# A Novel Big Data Storage Model for Protein-Protein Interaction and Gene-Protein Associations

M. Atif Sarwar
Department of Computer Science
COMSATS Institute of Information Technology
Sahiwal, Pakistan

Hira Yaseen
Department of Computer Science
COMSATS Institute of Information Technology
Sahiwal, Pakistan

Javed Ferzund
Department of Computer Science
COMSATS Institute of Information Technology
Sahiwal, Pakistan

Hina Farooq
Department of Computer Science
COMSATS Institute of Information Technology
Sahiwal, Pakistan

Azka Mahmood
Department of Computer Science
COMSATS Institute of Information Technology
Lahore, Pakistan

***Abstract*—NGS (Next Generation Sequencing) technology has resulted in huge amount of proteomics data that exists in the form of interactions (protein-protein, gene-protein, and gene-disease). ETL (Extraction, Transformation, and Loading) techniques are very useful for Databases. Existing Rational Databases are not unified and having SQL (Structured Query Language). Proteomics data requires improvement for Integration of different Data sources. With the usage of NoSQL (not only SQL), improve the efficiency and performance. For this, a novel based unified model has been designed for protein interactions data (P-P, G-G, and G-D) by using Apache HBase to evaluate given the model, different case studies have been used.**

***Keywords—Hadoop; HBase; Big Data; Apache Drill; Protein-Protein Interaction; Gene-Protein Association; Gene-Disease Associations***

## I. INTRODUCTION

Biological data plays an imperative role in Bioinformatics domain that comprises DNA, RNA, Proteins, and Genes (Microarray). With the passage of time, these data have been growing very quickly in the form of interactions/associations such as [1-3] protein-protein and protein-gene. These interactions provide valuable information about the structure of the cell and their controlling mechanism. For the detection of Protein and Disease interactions, a lot of approaches are [4, 5] designed that improve the accuracy of Biological data interactions.

Over the time, the volume of biological data has increased. It is very important to find out specific genomic disease [6, 7] with the help of Proteomics interactions. Many researchers are trying to find out Protein and Disease interactions that give important information about their functions and behaviours. Prediction of Biological Processes is very informative [8] for molecular interactions. Protein pathways and complexes are determined by molecular interactions.

By the upcoming era, large interactions data have increased in the perspective of variety and volume. This data is referred to as Big Data which needs to be stored in the database very effectively. Existing PPI (Protein-Protein Interaction) Relational databases are DIP (Database of Interacting Proteins), MIPS (The Munich Information Centre for Protein Sequences), HPRD (Human Protein Reference Database), MINT [9] (The Molecular Interaction Database), BOND (Bimolecular Object Network Databank), IntAct and Reactome. However, these databases do not store large Interactions data in a structured and efficient way.

DIP [10] is specially designed to determine Proteins interactions by combining multiple sources into a unique and consistent set of PPI (Protein-Protein Interaction). MIPS' [11] research centre is used to manage the methods in Microarray gene expressions and Proteins data in a systematic way. HPRD [12] is OO (Object Oriented) database that is developed for specific Protein-Disease association. It provides the functionality of query optimisation by displaying data dynamically. MINT is based on verified Protein interactions that are presented graphically. BOND [13] is powerful databank that is designed for a combination of interactions and multiple sequences. It includes GenBank and stack of tools. IntAct [14] is a valuable open-source database that provides tools for interactions. Reactome [15] is a project that provides the cross-referenced functionality for many sequence databases. The above-mentioned databases lack to find some specific associations hence an Integration of these databases is required.

To remove these bottlenecks, open source Apache Hadoop [16] Platform have been developed for parallel execution of tasks in distributed manner across thousands of nodes. Its main tools are HBase [17] and Hive [18]. HBase framework is used to access real-time data randomly. It is NoSQL (Not only Structured Query Language) technology because scalability of large data in RDBMS (Relational Database Management System) shows poor performance. NoSQL databases consist of CAP (Consistency, Availability, and Partition Tolerance) mechanism with ACID (Atomicity, Consistency, Isolation and Durability) characteristics for tables. Sharding occurs automatically for sparse data by using HBase. Its logical view contains specific row key, column family, column key, timestamp and cell value. Its main parts are Region, Master,

Region Server, HDFS (Hadoop Distributed File System) and API (Application Programming Interface). Its basic operations are created, read, update and delete in the put, get and delete commands. Hive is DWH (Data Warehouse) framework that is designed for ad-hoc queries and writing reports by providing HQL (Hive Query Interface) for large data analysis. Its components are a web browser, driver, thrift server and client that interact with Hadoop. Its meta store exists in the form of Embedded, Remote, and Local states. Its data units contain tables, buckets, and partitions. It supports primitive and complex data types such as integers, strings, binary, arrays, maps, union, and structs. It provides shell interface, built-in functions, relational and arithmetic operators.

In this paper, a model is designed for large Protein-Genes interactions by integrating existing Relational databases. It provides the meaningful information for specific interactions.

The objectives of this paper are:

- A unified model for integration of different data sources
- NoSQL storage model
- Empirical study using HBase

The rest of this paper is structured as follows: Section II highlights the related work. Section III explains proposed a model. Section IV represents evaluation and Case Studies of that model. Section V concludes the whole work and mentions the future research domains in this field.

## II. Related Work

Zanzoni et al. [9] have worked on the Protein interaction databases which signify distinctive tools to store this information disseminated in the scientific literature in a computer-understandable form. A systematic and easily accessible database permits the examination of wide interaction data sets and enables easy retrieval. MINT presents a database which helps to reserve data for functional interactions among proteins. It was also considered to keep further types of functional interactions, containing enzymatic alternations of one of the partner. On the other hand, it provides cataloging binary complexes.

Chaurasia et al. [19] worked on the Systematic mapping of protein. Mapping of protein has highly been observed as a dominant task while practically working on functions of genomics. Numerous policies have just been followed to map human PPI. However, the author has produced a different kind of data set that is of high value for medicine experts and biomolecular data researchers. An open data management system named UniHI has been introduced to store and query information for more than 17000 human proteins interactions.

Apweiler et al. worked on the Universal Protein Resource (UniProt) [20] which is considered as a vital source of protein sequencing in bioinformatics as it gives a practical demonstration using three data storage mechanisms. First one is UniProt knowledge base that manually explains protein annotations, second is UniProtKB/TrEMBL, that stores these annotations and the third one is UniProtKB/Swiss-Prot that annotates proteins itself. Not only this database stores protein annotations but also help researchers to query for annotations and cross-references by linking them to the previous work done. It is an open source project that can be freely downloaded and used to get complete proteomes.

Chen et al. worked to visualize human protein-protein interaction (PPIs) and functional role of the data. Though numerous human PPI databases were found at that time yet defining all features of data was poor. The author named this data management system as Human Annotated and Predicted Protein Interaction (HAPPI) [21] database that is positioned at extraction and integration of new proteins interaction databases, which consists of BIND, STRING HPRD, MINT, and OPHID by means of database assimilation procedures. HAPPI is an open project that provides annotated information to help discover new horizons in biomolecular networks.

Aryamontri et al. worked for the explanation and study of proteins genetically and chemical interactions for all the species and introduced the Biological General Repository for Interaction Datasets (BioGRID) [22]. BioGRID is an open hub that provides all biological process related to humans diseases and suggests treatment for them. This data store includes 27501 interactions of chemical proteins that help to discover drugs to cure diseases. BioGRID is a dynamic interactions network that relates genetics and proteins interactions including bioactive compounds. This system gives results in visualisation form that can be adjusted according to the user's requirements.

Saeed et al. have worked on the proteomics and genomics. Proteogenomics is a [23] evolving ground of structures. The author has used mass spectrometry for proteomics and next generation sequencing for genomics. To mine Proteogenomics data set the author assimilated next-generation sequencing and mass spectrometry. Also for sequencing and high-performance computing solutions for such a big and complex data are discussed. The author has described possible storage format and analysis problems for such a multidimensional, large, and unstructured Proteogenomics data set. The study helps research community to recognize challenges and work on future guidelines as discussed.

Lehne et al. given the info about the protein interaction [24] databases. As protein-protein interactions are growing up with the passage of time so to store all the possible information related to these interactions some easily accessible databases are available. The author collected useful information from six major databases, described as, the Biological General Repository for Datasets [BioGRID], the Molecular INTeraction database [MINT], the Biomolecular Interaction Network Database [BIND], the Database of Interacting Proteins [DIP], the IntAct molecular interaction database [IntAct] and the Human Protein Reference Database [HPRD]). All these databases show different information on PPI and annotations.

Zhang et al. used the model driven architecture [25] software, that can store DNA and protein sequences efficiently. The author stored overlapping and non-overlapping DNA sequences in Apache Hadoop platform for space efficiency.

Xu worked on the vast availability of protein data including protein functions, sequences, annotations, and structures. The

author has started a new area of research by studying relationships between proteins of one family, between different protein families of one genome, and between the protein of different species. This study helps researchers to mine relating data and do predictive analysis based upon PPIs. The author has done working in Hadoop and its MapReduce functionality is used to explore insights for a protein of protein data storage.

Taylor has extensively worked on the Hadoop platform using MapReduce framework. Because bio Scientists have started dealing with ultra-large-scale data set analytics [27], the author used Hadoop as an open software for implementations on data of petabyte scale for distributed environments. Hadoop provides an efficient and cheap solution for NGS analysis for ultra-large and distributed data set across the cloud. The implementation includes HBase data storage along with Hadoop's map reduce function for data analytics.

Sarwar et al. proposed the work on Bioinformatics tools for sequencing [28], which are helpful to store a large amount of genomics data within a short time. The analysis study has shown that conventional bioinformatics tools cannot cope with the rate of production of such large amount of genomics data. So, there is a need to update previous tools or develop new ones to find new research aspects by defining proper storage structures of data on genetics.

Ali et.al [29] have discussed Microarray data analysis which gives the details of many gene Selection/Extraction and Classification tests/Algorithms. They also discuss the performance of different algorithms and Machine Learning techniques. Ahmed et al. [30] have discussed the modern data formats (models) for the implementation of spark, techniques in Hadoop MapReduce and Machine Learning Algorithms. It also describes the performance comparison of different data formats. R. Rehman et al. [31] have explained the importance of Scala language for Bioinformatics Tools/ Algorithms. They demonstrate the supported languages for Motif Finding Tools, Multiple Sequence Alignment Tools, and Pairwise Alignment tools.

## III. DATA STATISTICS

This dataset consists of protein, gene and disease columns which have a different type of interaction among them. The data set contains different column families which can have one or number of columns. These columns have values according to the families. The proposed data set contains 7 column families and defines different numbers of columns in each family. This protein, gene and disease interaction values are taken from different protein-interaction databases such as BioGRID, HPRD, EntrezGene, Ensembl etc. This dataset is the Homo-Sapiens organism. The available data sets on these platforms are in the form of CSV file. HBase column-oriented database is used for the storage of data.

## IV. PROPOSED MODEL

A model is an object or a procedure that explains some particular phenomena. There are many models that exist for PPI data. These models are used to store, analyse and search information related to protein interactions and also specify the characteristics of PPI data. Different models are used for different sets of purposes and also cover their usage in various fields. These models are DIP, OMIM, BIOGRID, STRING, UNIPROT, HPRD, INTACT, and so on. The Database of Interacting Proteins (DIP) does experimental interactions to determine various organisms. DIP contains 20728 proteins. 57683 interactions, and eight species that are (coli, Escherichia, norevegics. Rattus, Homo sapiens, muscles, helicobacter pylori, drosophila melanogaster). Its query format works as of relational databases and the user can fire text query via a web browser that displays results in visual form. It is organized in five key tables consists of proteins, trials and related data.

MINT is designed to stock information on practical interactions among proteins. It contains both physical interactions and other types of molecules. It delivers an integrated data model that experimentally confirms proteins interactions given in scientific literature by proficient curators.

INTACT is a data repository completely based on open source software. It works on two important factors from bimolecular data. One is proteins and the other is DNA. The data model of this database works on three main characteristics termed as EXPERIMENT, INTERACTIONS, and INTERACTOR. It provides a web-based interface for query searching.

The main function of BioGRID is to store proteins and genetics data in various organisms. BioGRID is mainly focused on investigating the interactions of networks regarding human health.

The HPRD shows a unified platform that integrates human proteome information and relates interaction networks between proteomes and diseases. It represents the relationship between them visually. All information is in this database is manually mined and explored from available literature by the analysts using the object-oriented database in zope.

The string is a projected interface in the database of more than 8000 organisms, it is used to organize a massive class of biochemical relationships between proteins to proteins and DNA to DNA. Strings work for two interactions. One is physical and second is direct e.g. two proteins contributed in an identical path.

MIPS is a research center presented at Neuherberg, Germany with an emphasis on genomes that are concerned with bioinformatics. Its purpose is to support and preserve fungal and plant genomes feature in a regular generic database.

All of these models stores, analyse and search the information about proteins interactions and some other features of PPI data. These databases use Relational schema to store data and in a structured format. These PPI database models offer a simple mechanism for the storage of data. These models of PPI can’t store unstructured and/or semi-structured PPI data sets.

In contrast to these researchers, we have designed a new data model for protein-protein, gene-protein, and gene-disease interactions. This model has two distinct features as compared to other existing interaction models. First of all, we integrated all existing protein-protein interaction data models and protein-gene interactions. We provide the facility to query all information for gene/ protein such as what is protein

interaction, gene interaction, and disease related information, in one storage system. The second prominent feature of this model is to follow the schema-less structure to store PPI data. Our data model is NoSQL storage and that can keep structured, semi-structured and unstructured data of protein-protein and protein-gene interactions in specified formats.

There are many technologies available in NoSQL databases, but this model is developed using HBase, that is built on the upper layer of Apache Hadoop. HBase is that is a column-oriented, distributed database, designed after the development of Google's Big tables. This database manages structured, semi-structured and unstructured data. HBase includes non-relational, open source, versioning, compression scalability and garbage collection features. The data stored in HBase can be manipulated using the programming structure of Hadoop like MapReduce. The storage format of HBase tables is given below in Figure 1.

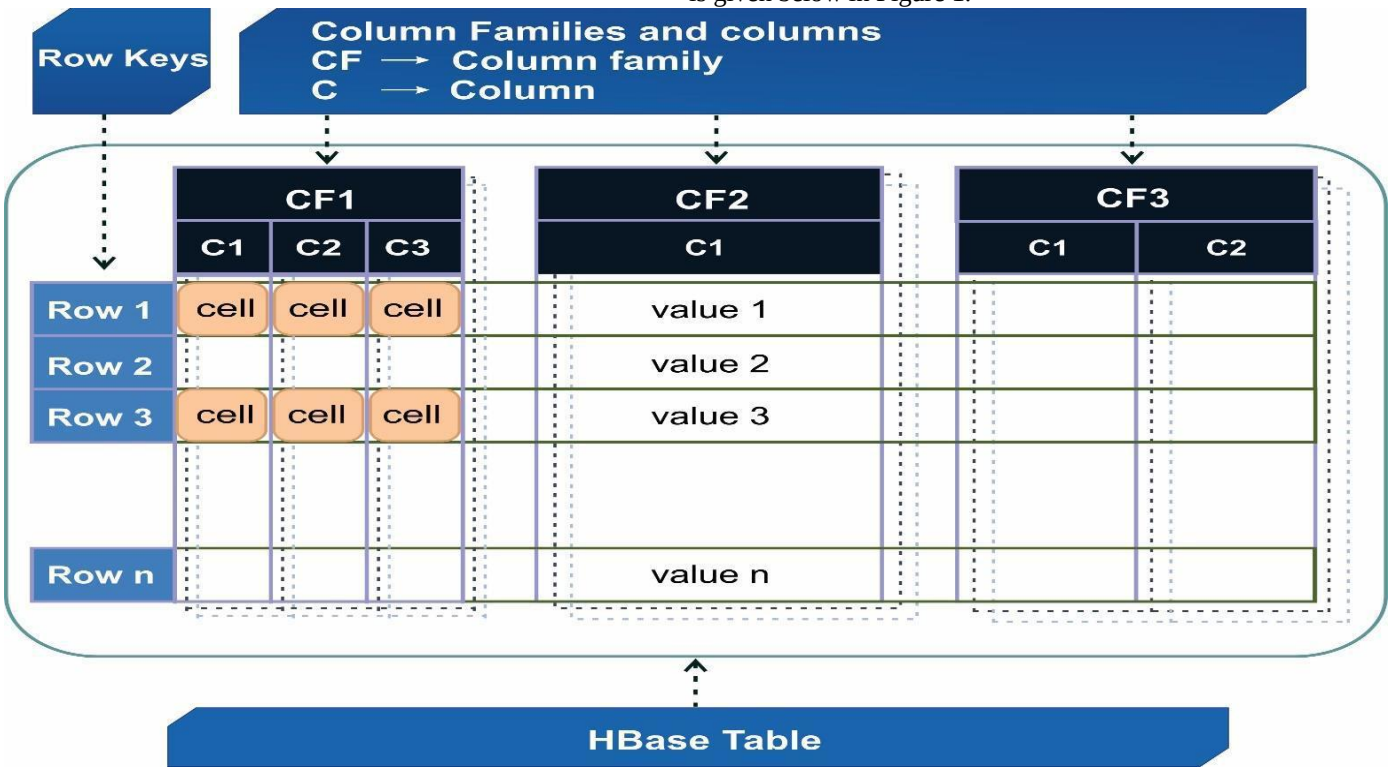


Fig. 1. HBase storage format

We applied our data model of protein-protein, protein-gene interaction in Apache HBase using column families for different purposes such as data source integration, Protein details, Gene details, RefSeq, Sequence in a different format, protein molecular information and biological information of protein/gene. These column families have different numbers of columns. The detail of data model is shown in Figure 2.

First column family is named as "Data-Integration-Source" has a defined number of columns in it. The first column contains Ids from the different data sources such as BioGRID, HPRD.

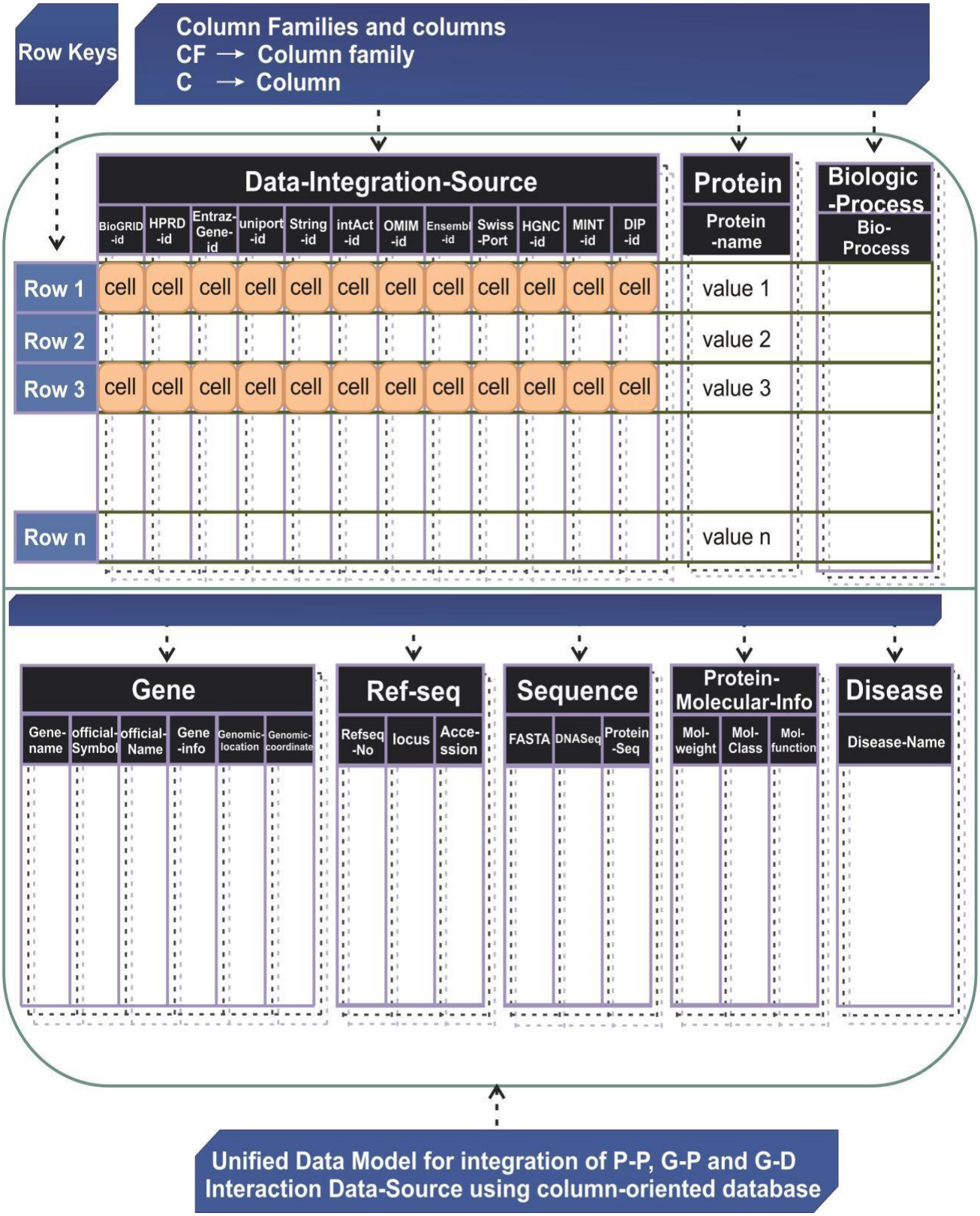


Fig. 2. Unified Data Model for P-P, G-P, and G-D Interaction

The IDs: Entrez-gene, Uniport, String, IntAct, OMIM, Ensembl, Swissport, HGNC, MINT, and DIP are different for the same protein in. Since this model integrates all existing models in a single column family so interaction types, interaction method, confidence scores and all the features of protein/genes can be viewed.

The column family "Protein" has a column named "protein-name" that gives information about protein name. "Gene" column family has four column named as Gene-name, official symbol, official name according to NCBI taxonomy and information about the gene.

The "Ref-seq" column family has three columns RefSeq-No, locus and Accession of protein. This family gives information about RefSeq of the protein, locus, and accession of the protein from the NCBI database. "Sequence" column family gives details about FASTA, DNA and protein /gene in three columns.

Two more column families are "Protein-Molecular-info" and "Biological-Process". In "Protein-Molecular-info" column family we have three columns that provide info of protein/gene such as the molecular-weight, molecular-class and molecular-function. The "Biological-Process" column family helps to get information about biological processes of protein/gene.

This NoSQL data model provides many advanced features that exhibit better performance, efficient storage, fast searching, deep analysis and integration of all models. This NoSQL model is a protein/gene interaction model that stores a huge number of data in a de-normalized form. It provides low latency operations for protein interaction data. They provide access to a single protein or gene interaction data from billions of interaction data records.

## V. Evaluation of Model

As our NoSQL data model is an integration of different protein-protein interaction databases like OMIM, BioGRID, Uniport, HPRD, Ensembl, UniHI, HAPPI, APID, and MiMI. The installation process for our data model starts from Apache Hadoop. Hadoop is an open-source, fast, reliable, low cost, distributed, and scale up from the individual server to thousands of machines. It provides storage and local computations that detect and handles the failures at applications layer. Hadoop by default uses HDFS (Hadoop Distributed File System) but our proposed data model stores data in HBase on top of Hadoop.

We wrote simple queries to identify different relationships and object of protein, gene, and diseases from the model that fetch the related records. These queries can easily fetch data according to user requirements from relevant columns of column families. After entering into HBase shell all operations on created table named 'protein data' can be applied. We write scan (keyword) followed by table name in single quotation marks to get all data entries in that table along with column names for every single column family. The output of applying scan query on HBase table is shown in Figure 3.

```
hbase (main) :001:0> scan 'proteindata'

entrez gene/locuslink:942          column=B:bio_process, timestamp=1491722208943, value=NP_001193854
entrez gene/locuslink:942          column=DSI:BioGrid_id, timestamp=1491722208943, value=107380
entrez gene/locuslink:942          column=DSI:EDS_id, timestamp=1491722208943, value=swiss_port-P42081|HGNC-1705|MINT-6631610|DIP-35606N
entrez gene/locuslink:942          column=DSI:EnGene_id, timestamp=1491722208943, value=942
entrez gene/locuslink:942          column=DSI:Ensembl_id, timestamp=1491722208943, value=CD86 ENSG00000114013
entrez gene/locuslink:942          column=DSI:HPRD_id, timestamp=1491722208943, value=3011
entrez gene/locuslink:942          column=DSI:IntAct_id, timestamp=1491722208943, value=P42081
entrez gene/locuslink:942          column=DSI:OMIM_id, timestamp=1491722208943, value=601020
entrez gene/locuslink:942          column=DSI:String_id, timestamp=1491722208943, value=9606.ENSP00000332049
entrez gene/locuslink:942          column=DSI:Uniprot_id, timestamp=1491722208943, value=UniProtKB - P42081 (CD86_HUMAN)
entrez gene/locuslink:942          column=G:coor, timestamp=1491722208943, value= 1 paralogue
entrez gene/locuslink:942          column=G:g_info, timestamp=1491722208943, value="This gene has 9 transcripts (splice variants)
entrez gene/locuslink:942          column=G:gen_loc, timestamp=1491722208943, value= 50 orthologues
entrez gene/locuslink:942          column=G:gene, timestamp=1491722208943, value=CD86
entrez gene/locuslink:942          column=G:ofF_name, timestamp=1491722208943, value=CD86 molecule
entrez gene/locuslink:942          column=G:off_Symbol, timestamp=1491722208943, value=CD86
entrez gene/locuslink:942          column=MI:mol_class, timestamp=1491722208943, value=NP_001193854.1
entrez gene/locuslink:942          column=MI:mol_fun, timestamp=1491722208943, value= NP_001193854            247 aa            linear   P
                                    06-OCT-2016
entrez gene/locuslink:942          column=MI:mol_weight, timestamp=1491722208943, value=142"
entrez gene/locuslink:942          column=RF:Accession, timestamp=1491722208943, value=" 3:122
entrez gene/locuslink:942          column=RF:Locus, timestamp=1491722208943, value=3q13.33
entrez gene/locuslink:942          column=RF:RefSeq_No, timestamp=1491722208943, value= is a member of 1 Ensembl protein family and is associ
                                   ated with 9 phenotypes."
entrez gene/locuslink:942          column=disease:disease, timestamp=1491722208943, value=">sp|P42081|CD86_HUMAN T-lymphocyte activation anti
                                   gen CD86 OS=Homo sapiens GN=CD86 PE=1 SV=2
entrez gene/locuslink:942          column=protein:protein, timestamp=1491722208943, value=T-lymphocyte activation antigen CD86
entrez gene/locuslink:942          column=seq:DNA, timestamp=1491722208943, value=361-122
entrez gene/locuslink:942          column=seq:FASTA, timestamp=1491722208943, value=055
entrez gene/locuslink:942          column=seq:ProSeq, timestamp=1491722208943, value=121
entrez gene/locuslink:9463         column=DSI:BioGrid_id, timestamp=1491722208943, value=114849
entrez gene/locuslink:9463         column=DSI:EDS_id, timestamp=1491722208943, value="swiss_port-\x09
entrez gene/locuslink:9463         column=DSI:EnGene_id, timestamp=1491722208943, value=9463
entrez gene/locuslink:9463         column=DSI:Ensembl_id, timestamp=1491722208943, value=ENSG00000100151
entrez gene/locuslink:9463         column=DSI:HPRD_id, timestamp=1491722208943, value=16176
entrez gene/locuslink:9463         column=DSI:IntAct_id, timestamp=1491722208943, value=Q9NRD5
entrez gene/locuslink:9463         column=DSI:OMIM_id, timestamp=1491722208943, value=605926
entrez gene/locuslink:9463         column=DSI:String_id, timestamp=1491722208943, value=\x099606.ENSP00000349465
entrez gene/locuslink:9463         column=DSI:Uniprot_id, timestamp=1491722208943, value=UniProtKB - Q9NRD5 (PICK1_HUMAN)
entrez gene/locuslink:9584         column=B:bio_process, timestamp=1491722208943, value=">sp|Q14498|RBM39_HUMAN RNA-binding protein 39 OS=Hom
                                   o sapiens GN=RBM39 PE=1 SV=2
```

Fig. 3. Scanning Data from ProteinData Table

The data can also be extracted from an entire column-family. 'Scan' command is used to extract all cells entries along with column names and time stamp. For example scanning a particular column family named as 'DSI' (Data-Source-integration) will result in all column names and data values in it. The names of columns in this column family are IDs from all specified databases, written as, BioGrid_id, EDS_id, EnGene_id, Ensembl_id, HPRD_id, IntAct_id, OMIM_id, String_id, and Uniprot_id as given in Figure 4.

```
hbase (main) :001:0> scan 'proteindata', {COLUMNS => 'DSI'}

entrez gene/locuslink:9113          column=DSI:BioGrid_id, timestamp=1491722208943, value=114563
entrez gene/locuslink:9113          column=DSI:EDS_id, timestamp=1491722208943, value=swiss_port-O95835|HGNC-6514
entrez gene/locuslink:9113          column=DSI:EnGene_id, timestamp=1491722208943, value=9113
entrez gene/locuslink:9113          column=DSI:Ensembl_id, timestamp=1491722208943, value=LATS1 ENSG00000131023
entrez gene/locuslink:9113          column=DSI:HPRD_id, timestamp=1491722208943, value=9147
entrez gene/locuslink:9113          column=DSI:IntAct_id, timestamp=1491722208943, value=O95835. 45 interactors.
entrez gene/locuslink:9113          column=DSI:OMIM_id, timestamp=1491722208943, value=603473
entrez gene/locuslink:9113          column=DSI:String_id, timestamp=1491722208943, value=9606.ENSP00000253339.
entrez gene/locuslink:9113          column=DSI:Uniprot_id, timestamp=1491722208943, value=UniProtKB - O95835 (LATS1_HUMAN)
```

Fig. 4. Extracting Data from DSI column-family

Similarly, for 'disease' column-family, the query will be written as 'scan (keyword)' followed by table name and then 'COLUMNS (keyword)' along with column family name, according to the syntax, to get all columns entries. The query results in all columns covering details of disease for particular genes and proteins as shown below in Figure 5.

```
hbase (main) :002:0> scan 'proteindata', {COLUMNS => 'disease'}

entrez gene/locuslink:84062         column=disease:disease, timestamp=1491722208943, value=">tr|A0A087WYP9|A0A087WYP9_HUMAN Dysbindin OS=Homo
                                    sapiens GN=DTNBP1 PE=1 SV=1
entrez gene/locuslink:84665         column=disease:disease, timestamp=1491722208943, value=">sp|Q86TC9|MYPN_HUMAN Myopalladin OS=Homo sapiens
                                    GN=MYPN PE=1 SV=2
entrez gene/locuslink:85453         column=disease:disease, timestamp=1491722208943, value=">sp|Q86VY4|TSYL5_HUMAN Testis-specific Y-encoded-l
                                    ike protein 5 OS=Homo sapiens GN=TSPYL5 PE=1 SV=2
entrez gene/locuslink:8767          column=disease:disease, timestamp=1491722208943, value=">sp|O43353|RIPK2_HUMAN Receptor-interacting serine
                                    /threonine-protein kinase 2 OS=Homo sapiens GN=RIPK2 PE=1 SV=2
entrez gene/locuslink:8797          column=disease:disease, timestamp=1491722208943, value=">sp|O00220|TR10A_HUMAN Tumor necrosis factor recep
                                    tor superfamily member 10A OS=Homo sapiens GN=TNFRSF10A PE=1 SV=3
entrez gene/locuslink:8848          column=disease:disease, timestamp=1491722208943, value=">tr|A0A087X0H8|A0A087X0H8_HUMAN TSC22 domain famil
                                    y protein 1 OS=Homo sapiens GN=TSC22D1 PE=1 SV=1
```

Fig. 5. Extracting Data from 'disease' column-family

Similarly to scan 'gene (G)' column family the query would be written as 'scan (keyword)' followed by table name and then 'COLUMNS (keyword)' along with column family name, according to the syntax, to get all columns entries. The query results in all columns covering details of genes such as gene name, gene symbol, gene location, coordinates of a gene and gene information. The "G" stands for the gene in the query. In Figure 6, different genes attributes are given.

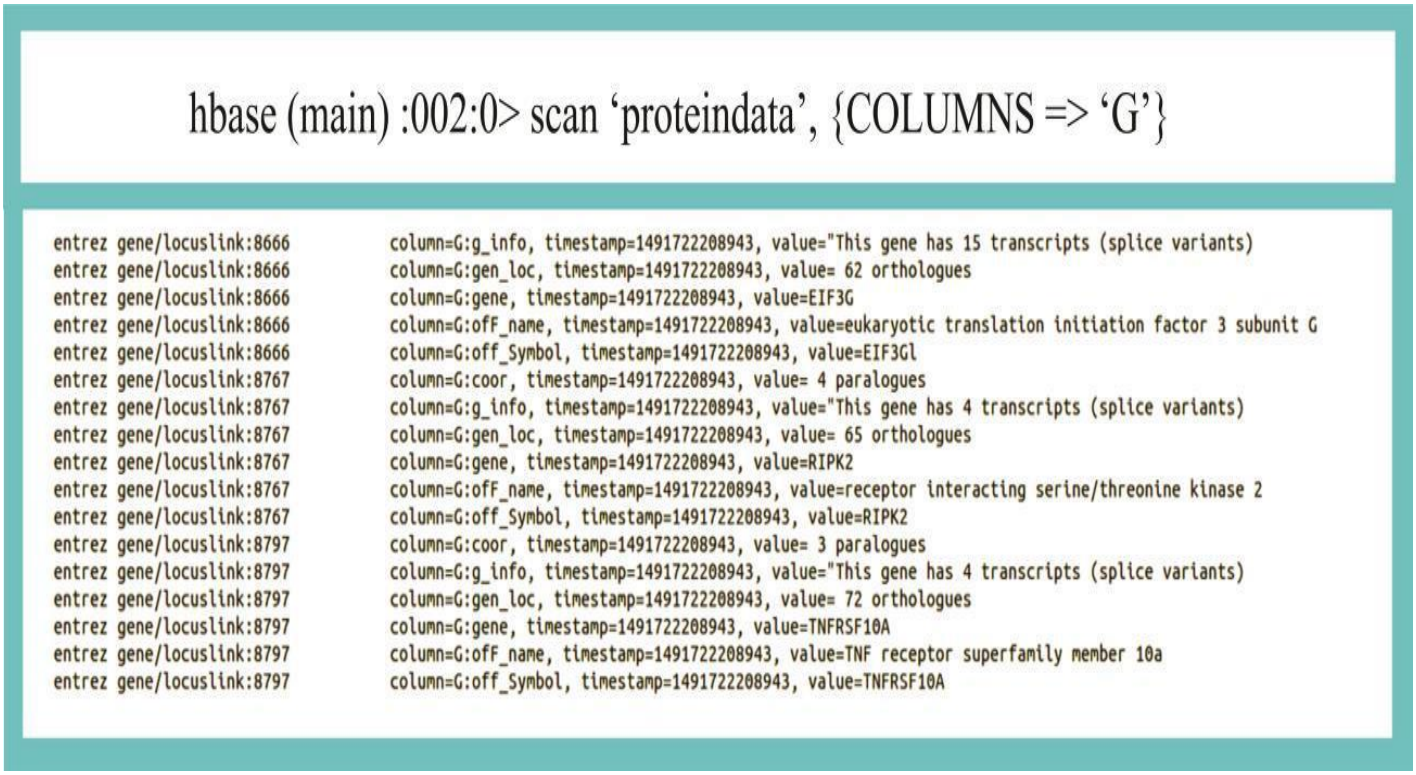


Fig. 6. Extracting Data from G column-family

To get details of all columns in all column families against a particular entity we have to specify the index for that row. For example 'Entrez gene/locuslink: 8797' is used as an index to get all entries for this record. And it shows a separate list of all column families followed by a colon (:) and their column names that have data entries in it. The query format and its results are shown below in Figure 7.

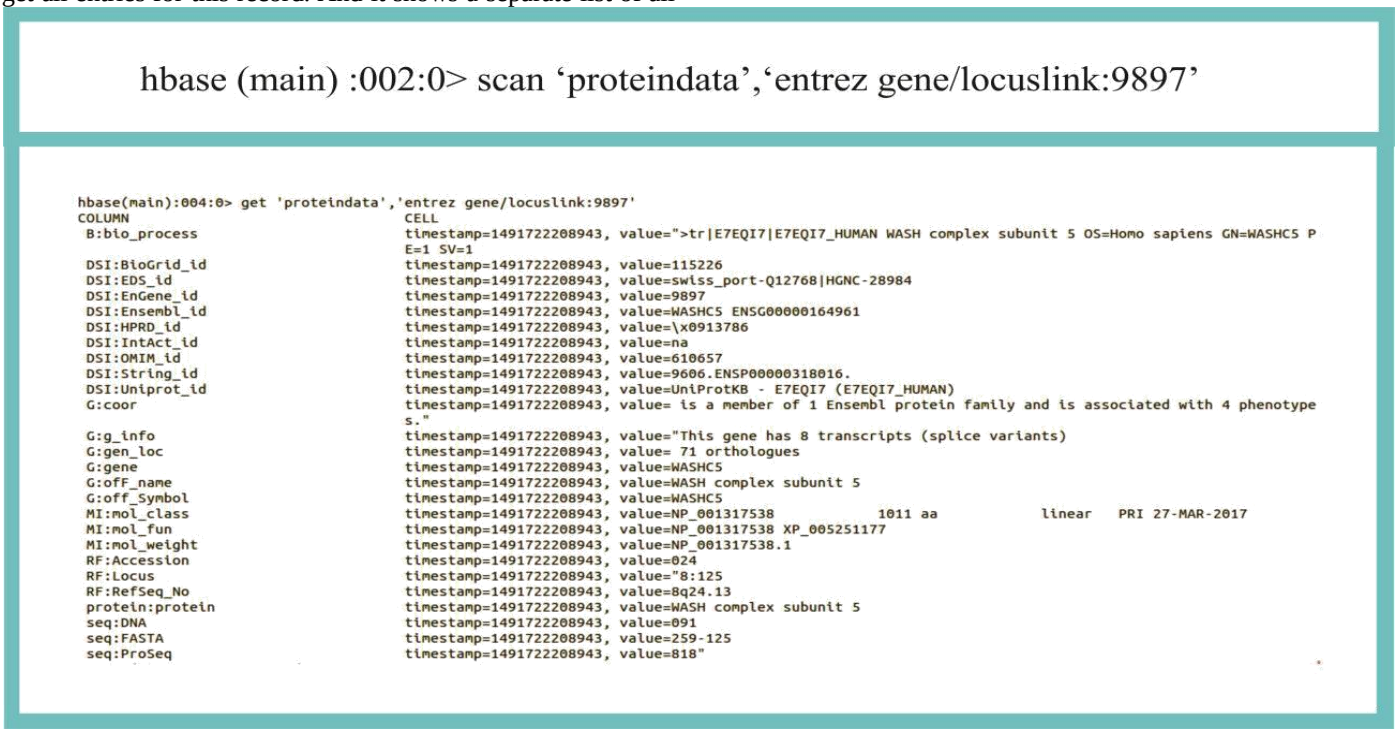


Fig. 7. Extracting Data of Specific Gene/Protein

Apache drill is an open-source platform implementing SQL queries on NoSQL databases that store big data. The main purpose of introducing this framework is to provide a standard language like SQL that can query big data applications' data sets (that can be semi-structured and/or unstructured) stored in NoSQL data storage formats. Drill by default does not support Apache Hive and Apache HBase but we have to enable these storage formats in it and enable data ports on which our local host is working. It provides the functionality to query multiple data storage systems in one single query. For example, a user can query accountant information from HBase and event logs from local HDFS in Hadoop. Drill facilitates researchers with its datastore-aware optimizer that can automatically rebuild queries to leverage its datastore's internal processing capabilities. Apache drill also provides data locality, so keeping drill and datastore on same nodes can save time and provide faster results.

In this model, we use Apache Drill in integration with Apache HBase for getting results of protein and gene interactions datasets. Query format for Apache Drill is different from HBase. For our proposed data model, drill query to get all entries of columns from the same column family can be defined so easily. For example, if we want to get gene IDs of all databases stored under 'DSI' column-family, we have to mention table name, column-family Name, column-Name from HBase table. The query format and its results are shown below in Figure 8.

Drill query to retrieve data from different column families at a time to predict different relations in our proposed model is shown as below. First of all, we mention 'Gene_id' as row_key for indexing and after that required column names are called using dot operator for related column families and table name. Query to get information of disease ID named as 'OMIM_id' from 'disease' column family and associated gene name from 'G' column family is shown below in Figure 9.

```
SELECT CONVERT_FROM (row_key, 'UTF8') As Gene_id, CONVERT_FROM
(proteindata.G.gene, 'UTF8') AS Gene_Name, and CONVERT_FROM
(proteindata.DSI.OMIM_id, 'UTF8') As OMIM_id, FROM proteindata;
```

| Gene_id | Gene_Name | OMIM_id |
|---|---|---|
| entrez gene/locuslink:22846 | VASH1 | 609011 |
| entrez gene/locuslink:22937 | SCAP | 601510 |
| entrez gene/locuslink:22938 | SNW1 | 603055 |
| entrez gene/locuslink:23013 | SPEN | 613484 |
| entrez gene/locuslink:23043 | TNIK | 610005 |
| entrez gene/locuslink:23047 | PDS5B | 605333 |
| entrez gene/locuslink:23085 | ERC1 | 607127 |
| entrez gene/locuslink:23111 | SPG20 | 607111 |
| entrez gene/locuslink:23161 | null | 606589 |
| entrez gene/locuslink:23184 | MESDC2 | 607783 |
| entrez gene/locuslink:23190 | null | 611216 |
| entrez gene/locuslink:23263 | null | 609499 |
| entrez gene/locuslink:23295 | MGRN1 | 607559 |
| entrez gene/locuslink:23304 | Nbla10317 | 116896 |

Fig. 8. Extracting Column-ID of DSI Column Family using Apache Drill

```
SELECT CONVERT_FROM (row_key, 'UTF8') As Gene_id, CONVERT_FROM
(proteindata.DSI.BioGrid_id, 'UTF8') AS BioGRID_ID, CONVERT_FROM
(proteindata.DSI.HPRD_id, 'UTF8') As HPRID_ID, CONVERT_FROM
(proteindata.DSI.Uniprot_id, 'UTF8' AS Uniprot_id FROM proteindata;
```

| Gene_id | BioGRID_ID | HPRD_ID | EntranzGene_id | Uniprot_id |
|---|---|---|---|---|
| entrez gene/locuslink:50807 | 119126 | 5809 | 50807 | UniProtKB - Q9ULH1 (ASAP1_HUMAN) |
| entrez gene/locuslink:50838 | 119148 | 11991 | 50838 | UniProtKB - Q9NYV9 (T2R13_HUMAN) |
| entrez gene/locuslink:51035 | 119239 | 11266 | entrez gene/locuslink:51035 | UniProtKB - A0A024R598 (A0A024R598_HUMAN) |
| entrez gene/locuslink:51109 | 119298 | 9707 | 51109 | UniProtKB - Q8TC12 (RDH11_HUMAN) |
| entrez gene/locuslink:51127 | 119314 | 5839 | 51127 | UniProtKB - Q9Y577 (TRI17_HUMAN) |
| entrez gene/locuslink:51196 | 119370 | 07087 | 51196 | UniProtKB - B7ZM61 (B7ZM61_HUMAN) |
| entrez gene/locuslink:51330 | 119478 | 5801 | 51330 | UniProtKB - Q9NP84 (TNR12_HUMAN) |
| entrez gene/locuslink:51366 | 119501 | 06436 | 51366 | UniProtKB - E7EMW7 (E7EMW7_HUMAN) |
| entrez gene/locuslink:51386 | 119516 | 10932 | 51386 | UniProtKB - B0QY90 (B0QY90_HUMAN) |
| entrez gene/locuslink:51400 | 119524 | 17864 | 51400 | UniProtKB - Q9Y570 (PPME1_HUMAN) |
| entrez gene/locuslink:51429 | 119535 | 12072 | 51429 | UniProtKB - Q9Y5X1 (SNX9_HUMAN) |
| entrez gene/locuslink:51460 | 119553 | 9539 | 51460 | UniProtKB - Q9UHJ3 (SMBT1_HUMAN) |
| entrez gene/locuslink:51497 | 119572 | 16096 | 51497 | UniProtKB - Q8IXH7 (NELFD_HUMAN) |
| entrez gene/locuslink:51510 | 119579 | 15408 | 51510 | UniProtKB - Q9NZZ3 (CHMP5_HUMAN) |
| entrez gene/locuslink:51528 | 119590 | 12623 | entrez gene/locuslink:51528 | UniProtKB - A0A087WVY6 (A0A087WVY6_HUMAN) |
| entrez gene/locuslink:51534 | 119595 | 9855 | 51534 | UniProtKB - A0A087WY55 (A0A087WY55_HUMAN) |
| entrez gene/locuslink:5156 | 111182 | 1429 | 5156 | UniProtKB - P16234 (PGFRA_HUMAN) |
| entrez gene/locuslink:51699 | 119683 | 9499 | 51699 | UniProtKB - F8VXU5 (F8VXU5_HUMAN) |
| entrez gene/locuslink:5320 | 111337 | 1397 | 5320 | UniProtKB - A0A024RA96 (A0A024RA96_HUMAN) |
| entrez gene/locuslink:53358 | 119752 | 12005 | 53358 | UniProtKB - Q92529 (SHC3_HUMAN) |
| entrez gene/locuslink:5340 | 111356 | 1417 | 5340 | UniProtKB - P00747 (PLMN_HUMAN) |
| entrez gene/locuslink:53981 | 119826 | 5824 | 53981 | UniProtKB - Q9P2I0 (CPSF2_HUMAN) |
| entrez gene/locuslink:54106 | 119902 | 5685 | 54106 | UniProtKB - Q9NR96 (TLR9_HUMAN) |
| entrez gene/locuslink:54331 | 119933 | 16234 | 54331 | UniProtKB - P59768 (GBG2_HUMAN) |
| entrez gene/locuslink:5437 | 111433 | 9082 | entrez gene/locuslink:5437 | UniProtKB - P52434 (RPAB3_HUMAN) |
| entrez gene/locuslink:54431 | 119947 | 9722 | 54431 | UniProtKB - Q8IXB1 (DJC10_HUMAN) |
| entrez gene/locuslink:54455 | 119962 | NA | 54455 | UniProtKB - A0A024QZB0 (A0A024QZB0_HUMAN) |
| entrez gene/locuslink:5450 | 111446 | 03126 | 5450 | UniProtKB - Q16633 (OBF1_HUMAN) |

Fig. 9. Disease ID against Gene/Protein in Model using Apache Drill

This NoSQL data model provides the opportunity to search data against a particular value, from any column of one or more column families. For example, if we want to get gene ID against a specific BioGrid_id='121229' from the full table, we'll use 'WHERE' clause followed by data entry to get matched to. In this case, the query and retrieved information are shown below in Figure 10.

As we have extracted specific ID of data-source column family and "G" gene column family in HBase shell, in the same way, can use drill query to get sequences of a diverse type like FASTA, DNA and protein sequences from 'seq' column-family. For our data model, drill query to get all this information along with proteins and their related genes is given in Figure 11.

```
SELECT CONVERT_FROM (row_key,'UTF8') FROM
proteindata WHERE proteindata.DSI.BioGrid_id='121229';
```

| Gene_id |
|---|
| entrez gene/locuslink:56899 |
| ntrez gene/locuslink:23294 |

Fig. 10. Extraction of Column-id from DSI column familyUsing Apache Drill

```
SELECT CONVERT_FROM (row_key, 'UTF8') As Gene_id, CONVERT_FROM
(proteindata.seq.FASTA, 'UTF8') AS FASTA, CONVERT_FROM
(proteindata.seq.DNA, 'UTF8') As DNA, and CONVERT_FROM (proteindata.seq.Proseq,
'UTF8' AS ProSeq FROM proteindata;
```

```
| CCGGCTGAGC CAGCGGCTCT TGGGAGGCTG CGTCCGCGCG CCGGCGAGGC GAGGCGGCCG GGCCCTGCGC GTCAGGCCTG AGACCTGGGA GGAAGCTGGA GAAAAGATGC CCTCTGAATC TTTCTGTTTG
 GCTGCCCAGG CTCGCCTCGA CTCCAAATGG TTGAAAACAG ATATACAGCT TGCATTCACA AGAGATGGGC TCTGTGGTCT GTGGAATGAA ATGGTTAAAG ATGGAGAAAT TGTATACACT GGAACAGAAT
CAACCCAGAA CGGAGAGCTC CCTCCTAGAA AAGATGATAG TGTCGAACCA AGTGGAACAA AGAAAGAAGA TCTGAATGAC AAAGAGAAAA AAGATGAAGA AGAAACTCCT GCACCTATAT ATAGGGCCAA G
TCAATTCTG GACAGCTGGG TATGGGGCAA GCAACCAGAT GTGAATGAAC TGAAGGAGTG TCTTTCTGTG CTGGTTAAAG AGCAGCAGGC CCTGGCCGTC CAGTCAGCCA CCACCACCCT CTCAGCCCTG CG
ACTCAAGC AGAGGCTGGT GATCTTGGAG CGCTATTTCA TTGCCTTGAA TAGAACCGTT TTTCAGGAGA ATGTCAAAGT TAAGTGGAAA AGCAGCGGTA TTTCTCTGCC TCCTGTGGAC AAAAAAAGTT CCC
GGCCTGC GGGCAAAGGT GTGGAGGGGC TCGCCAGAGT GGGATCCCGA GCGGCGCTGT CTTTTGCCTT TGCCTTCCTG CGCAGGGCCT GGCGATCAGG CGAGGATGCG GACCTCTGCA GTGAGCTGTT GCAG
GAGTCC CTGGACGCCC TGCGAGCACT TCCCGAGGCC TCGCTCTTTG ACGAGAGCAC CGTGTCCTCT GTGTGGCTGG AGGTGGTGGA GAGAGCGACC AGGTTCCTCA GGTCCGTCGT GACGGGGGAT GTTCA
CGGAA CGCCAGCCAC CAAAGGGCCA GGAAGCATCC CCCTGCAGGA CCAGCACTTG GCCCTGGCCA TCCTGCTGGA GCTGGCTGTG CAGAGAGGCA CGCTGAGCCA AATGTTGTCT GCCATCCTGT TGTTGC
TTCA GCTGTGGGAC AGCGGGGCAC AGGAGACTGA CAATGAGCGT TCCGCCCAGG GCACCAGCGC CCCACTTTTG CCCTTGCTGC AAAGGTTCCA GAGCATCATT TGCAGGAAGG ATGCACCCCA CTCCGAG
GGC GACATGCACC TTTTGTCTGG CCCTCTGAGC CCCAATGAGA GTTTCCTGAG GTACCTCACC CTTCCACAAG ACAACGAGCT TGCCATTGAT CTGCGACAAA CGGCGGTTGT TGTCATGGCC CATTTAGA
CC GTCTGGCTAC GCCCTGTATG CCTCCGCTGT GTAGCTCTCC GACATCTCAT AAGGGATCAT TGCAAGAGGT CATAGGTTGG GGGTTAATAG GATGGAAATA CTATGCCAAT GTGATTGGTC CAATCCAGT
G CGAAGGCCTG GCCAACCTGG GAGTCACACA GATTGCCTGT GCAGAGAAGC GTTTCCTGAT TCTGTCACGC AATGGCCGCG TGTACACACA GGCCTATAAT AGTGACACGC TGGCCCCACA GCTGGTCCAA
 GGCCTTGCCT CCAGAAACAT TGTAAAAATT GCTGCCCATT CTGATGGTCA CCACTACCTA GCCTTGGCTG CTACTGGAGA GGTGTACTCC TGGGGCTGTG GGGACGGCGG ACGGCTGGGC CATGGGGACA
CTGTGCCTTT GGAGGAGCCT AAGGTGATCT CCGCCTTCTC TGGAAAGCAG GCCGGGAAGC ACGTGGTGCA CATCGCTTGC GGGAGCACTT ACAGTGCGGC CATCACTGCC GAGGGGGAGC TGTACACCTG G
GGCCGCGGG AACTACGGCC GGCTGGGCCA TGGCTCCAGT GAGGACGAGG CCATTCCGAT GCTGGTAGCC GGGCTTAAAG GACTGAAGGT CATCGATGTG GCGTGTGGGA GTGGGGATGC TCAAACCCTG GC
TGTCACTG AGAACGGGCA AGTGTGGTCT TGGGGAGATG GTGACTATGG GAAATTGGGC AGAGGTGGTA GTGATGGCTG CAAAACCCCA AAGCTGATTG AAAAGCTTCA AGACTTGGAT GTGGTCAAAG TCC
GCTGTGG AAGTCAGTTT TCCATTGCTT TGACGAAAGA TGGCCAAGTT TATTCATGGG GAAAAGGTGA CAACCAGAGA CTTGGACATG GAACAGAGGA ACATGTTCGT TATCCAAAAC TCTTAGAAGG CTTG
CAAGGG AAGAAGGTGA TTGATGTGGC TGCAGGCTCC ACCCACTGCC TGGCTCTGAC TGAGGACAGC GAGGTCCACA GCTGGGGGAG CAACGACCAG TGCCAGCACT TTGACACCTT GCGCGTGACC AAGCC
AGAAC CTGCAGCATT GCCAGGACTG GACACCAAAC ACATAGTGGG AATTGCCTGT GGGCCTGCCC AGAGCTTTGC TTGGTCATCA TGTTCTGAGT GGTCCATTGG CCTCCGTGTC CCTTTTGTGG TGGACA
TCTG CTCAATGACT TTTGAGCAGC TGGATCTCCT GCTTCGGCAG GTGAGTGAGG GGATGGATGG TTCCGCGGAC TGGCCCCCGC CCCAGGAGAA AGAGTGTGTG GCCGTGGCAA CGCTGAATCT TCTACGA
CTT CAGTTGCATG CTGCCATTAG TCACCAGGTT GACCCGGAAT TCCTTGGTTT AGGTCTGGGC AGCATCCTCC TGAACAGCCT GAAGCAGACG GTGGTGACCC TGGCCAGCAG TGCGGGCGTG CTGAGCAC
CG TGCAGTCGGC CGCCCAGGCC GTGCTGCAGA GTGGCTGGTC CGTGCTGCTG CCCACCGCGG AGGAGCGGGC CCGGGCACTC TCTGCTCTCC TGCCCTGCGC AGTTTCAGGC AATGAAGTGA ACATAAGTC
C AGGTCGTCGA TTCATGATTG ATCTTCTGGT GGGCAGCTTG ATGGCTGATG GAGGGTTGGA GTCAGCCTTA CACGCAGCCA TTACTGCAGA GATCCAGGAT ATTGAAGCCA AAAAAGAAGC ACAGAAGGAA
 AAAGAAATTG ATGAACAGGA AGCGAATGCC TCAACATTTC ATAGAAGCAG GACTCCACTG GATAAAGACC TTATTAATAC GGGGATCTGT GAGTCTTCTG GCAAACAGTG TTTGCCTCTG GTTCAGCTCA
TACAACAGCT TCTTAGAAAC ATTGCTTCTC AGACTGTAGC CAGATTGAAA GATGTTGCCC GTCGGATTTC ATCATGTCTG GACTTTGAGC AACACAGTCG TGAAAGATCT GCTTCATTGG ATTTGTTACT G
CGTTTTCAA CGTTTGCTTA TTAGTAAACT TTATCCAGGA GAAAGTATTG GTCAGACCTC AGATATTTCT AGTCCAGAGC TAATGGGTGT TGGTTCCTTG CTGAAGAAGT ACACAGCCCT CCTGTGCACG CA
CATTGGAG ATATACTGCC TGTGGCCGCC AGCATTGCTT CTACCAGCTG GCGGCACTTC GCGGAGGTGG CTTACATTGT GGAAGGGGAC TTTACTGGTG TTCTCCTTCC AGAACTAGTA GTTTCTATAG TGC
TTCTGCT CAGTAAAAAT GCTGGTCTCA TGCAAGAGGC TGGAGCTGTA CCTCTGCTGG GTGGCCTGTT GGAACATCTG GATCGGTTCA ACCATCTGGC ACCAGGAAAG GAACGGGATG ATCATGAAGA GTTA
GCCTGG CCTGGCATAA TGGAGTCATT TTTTACAGGT CAGAACTGTA GAAATAATGA GGAAGTGACA CTTATACGCA AAGCTGATTT GGAGAACCAT AATAAAGATG GAGGCTTCTG GACTGTGATT GACGG
GAAGG TGTATGATAT AAAGGACTTC CAGACACAGT CGTTAACAGG AAATAGTATT CTTGCTCAGT TTGCAGGGGA AGACCCAGTG GTAGCTTTGG AAGCTGCTTT GCAGTTTGAA GACACCCGGG AATCCA
TGCA CGCGTTTTGT GTTGGCCAGT ATTTGGAGCC TGACCAAGAA ATCGTCACCA TACCAGATCT GGGGAGTCTC TCTTCACCTC TGATAGACAC AGAGAGGAAT CTGGGCCTGC TTCTCGGATT ACACGCT
TCG TATTTGGCAA TGAGCACACC GCTGTCTCCT GTCGAGATTG AATGTGCCAA ATGGCTTCAG TCATCCATCT TCTCTGGAGG CCTGCAGACC AGCCAGATCC ACTACAGCTA CAACGAGGAG AAAGACGA
GG ACCACTGCAG CTCCCCAGGG GGCACACCTG CCAGCAAATC TCGACTCTGC TCCCACAGAC GGGCCCTGGG GGACCATTCC CAGGCATTTC TGCAAGCCAT TGCAGACAAC AACATTCAGG ATCACAACG
T GAAGGACTTT TTGTGTCAAA TAGAAAGGTA CTGTAGGCAG TGCCATTTGA CCACACCGAT CATGTTTCCC CCCGAGCATC CCGTGGAAGA GGTCGGTCGC TTGTTGTTAT GTTGCCTCTT AAAACATGAA
 GATTTAGGTC ATGTGGCATT ATCTTTAGTT CATGCAGGTG CACTTGGTAT TGAGCAAGTA AAGCACAGAA CGTTGCCTAA GTCAGTGGT" |
```

Fig. 11. Extraction of Sequence column familyUsing Apache Drill

We have written another drill query to show some important relations between gene names and protein names against a particular Gene_id as defined in NCBI's Entrez database. This query searches for gene name for a particular gene_id and shows the name of the protein that it makes interactions with. Query to extract Gene and protein names from "DSI" (data source integration) and "G" gene column family respectively, is shown in Figure 12.

```
SELECT CONVERT_FROM (row_key, 'UTF8') As Gene_id, CONVERT_FROM
(proteindata.G.gene, 'UTF8') AS Gene_Name, CONVERT_FROM
(proteindata.protein.protein,'UTF8') As protein_name FROM proteindata;
```

| Gene_id | Gene_Name | protein_name |
|---|---|---|
| entrez gene/locuslink:22846 | VASH1 | Vasohibin-1 |
| entrez gene/locuslink:22937 | SCAP | Sterol regulatory element-binding protein cleavage-activating protein |
| entrez gene/locuslink:22938 | SNW1 | SNW domain-containing protein 1 |
| entrez gene/locuslink:23013 | SPEN | Msx2-interacting protein |
| entrez gene/locuslink:23043 | TNIK | TRAF2 and NCK-interacting protein kinase |
| entrez gene/locuslink:23047 | PDS5B | Sister chromatid cohesion protein PDS5 homolog B |
| entrez gene/locuslink:23085 | ERC1 | ELKS/Rab6-interacting/CAST family member 1 |
| entrez gene/locuslink:23111 | SPG20 | Submitted name: Spastic paraplegia 20 isoform 1 |
| entrez gene/locuslink:23161 | null | null |
| entrez gene/locuslink:23184 | MESDC2 | LDLR chaperone MESD |
| entrez gene/locuslink:23190 | null | null |
| entrez gene/locuslink:23263 | null | null |
| entrez gene/locuslink:23295 | MGRN1 | E3 ubiquitin-protein ligase MGRN1 |
| entrez gene/locuslink:23304 | Nbla10317 | Putative uncharacterized protein Nbla10317 |
| entrez gene/locuslink:23325 | WASHC4 | WASH complex subunit 4 |
| entrez gene/locuslink:23365 | null | null |
| entrez gene/locuslink:23397 | NCAPH | Condensin complex subunit 2 |
| entrez gene/locuslink:23533 | PIK3R5 | Phosphoinositide 3-kinase regulatory subunit 5 |
| entrez gene/locuslink:23550 | null | null |
| entrez gene/locuslink:23644 | EDC4 | Enhancer of mRNA-decapping protein 4 |
| entrez gene/locuslink:2444 | FRK | Tyrosine-protein kinase FRK |

Fig. 12. Extraction of Gene_Name and Protein_Name Using Apache Drill

## VI. CONCLUSION

It is concluded from the above discussion that an integrated NoSQL data model for protein-protein, protein-gene, and gene-disease interactions can help researchers to get insights of biomolecule networks. The data model can return all important factors that can take part for interactions such as gene ID, Gene name, gene location, gene code, protein name, protein structure, disease ID, and disease name all at one place. The proposed data model provides best storage format for this type of data sets (that are huge, complex and unstructured) to overcome the limitations of relational databases. This model has been implemented for 8000 different entries of all defined interactions and obtained search results are faster end effective than existing data models. This data model is an organized compilation of genes, proteins, and diseases from all known available resources to relate different factors amongst them. Apache drill queries written for proposed data model are easy to implement on any biomolecular dataset of this type. Drill provides users/researchers an opportunity of column-wise querying, to get values from required column/s and non-relating entries against that particular queried value will not be displayed. Future work may involve unifying all gene-phenotypes associations for the diseases or other important features such as treatment of diseases or environmental risk factors that cause gene mutations.

## REFERENCES

[1] Gavin, A.-C., et al., Functional organization of the yeast proteome by systematic analysis of protein complexes. Nature, 2002. 415(6868): p. 141-147.

[2] Ho, Y., et al., Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. Nature, 2002. 415(6868): p. 180-183.

[3] Rolland, T., et al., A proteome-scale map of the human interactome network. Cell, 2014. 159(5): p. 1212-1226.

[4] Tong, A.H.Y., et al., Systematic genetic analysis with ordered arrays of yeast deletion mutants. Science, 2001. 294(5550): p. 2364-2368.

[5] Uetz, P., et al., A comprehensive analysis of protein–protein interactions in Saccharomyces cerevisiae. Nature, 2000. 403(6770): p. 623-627.

[6] Huttlin, E.L., et al., The BioPlex network: a systematic exploration of the human interactome. Cell, 2015. 162(2): p. 425-440.

[7] Yang, X., et al., Widespread expansion of protein interaction capabilities by alternative splicing. Cell, 2016. 164(4): p. 805-817.

[8] Ito, T., et al., A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proceedings of the National Academy of Sciences, 2001. 98(8): p. 4569-4574.

[9] Zanzoni, A., et al., MINT: a Molecular INTeraction database. FEBS letters, 2002. 513(1): p. 135-140.

[10] Xenarios, I., et al., DIP: the database of interacting proteins. Nucleic acids research, 2000. 28(1): p.289-291.

[11] Mewes, H.-W., et al., MIPS: a database for genomes and protein sequences. Nucleic acids research, 2002. 30(1): p. 31-34.

[12] Wilson, N., Human protein reference database. Nature Reviews Molecular Cell Biology, 2004. 5(1): p. 4-4.

[13] Isserlin, R., R.A. El-Badrawi, and G.D. Bader, The biomolecular interaction network database in PSI-MI 2.5. Database, 2011. 2011: p. baq037.

[14] Hermjakob, H., et al., IntAct: an open source molecular interaction database. Nucleic acids research, 2004. 32(suppl 1): p. D452-D455.

[15] Croft, D., et al., Reactome: a database of reactions, pathways and biological processes. Nucleic acids research, 2010: p. gkq1018.

[16] Hadoop, A., Apache hadoop. 2011.

[17] HBase, A., Apache HBase. 2013, October.

[18] 18.Thusoo, A., et al., Hive: a warehousing solution over a map-reduce framework. Proceedings of the VLDB Endowment, 2009. 2(2): p. 1626-1629.

[19] Chaurasia, G., et al., UniHI: an entry gate to the human protein interactome. Nucleic acids research, 2007. 35(suppl 1): p. D590-D594.

[20] Wu, C.H., et al., The Universal Protein Resource (UniProt): an expanding universe of protein information. Nucleic acids research, 2006. 34(suppl 1): p. D187-D191.

[21] Chen, J.Y., S. Mamidipalli, and T. Huan, HAPPI: an online database of comprehensive human annotated and predicted protein interactions. BMC genomics, 2009. 10(1): p. S16.

[22] Stark, C., et al., BioGRID: a general repository for interaction datasets. Nucleic acids research, 2006. 34(suppl 1): p. D535-D539.

[23] Saeed, F. Big data proteogenomics and high-performance computing: Challenges and opportunities. in Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on. 2015. IEEE.

[24] Lehne, B. and T. Schlitt, Protein-protein interaction databases: keeping up with growing interactomes. Human genomics, 2009. 3(3): p. 291.

[25] Zhang, C., P. Gu, and W. Feng. Transform biological data into HBase with MDA. in Computer Science and Network Technology (ICCSNT), 2015 4th International Conference on. 2015. IEEE.

[26] Xu, D., and Y. Xu, Protein databases on the internet. Current Protocols in Protein Science, 2004: p. 2.6. 1-2.6. 15.

[27] Taylor, R.C., An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. BMC bioinformatics, 2010. 11(12): p. S1.

[28] Sarwar, M.A., A. Rehman, and J. Ferzund, Database Search, Alignment Viewer and Genomics Analysis Tools: Big Data for Bioinformatics. International Journal of Computer Science and Information Security, 2016. 14(12): p. 317.

[29] M. U. Ali, S. Ahmad, and J. Ferzund, "Harnessing the Potential of Machine Learning for Bioinformatics using Big Data Tools," International Journal of Computer Science and Information Security (IJCSIS), vol. 14, no. 10, pp. 668-675, 2016.

[30] S. Ahmed, M. U. Ali, J. Ferzund, M. A. Sarwar, A. Rehman and A. Mehmood, "Modern Data Formats for Big Bioinformatics Data Analytics," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 8, no. 4, 2017.

[31] Rehman, A. Abbas, M. A. Sarwar and J. Ferzund, "Need and Role of Scala Implementations in Bioinformatics," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 08, no. 02, 2017.