database for all the information about ligand, protein, docking and diseases cause. Several protein-ligand docking software applications are available, such as AutoDock [1] and Dock [10]. There are also web service (Molecular Docking Server, Swiss Dock) that calculate the site, geometry and energy of small molecules interacting with proteins.

As there are a vast number of proteins and various types of ligand used as drug to prevent various diseases. It is difficult to store all the information. So databases are required to store all these information.

2. MATERIALS AND METHODS

2.1 Materials

To design the database we used SQL 7.0 SERVER, an RDBMS package which follows Standard Query Language specification [9].It helps us to design and maintain a database. Visual Basic 6.0 was used as front end. VB is an Integrated Development Environment in which one can develop, run, and test and debug applications [7]. It is known as RAD (Rapid Application Development) [7]. For connectivity, we have used ADO, which is a Microsoft Technology for Database connectivity and SQLOLEDB is used as Provider.

2.2 Methods

The database design starts from identifying entities and relationship among entities followed by designing ER Diagram, then mapping ER Model to the physical database. For designing the software, Classic Life Cycle [7] Model of SDLC is followed. In the design phase we have constructed the DFD and accordingly the software is constructed.

About Entity Relationships

We have used SQL 7.0 for creating a database. The database design process is divided into few steps mentioned below:

The entity-relationship (ER) data model allows us to describe the data involved in a real-world enterprise in terms of objects and their relationships and is widely used to develop an initial database design [3,7]. The ER model is important primarily for its role in database design. It provides useful concepts that allow us to move from an informal description of what users want from their database to a more detailed and precise, description that can be implemented in DBMS. The ER diagram is an approximate description of the data, constructed through a subjective evaluation of the information collected during requirements analysis [3].

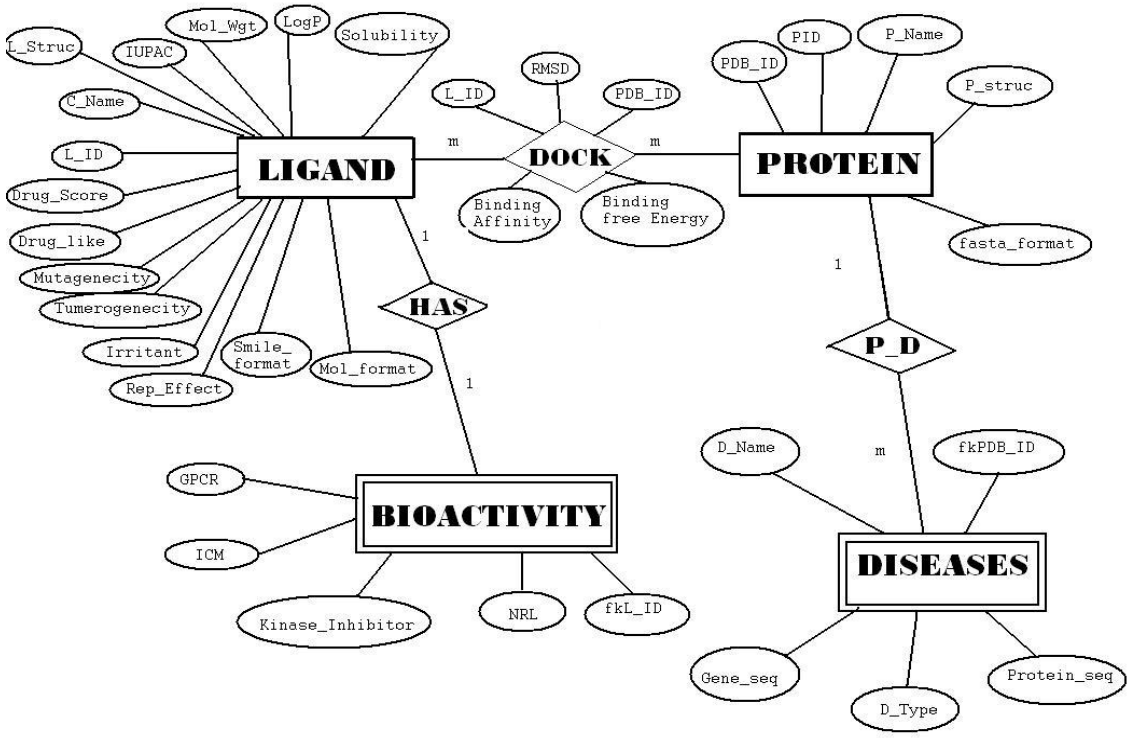

Figure 1: Entity-relationship diagram

To design a database which conforms to an ER diagram can be represented by a set of tables. For each entity-entity relationship there is a unique table which is assigned the name of corresponding entity set [7]. These relationships may exist as One-to-Many, One-to-One and Many-to-Many where primary key of one entity moves to the other entity and becomes the foreign key of that entity or both the primary keys of the entities go to the relationship and form the foreign key as found in One-to-One and Many-to-Many relationships. An entity usually has an attribute whose values are distinct for each individual entity in the collection and is known as key attribute .The ER diagram of the present study is shown in Figure 1.

a. Mapping of Binary 1: N relationship Types:

For each regular binary 1: N relation type R, identify the relation S that represents the participating entity type at N-side relationship type. Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R; we do this because each entity instance on the N-side is related to most one entity instance on the 1-side of the relationship type.

b. Mapping of Binary M:N Relationship Types

For each M:N relationship type R, create a new relation S to represent R. Include as foreign key attribute in S the primary keys of the relation that represents the participating entity types; their combination will from the primary key of S. The detailed process may not include for length constraint of the paper.

3. RESULT

The result of our work is the database created using SQL 7.0 as well as the software product which will be used to efficiently access the data and maintain the database .The ER diagram of the database and schema of the database which is generated from SQL7.0 Manager is shown below (Figure 2).

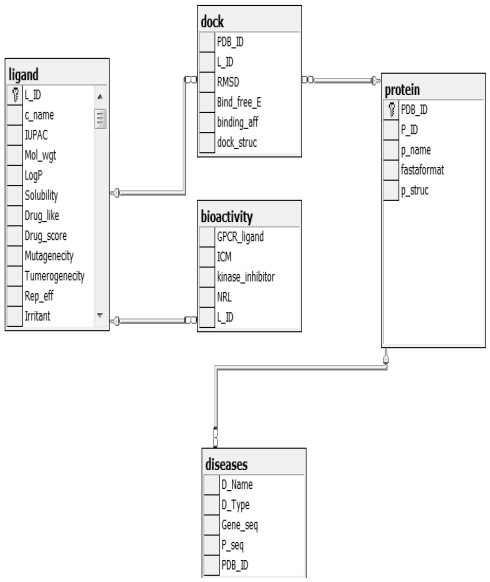

Figure 2: SQL Server 7.0 generated Schema of Database

After designing the database we have drawn the DFD (Data Flow Diagram) Figure.4. DFD shows the flow of data from one process to another process. The DFD shown is the blueprint of the software design. The first context diagram or 0 level DFD has been drawn context diagram is nothing but the bird's view of overall software operation which shows the interaction between the external entity and the main system.

The overall working of the software can be represented with a Data flow Diagram (Figure.3) which has the following process:

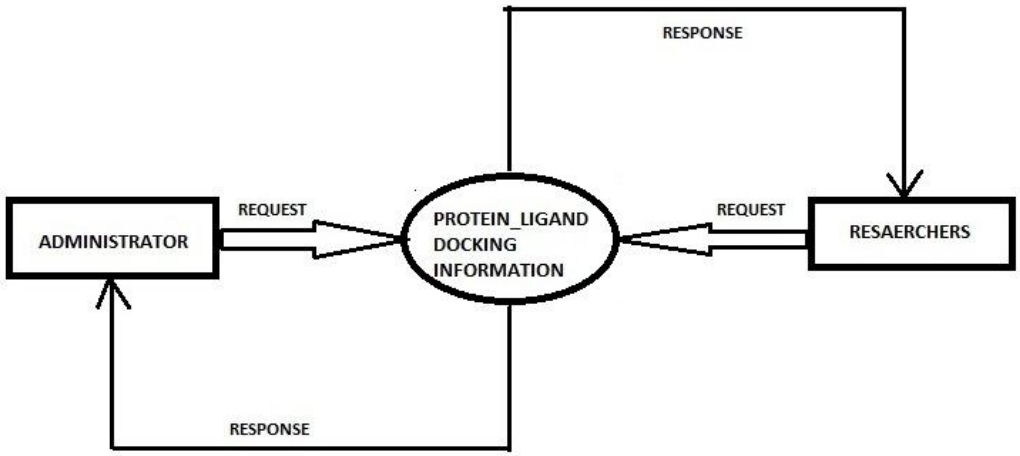

Figure 3: Zero Level DFD (Context Diagram)

In level one DFD we have seven processes as described below-

Process 1.0: Disease process

This process is responsible for save, delete and update of disease records.

Process 2.0: Protein process

This process is responsible for save, delete and update of protein type and other information on protein. The administrator will chooses a protein and enters PDB Id and

all other information on protein and it will be stored in the protein table.

Process 3.0: Ligand process

This process is responsible for save, delete and update of ligand type and other information on ligand molecule.

Process 4.0: Docking Process

This process depends on the protein and ligand binding with it. The administrator will choose a protein name and different ligand binding with it, their binding affinity and other properties of docking as a whole the record is stored in the docking table. This process is responsible for save, delete, update of docking information.

Process 5.0: Bioactivity

This process is dependent on the ligand table. The administrator will choose the ligand ID and will enter its GPCR ligand, NRL, kinase inhibitor, ICM and the record of ligand will be stored in the bioactivity table.

Process 6.0: Photo process

This process depends on the protein table, ligand table as well as the protein-ligand docking table. The administrator will choose a Protein name or ID, ligand name or ID and will load the photos of proteins, ligands and PLD photos which will be stored in the Photo table. It is also concerned with saving, deleting, updating the photos.

Process 7.0: Researchers Process

The 7.0 is a search process. The researchers will get the whole information of protein, ligand, protein-ligand docking through search.

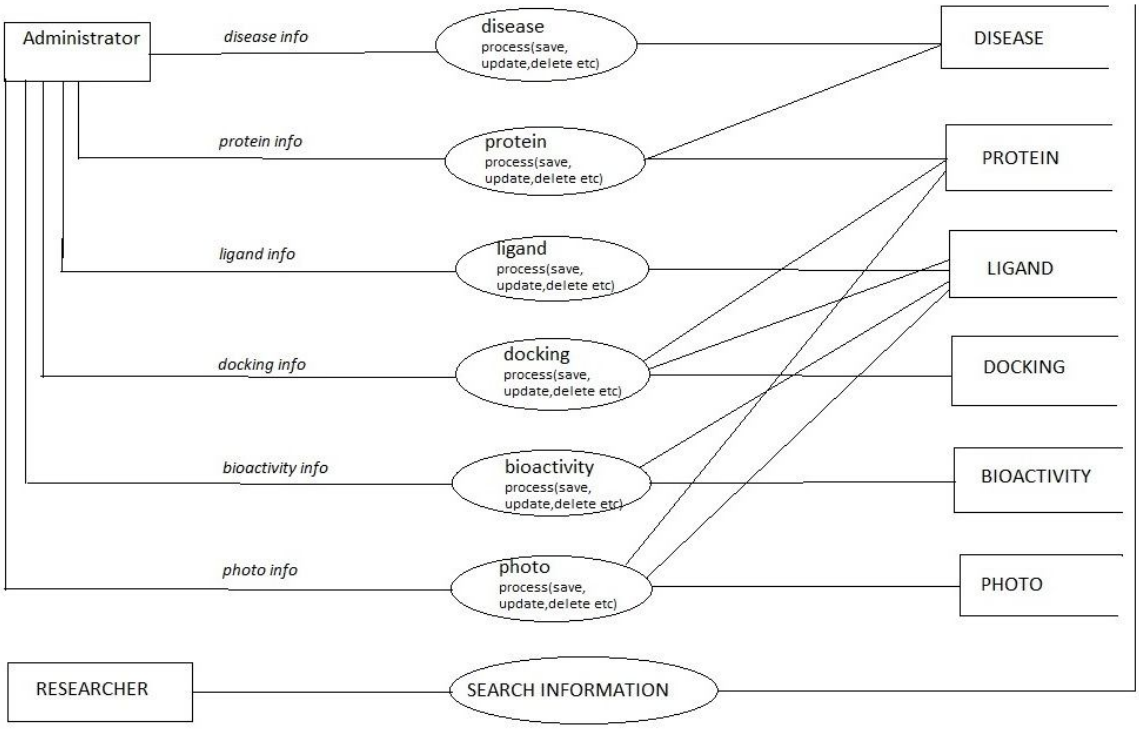

Figure. 4: Level 1 DFD for Protein-Ligand Docking Information System

After the designing of database the further work is processed on Visual Basics 6.0 for the development of the Protein-Ligand Docking (PLD) database as Protein-Ligand Docking software. The various forms and coding has been created for different tables. The main MDI form of the PLD software is shown in Figure 5.

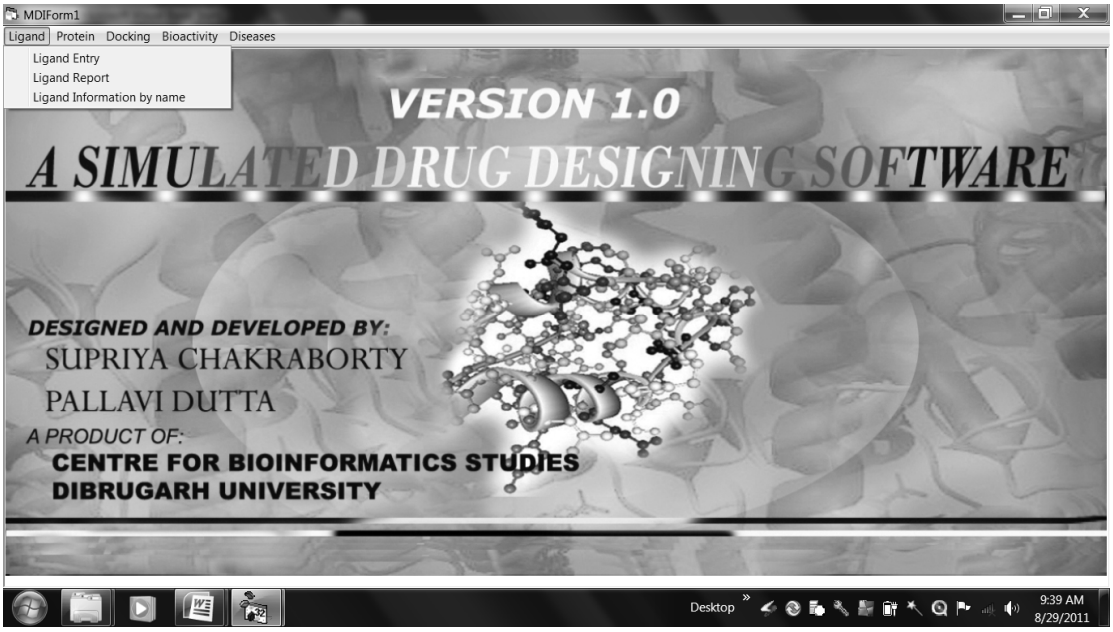

Figure.5 : Snapshot of Protein-Ligand Docking Software

4. CONCLUSION

The designed software is a standalone desktop based application, which runs on windows environment only. The software gives details about all relevant information about protein-ligand docking. The system designed can be used for research purposes. As the system runs only in Windows environment so we are trying to make a web based version of the designed software using Visual basic 6.0. That will give our application platform neutrality without changing any design issue of the database. In the present work we have only concentrated on the relational aspect of the database and the construction of the database. The further incorporation of data can be done through the software interfaces. Also we are incorporating more information and features in our database and software respectively. As a whole we can say that the software which is a standalone version can be helpful to students, researchers and educators of the same field.

REFERENCES

1. Mohan V, Gibbs C Alan, Cummings D, Maxwell et al. **Docking : Success and Challenges**, Current Pharmaceutical Design, 2005, 11, pp. 323-333.
2. Gaba Monica, Gaba Punam et al., **An Overview of Molecular Docking,** International Journal of Drug Dev. &Rev., Vo,.2(2), pp. 219-231, 2010.
3. Elmasri R, Navathe S. B. **Fundamentals of Database Systems**, 5th edition, Pearson Education, McGraw-Hill, pp. 228-231.
4. [Web Site] **Protein Data bank** , www. pdb.org
5. [Web site] **ChemSpider Database**, www.chemspider .com
6. [Web site] **KEGG**, www.genone.jp/kegg/
7. [Web site] **AUTODOCK**, www.autodock.scripps.edu/
8. [Web site] **UCSF DOCK**, www.dock.compbio.ucsf.edu/
9. Ramakrishnan R, and Gehrke J. **Database Management Systems**, 3rd edition, MCGraw-Hill, pp. 25-26, 2003.
10. Petroutsos E. **Mastering Visual Basic 6**, Willey India Edition, pp. 134-136.