Project 1: Data Warehouse / OLAP System

CSE 601: Data Mining and Bioinformatics

AVIJEET MISHRA
(AVIJEETM@BUFFALO.EDU)
UB#:50169242

PRITHVI GOLLU INDRAKUMAR
(PGOLLUIN@BUFFALO.EDU)
UB#:50169089

DEPT OF COMPUTER SCIENCE,
UNIVERSITY AT BUFFALO

Contents

Introduction

We have designed and developed a data warehouse based on the logical data model named “BioStar” which deals with biomedical data described in the paper “BioStar models of clinical and genomic data for biomedical data warehouse design”. This data warehouse incorporates the datasets of the biomedical field for the study of human diseases.

PART 1: Implement your data warehouse schema in the Oracle system.

The given data was cleaned as per requirement and converted to .xlsx format for importing it into the data warehouse. The six data spaces clinical data space, sample data space, microarray data space, proteomic data space, experiment data space, and gene data space was efficiently modeled in the data warehouse.

Each data space was created using the following tables.

1. Clinical data space

Tables: patient, disease, diagnosis, drug, druguse, testresult, clinicaltest and patientsample

2. Sample data space

Tables: clinicalsample, geneticmarker, geneticscreen, biochemassay, assayresult, sampleanatomy and anatomyterm

3. Microarray and proteomic data space

Tables: mrnaexpression, arrayprobe, genesequence and measurementunit

4. Gene data space

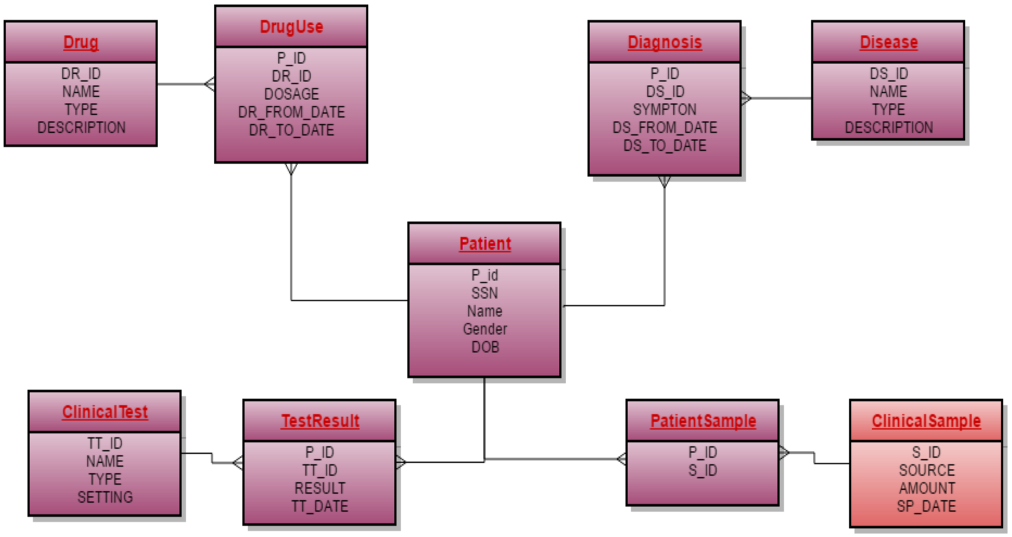
Tables: genecluster, clustermaster, goterm, goannotation, genesequence, genepromoter, promoter, proteininteract, genedomain and domainmodel

5. Experiment data space

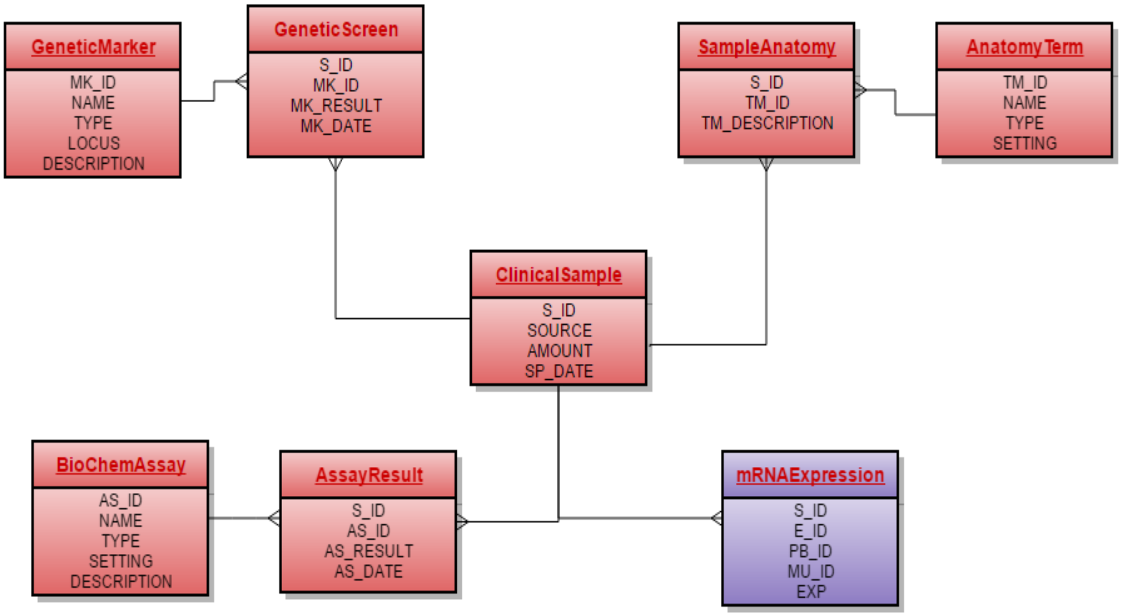
Tables: experimentmaster, project, platform

DataWarehouse Schema

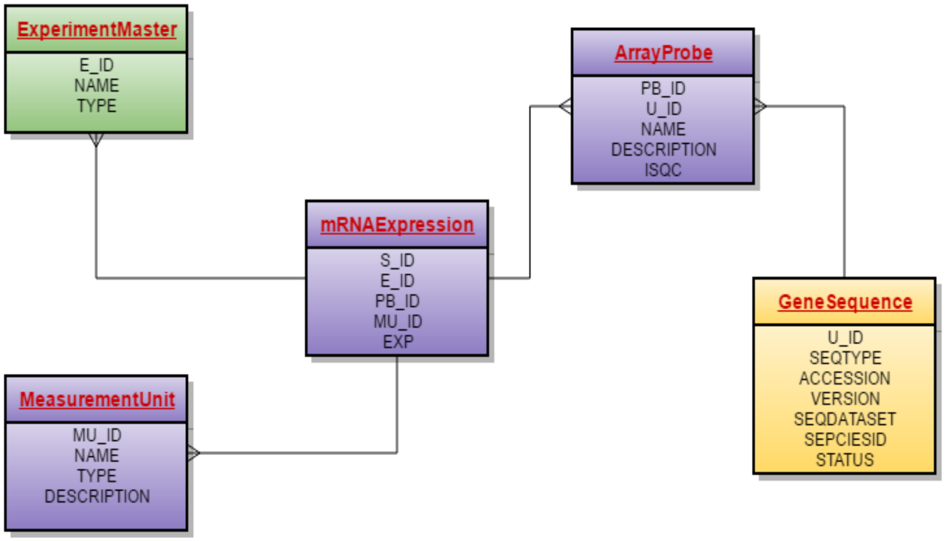
- Clinical data space



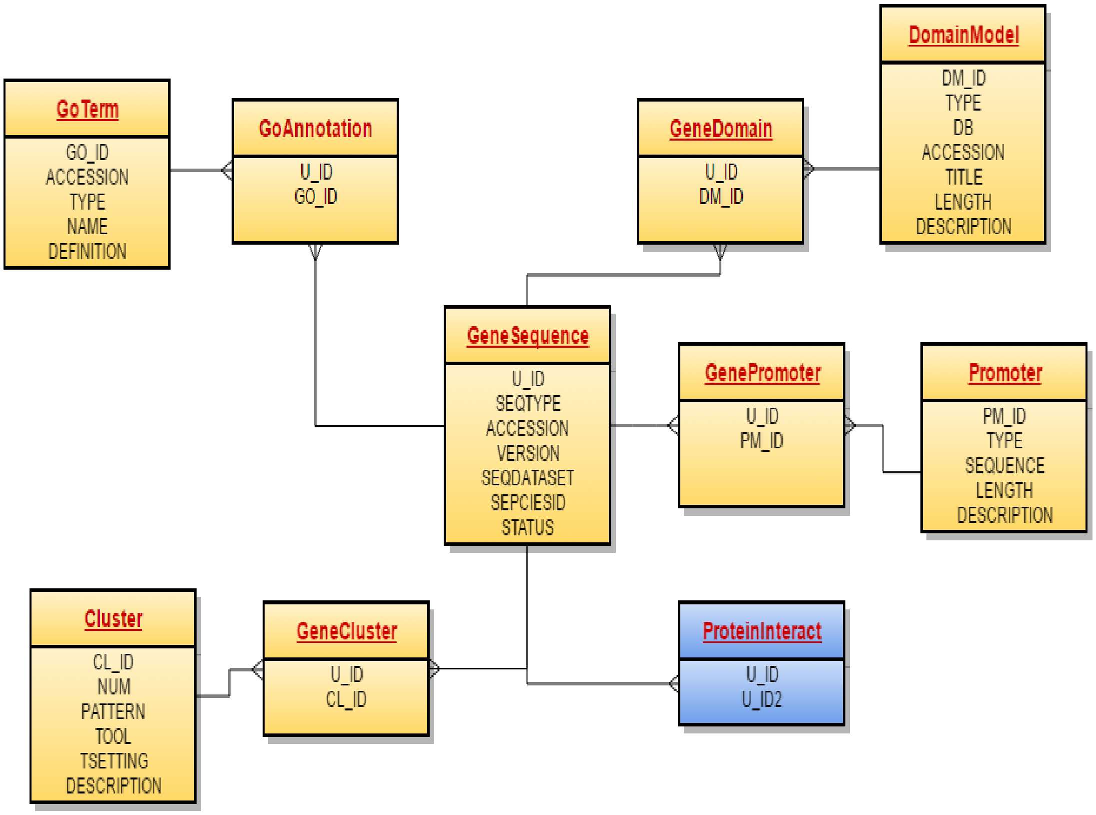
- Sample data space



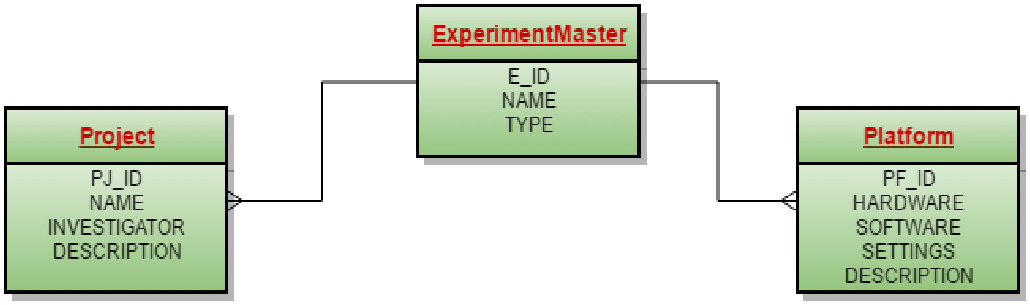
- Microarray and proteomic data space



- Gene data space



- Experiment data space



The multi-dimensional view of all these 6 data spaces was captured using the “BioStar Schema”.

Optimization: We have also used few connector tables as an enhancement to Bio-star schema to increase optimization. All tables have a primary key or multiple primary key pairs. Also the tables are linked using primary key and foreign key constraints. So by creating indexes on the fields in the tables, instead of using Linear Search which on an average requires N/2 block accesses, we use other searching techniques such as Binary Search which has log2 N block accesses. Creating an index on a field in a table creates another data structure which holds the field value, and pointer to the record it relates to. The index structure is then sorted, allowing Binary Searches to be performed on it, there by optimizing query retrieval. We implemented the data warehouse schema in the Oracle Sql Developer system and populated it with the provided data sets.

By cleaning the data and storing only the relevant fields, we have made sure that during a query, the database is accessed only once. That is by using primary index, in a single go all the required data is being retrieved. Though the OLAP layer adds an additional amount of complexity.

PART 2: OLAP and Statistical Operations

The OLAP and statistical operations was implemented on top of the Oracle database in C# using Microsoft Visual Studio as a platform. All the queries are made dynamic by using dropdown list and checkboxes, i.e. we can select the constraints in the queries such as the disease name, go_id dynamically which is then queried from the data warehouse.

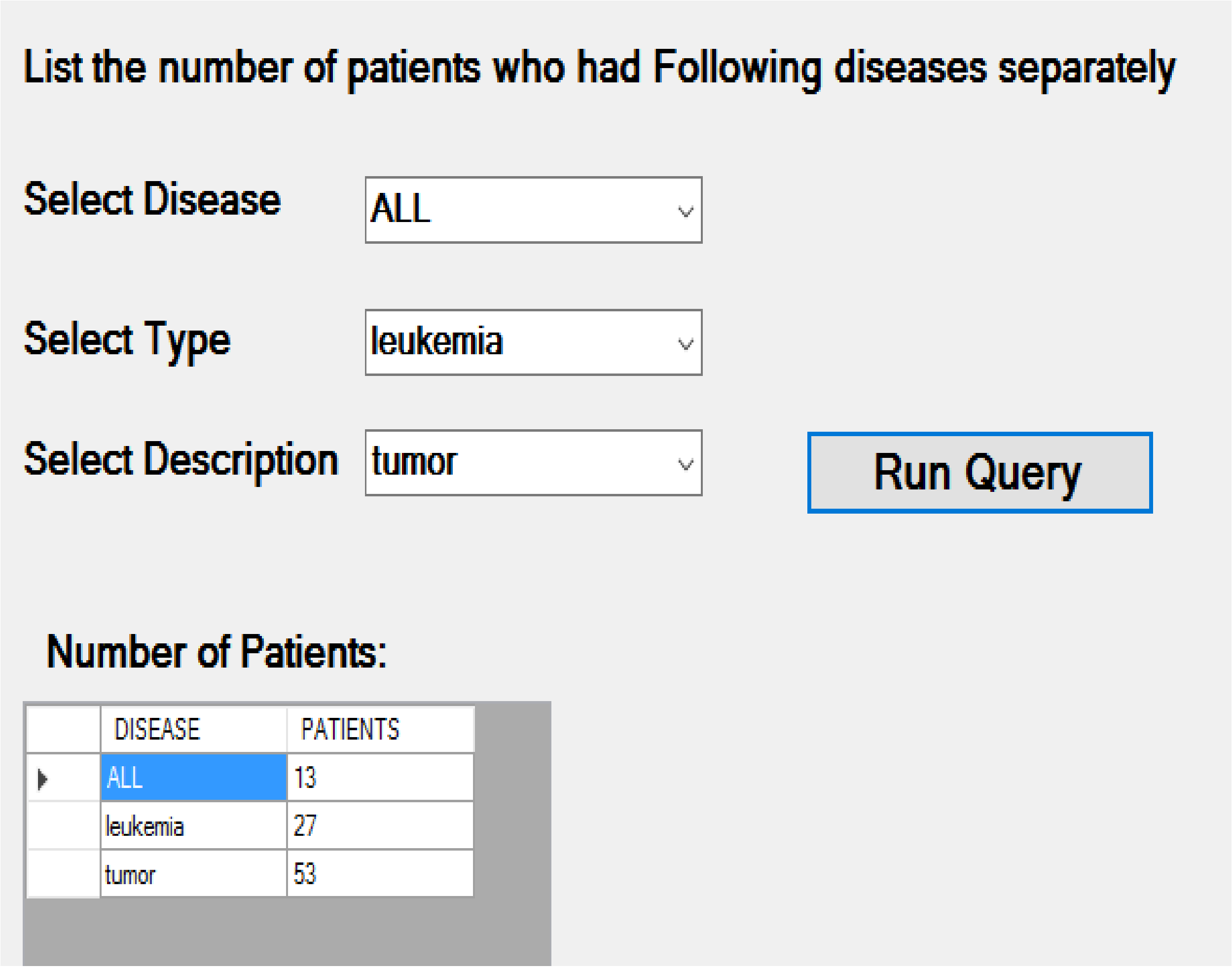
Below is the list of queries along with its Sql code and its output as shown by the UI.

1: List the number of patients who had "tumor" (disease description), "leukemia" (disease type) and "ALL" (disease name), separately.

The Time Complexity of this query is O(log(m). log(n)), where m and n represent number of tuples in disease and diagnosis respectively. Here it is Log because we have used indexes which reduces the complexity form m to log m.

Query:

```
SELECT b.description Disease, count(a.P_ID) Patients FROM diagnosis a , disease b WHERE a.ds_id = b.ds_id AND b. description = 'tumor' GROUP BY b.description
UNION
SELECT b.type, count(a.P_ID) FROM diagnosis a ,disease b WHERE a.ds_id = b.ds_id AND b.type = 'leukemia' GROUP BY b.type
UNION
SELECT b.name, count(a.P_ID) FROM diagnosis a , disease b WHERE a.ds_id = b.ds_id AND b.name = 'ALL' GROUP BY b.name
```



List the number of patients who had Following diseases separately

Select Disease: ALL

Select Type: leukemia

Select Description: tumor

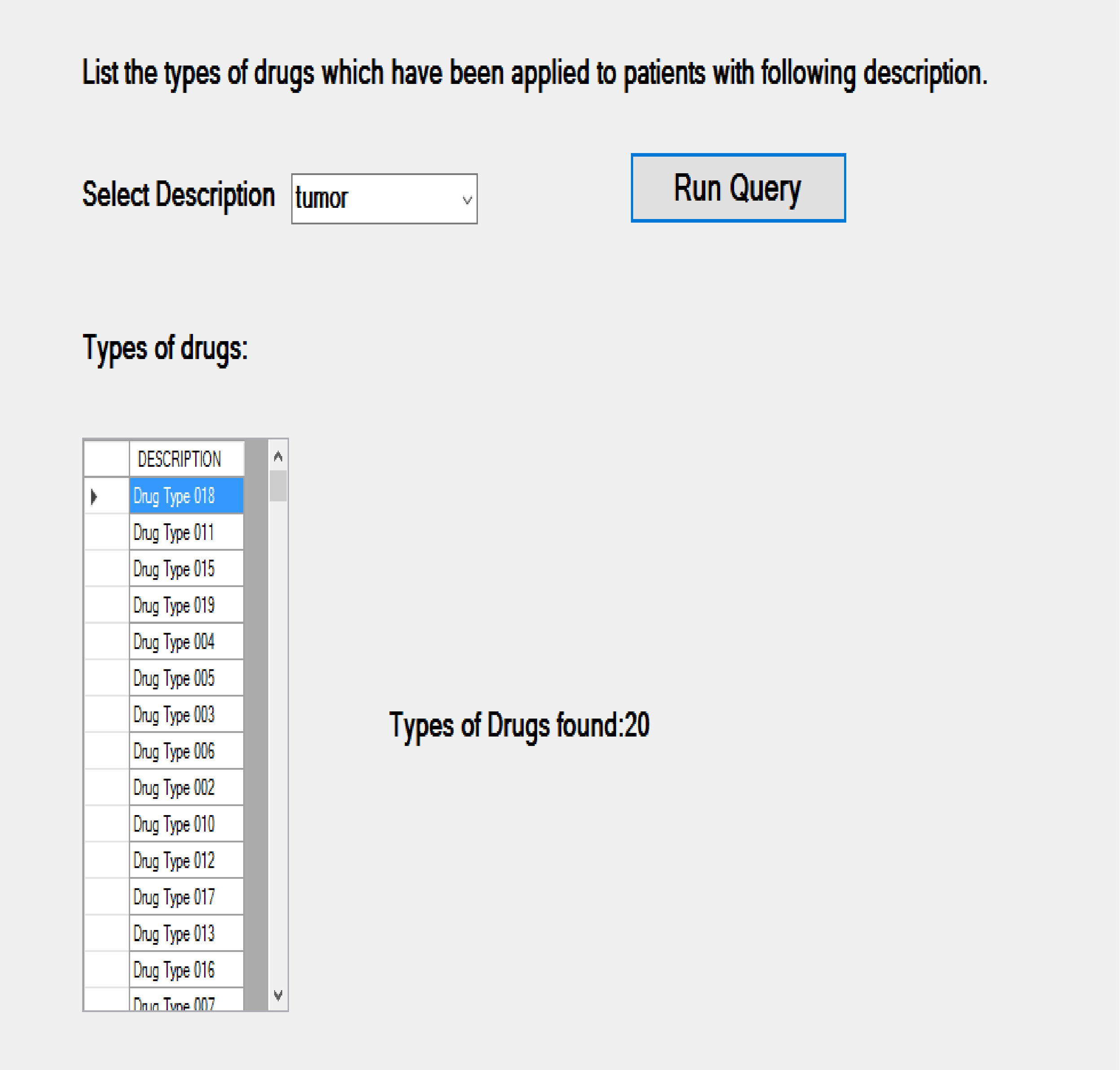
Number of Patients:

DISEASE	PATIENTS
ALL	13
leukemia	27
tumor	53

2: List the types of drugs which have been applied to patients with "tumor".

The Time Complexity of this query is O(log(m).log(n)), where m and n represent number of tuples in druguse and diagnosis respectively.

Query: *SELECT DISTINCT type FROM drug WHERE DR_id IN (SELECT a.dr_id FROM druguse a WHERE a.p_id IN (SELECT DISTINCT b.p_id FROM diagnosis b, disease c WHERE b.ds_id = c.ds_id AND c.description = 'tumor'))*



List the types of drugs which have been applied to patients with following description.

Select Description tumor

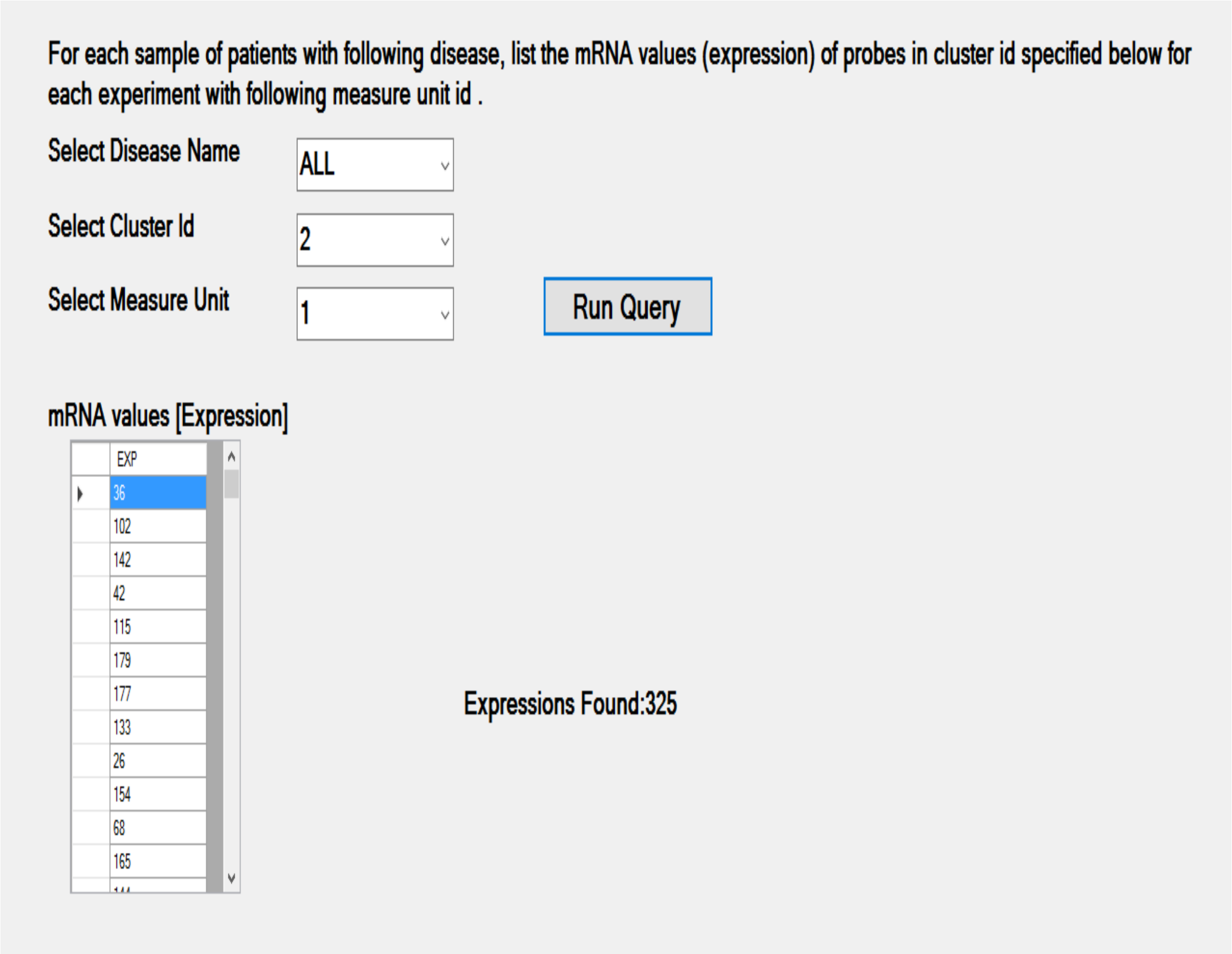
Types of drugs:

	DESCRIPTION
▶	Drug Type 018
	Drug Type 011
	Drug Type 015
	Drug Type 019
	Drug Type 004
	Drug Type 005
	Drug Type 003
	Drug Type 006
	Drug Type 002
	Drug Type 010
	Drug Type 012
	Drug Type 017
	Drug Type 013
	Drug Type 016
	Drug Type 007

Types of Drugs found:20

3: For each sample of patients with "ALL", list the mRNA values (expression) of probes in cluster id "00002" for each experiment with measure unit id = "001".

Query: *SELECT exp FROM mrnaexpression WHERE s_id IN (SELECT s_id FROM clinicalsample WHERE s_id IN (SELECT s_id FROM patientsample WHERE p_id IN(SELECT* *p_id FROM diagnosis WHERE ds_id IN(SELECT ds_id FROM disease WHERE name = 'ALL')))) AND pb_id IN (SELECT pb_id FROM arrayprobe WHERE u_id IN (SELECT u_id FROM genecluster WHERE cl_id = '00002')) AND mu_id = '001'*



For each sample of patients with following disease, list the mRNA values (expression) of probes in cluster id specified below for each experiment with following measure unit id .

Select Disease Name: ALL

Select Cluster Id: 2

Select Measure Unit: 1

mRNA values [Expression]

	EXP
▸	36
	102
	142
	42
	115
	179
	177
	133
	26
	154
	68
	165

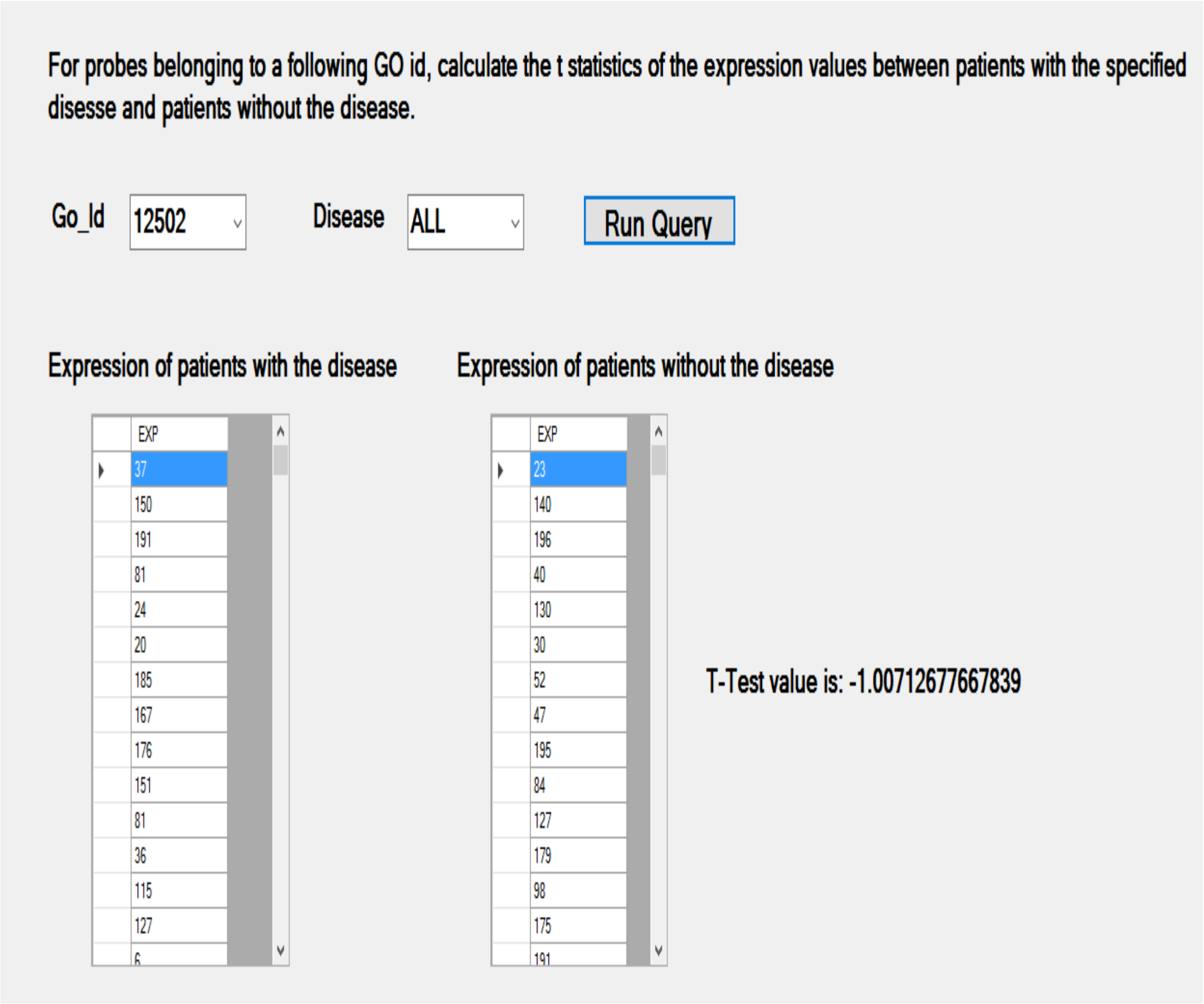
Expressions Found:325

4: For probes belonging to GO with id = "0012502", calculate the t statistics of the expression values between patients with "ALL" and patients without "ALL".

Query1: *SELECT exp FROM mrnaexpression WHERE s_id IN (SELECT s_id FROM clinicalsample WHERE s_id IN(SELECT s_id FROM patientsample WHERE p_id IN(SELECT p_id FROM diagnosis WHERE ds_id IN(SELECT ds_id FROM disease WHERE name =*

'ALL ''')))) AND pb_id IN(SELECT pb_id FROM arrayprobe WHERE u_id IN(SELECT u_id FROM goannotation WHERE go_id = '0012502 '))

Query2: *SELECT exp FROM mrnaexpression WHERE s_id IN (SELECT s_id FROM clinicalsample WHERE s_id IN(SELECT s_id FROM patientsample WHERE p_id IN(SELECT p_id FROM diagnosis WHERE ds_id IN(SELECT ds_id FROM disease WHERE name != 'ALL ''')))) AND pb_id IN(SELECT pb_id FROM arrayprobe WHERE u_id IN(SELECT u_id FROM goannotation WHERE go_id = '0012502 '))*



For probes belonging to a following GO id, calculate the t statistics of the expression values between patients with the specified disesse and patients without the disease.

Go_Id: 12502

Disease: ALL

Expression of patients with the disease

	EXP
▶	37
	150
	191
	81
	24
	20
	185
	167
	176
	151
	81
	36
	115
	127
	6

Expression of patients without the disease

	EXP
▶	23
	140
	196
	40
	130
	30
	52
	47
	195
	84
	127
	179
	98
	175
	191

T-Test value is: -1.00712677667839

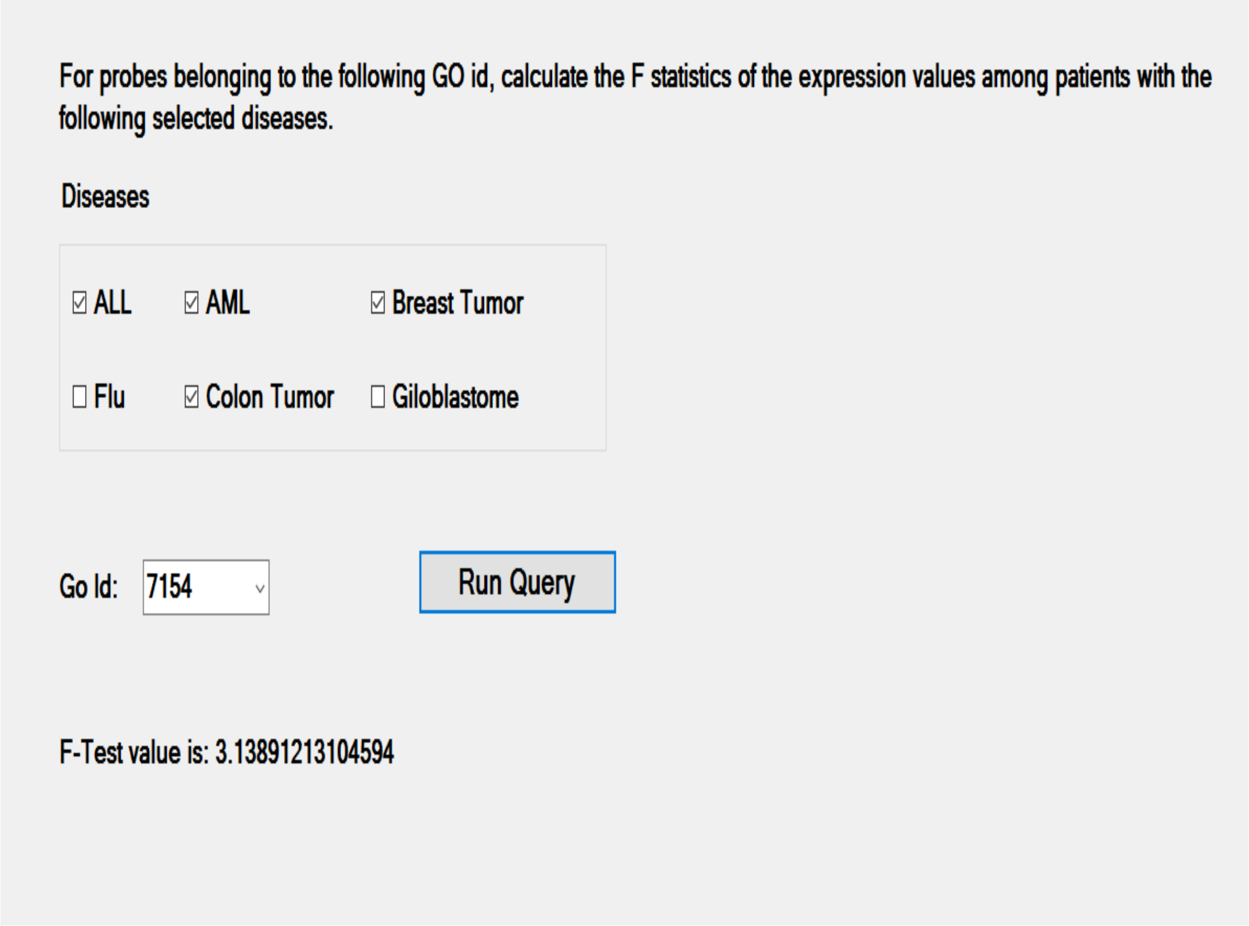
5: For probes belonging to GO with id=“0007154”, calculate the F statistics of the expression values among patients with “ALL”, “AML”, “Colon tumor” and “Breast tumor”.

Query1: *SELECT exp FROM mrnaexpression WHERE s_id IN (SELECT s_id FROM clinicalsample WHERE s_id IN(SELECT s_id FROM patientsample WHERE p_id IN(SELECT p_id FROM diagnosis WHERE ds_id IN(SELECT ds_id FROM disease WHERE name = 'ALL’')))) AND pb_id IN(SELECT pb_id FROM arrayprobe WHERE u_id IN(SELECT u_id FROM goannotation WHERE go_id = '0007154 '))*

Query2: *SELECT exp FROM mrnaexpression WHERE s_id IN (SELECT s_id FROM clinicalsample WHERE s_id IN(SELECT s_id FROM patientsample WHERE p_id IN(SELECT p_id FROM diagnosis WHERE ds_id IN(SELECT ds_id FROM disease WHERE name = 'AML’')))) AND pb_id IN(SELECT pb_id FROM arrayprobe WHERE u_id IN(SELECT u_id FROM goannotation WHERE go_id = '0007154 '))*

Query3: *SELECT exp FROM mrnaexpression WHERE s_id IN (SELECT s_id FROM clinicalsample WHERE s_id IN(SELECT s_id FROM patientsample WHERE p_id IN(SELECT p_id FROM diagnosis WHERE ds_id IN(SELECT ds_id FROM disease WHERE name = 'Colon tumor’')))) AND pb_id IN(SELECT pb_id FROM arrayprobe WHERE u_id IN(SELECT u_id FROM goannotation WHERE go_id = '0007154 '))*

Query4: *SELECT exp FROM mrnaexpression WHERE s_id IN (SELECT s_id FROM clinicalsample WHERE s_id IN(SELECT s_id FROM patientsample WHERE p_id IN(SELECT p_id FROM diagnosis WHERE ds_id IN(SELECT ds_id FROM disease WHERE name = 'Breast tumor’'’')))) AND pb_id IN(SELECT pb_id FROM arrayprobe WHERE u_id IN(SELECT u_id FROM goannotation WHERE go_id = '0007154 '))*



For probes belonging to the following GO id, calculate the F statistics of the expression values among patients with the following selected diseases.

Diseases

☑ ALL ☑ AML ☑ Breast Tumor

☐ Flu ☑ Colon Tumor ☐ Giloblastome

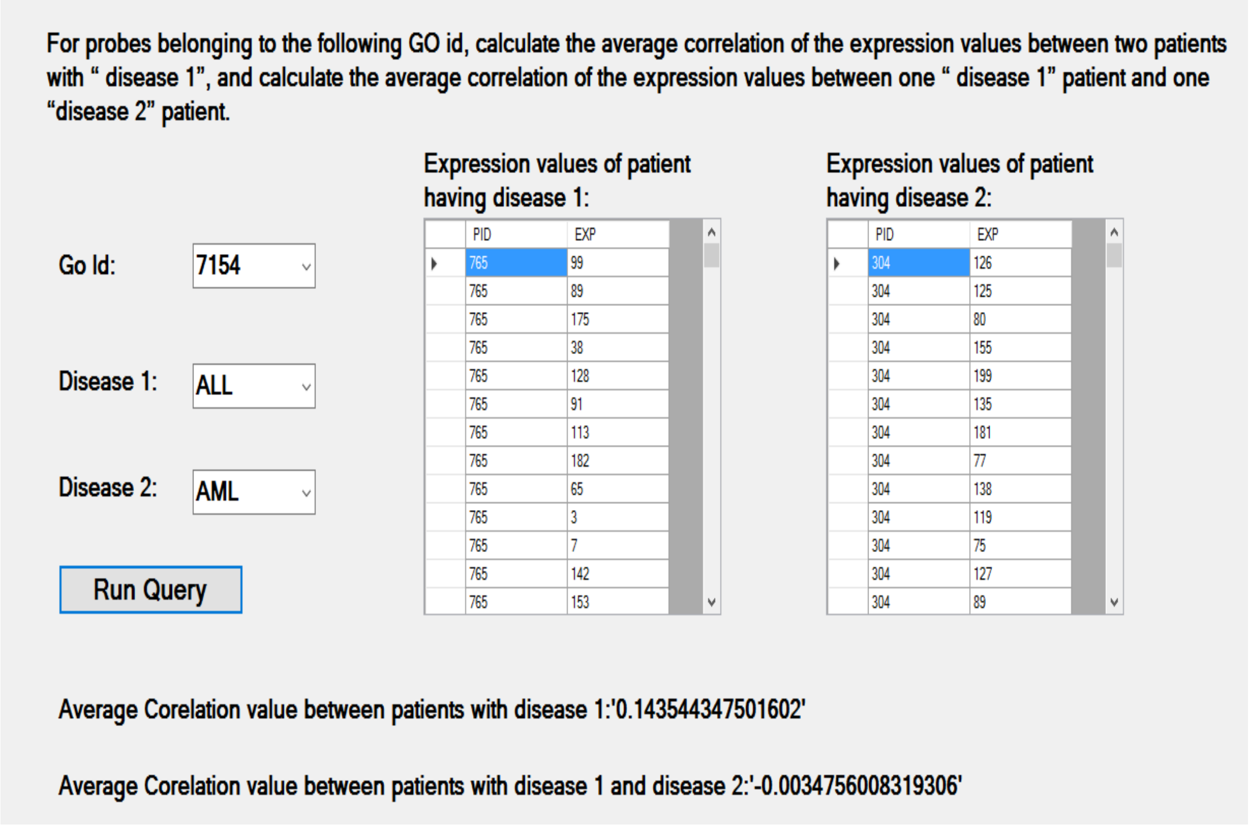
Go Id: 7154

F-Test value is: 3.13891213104594

6: For probes belonging to GO with id="0007154", calculate the average correlation of the expression values between two patients with "ALL", and calculate the average correlation of the expression values between one "ALL" patient and one "AML" patient.

Query1: *SELECT d.p_id, mn.exp FROM mrnaexpression mn, clinicalsample cs, patientsample ps, patient p, diagnosis d WHERE mn.s_id = cs.s_id AND cs.s_id = ps.s_id AND ps.p_id = p.p_id AND p.p_id = d.p_id AND d.ds_id IN (SELECT ds_id FROM disease WHERE name = 'ALL') AND mn.pb_id IN (SELECT ap.pb_id FROM arrayprobe ap,genesequence gs,goannotation ga WHERE ap.u_id = gs.u_id AND gs.u_id = ga.u_id AND ga.go_id ='0007154') ORDER by d.p_id, mn.pb_id*

Query2: *SELECT d.p_id, mn.exp FROM mrnaexpression mn, clinicalsample cs, patientsample ps, patient p, diagnosis d WHERE mn.s_id = cs.s_id AND cs.s_id = ps.s_id AND ps.p_id = p.p_id AND p.p_id = d.p_id AND d.ds_id IN (SELECT ds_id FROM disease WHERE name = 'AML') AND mn.pb_id IN (SELECT ap.pb_id FROM arrayprobe ap,genesequence gs,goannotation ga WHERE ap.u_id = gs.u_id AND gs.u_id = ga.u_id AND ga.go_id ='0007154') ORDER by d.p_id, mn.pb_id*



For probes belonging to the following GO id, calculate the average correlation of the expression values between two patients with " disease 1", and calculate the average correlation of the expression values between one " disease 1" patient and one "disease 2" patient.

Go Id: 7154

Disease 1: ALL

Disease 2: AML

Expression values of patient having disease 1:

PID	EXP
765	99
765	89
765	175
765	38
765	128
765	91
765	113
765	182
765	65
765	3
765	7
765	142
765	153

Expression values of patient having disease 2:

PID	EXP
304	126
304	125
304	80
304	155
304	199
304	135
304	181
304	77
304	138
304	119
304	75
304	127
304	89

Average Corelation value between patients with disease 1:'0.143544347501602'

Average Corelation value between patients with disease 1 and disease 2:'-0.0034756008319306'

PART 3: Knowledge Discovery

By utilizing the data warehouse, OLAP operations and statistical operations such as T-statistic and correlation, we are able to gain knowledge about the informative genes for any particular disease. This can be used to classify if new patients have the disease or not.

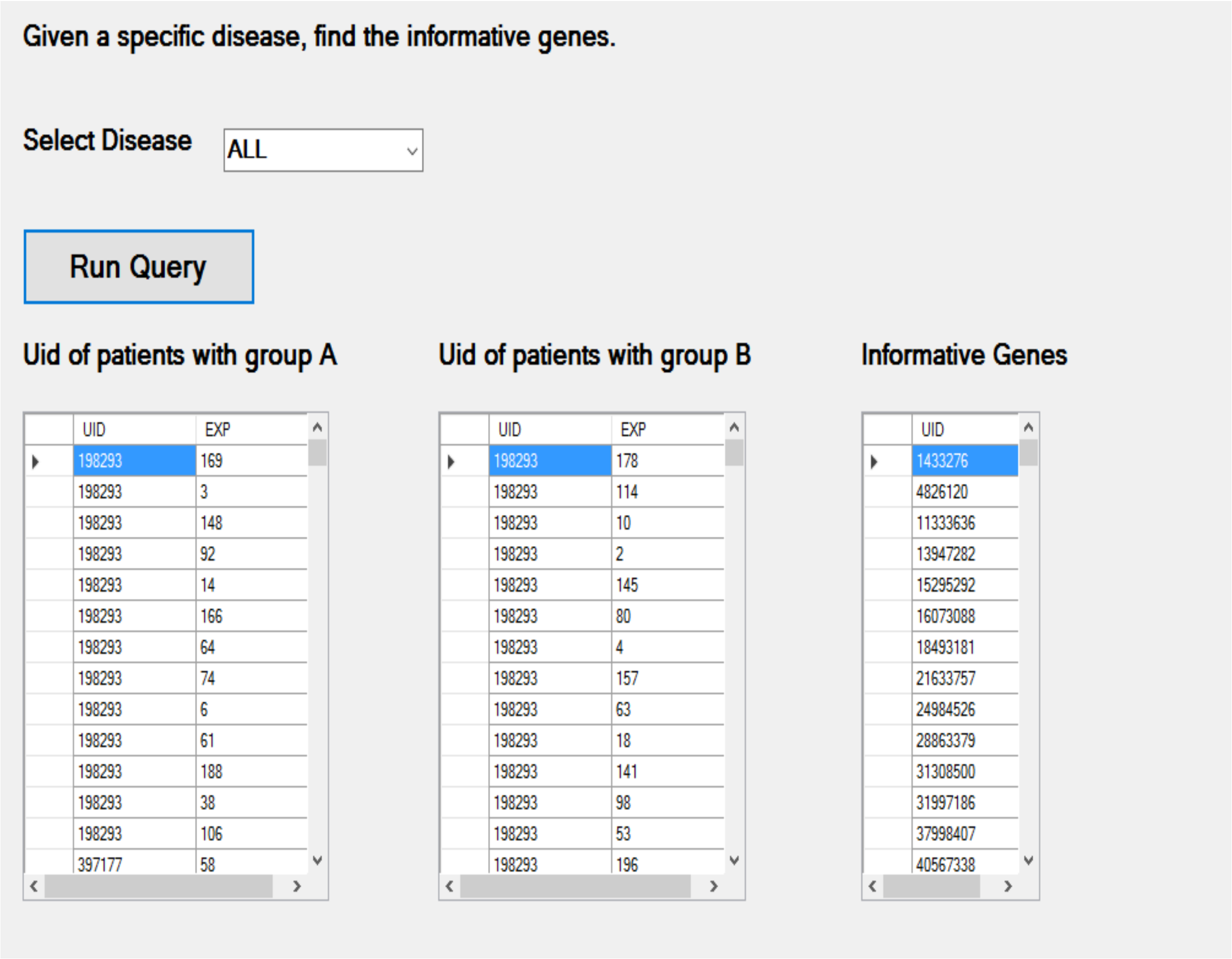
1: Given a specific disease, find the informative genes.

To find the informative genes, first we found the list of patients who have the disease and those who don't have it. Then for each gene, we calculated T- statistics for the expression values which was available in table mRNAExpression between both the lists. Based on the P-value (smaller than 0.01), we segregated the genes as informative. Here informative genes for the disease "ALL" has been calculated. The user can dynamically select which disease he wants to calculate the informative genes.

Query1: *SELECT ap.u_id, mn.exp FROM mrnaexpression mn INNER JOIN arrayprobe ap ON mn.pb_id=ap.pb_id WHERE mn.s_id IN (SELECT s_id FROM clinicalsample WHERE s_id IN (SELECT s_id FROM patientsample WHERE p_id IN (SELECT p_id FROM diagnosis WHERE ds_id IN (SELECT ds_id FROM disease WHERE name = 'ALL'))))ORDER BY ap.u_id";*

Query2: *SELECT ap.u_id, mn.exp FROM mrnaexpression mn INNER JOIN arrayprobe ap ON mn.pb_id=ap.pb_id WHERE mn.s_id IN (SELECT s_id FROM clinicalsample WHERE s_id IN (SELECT s_id FROM patientsample WHERE p_id IN (SELECT p_id FROM diagnosis WHERE ds_id IN (SELECT ds_id FROM disease WHERE name != 'ALL'))))ORDER BY ap.u_id";*

The informative genes for the user selected disease are displayed along with the UID's and expression values from patients with and without the disease as shown below.



Given a specific disease, find the informative genes.

Select Disease: ALL

Uid of patients with group A

UID	EXP
198293	169
198293	3
198293	148
198293	92
198293	14
198293	166
198293	64
198293	74
198293	6
198293	61
198293	188
198293	38
198293	106
397177	58

Uid of patients with group B

UID	EXP
198293	178
198293	114
198293	10
198293	2
198293	145
198293	80
198293	4
198293	157
198293	63
198293	18
198293	141
198293	98
198293	53
198293	196

Informative Genes

UID
1433276
4826120
11333636
13947282
15295292
16073088
18493181
21633757
24984526
28863379
31308500
31997186
37998407
40567338

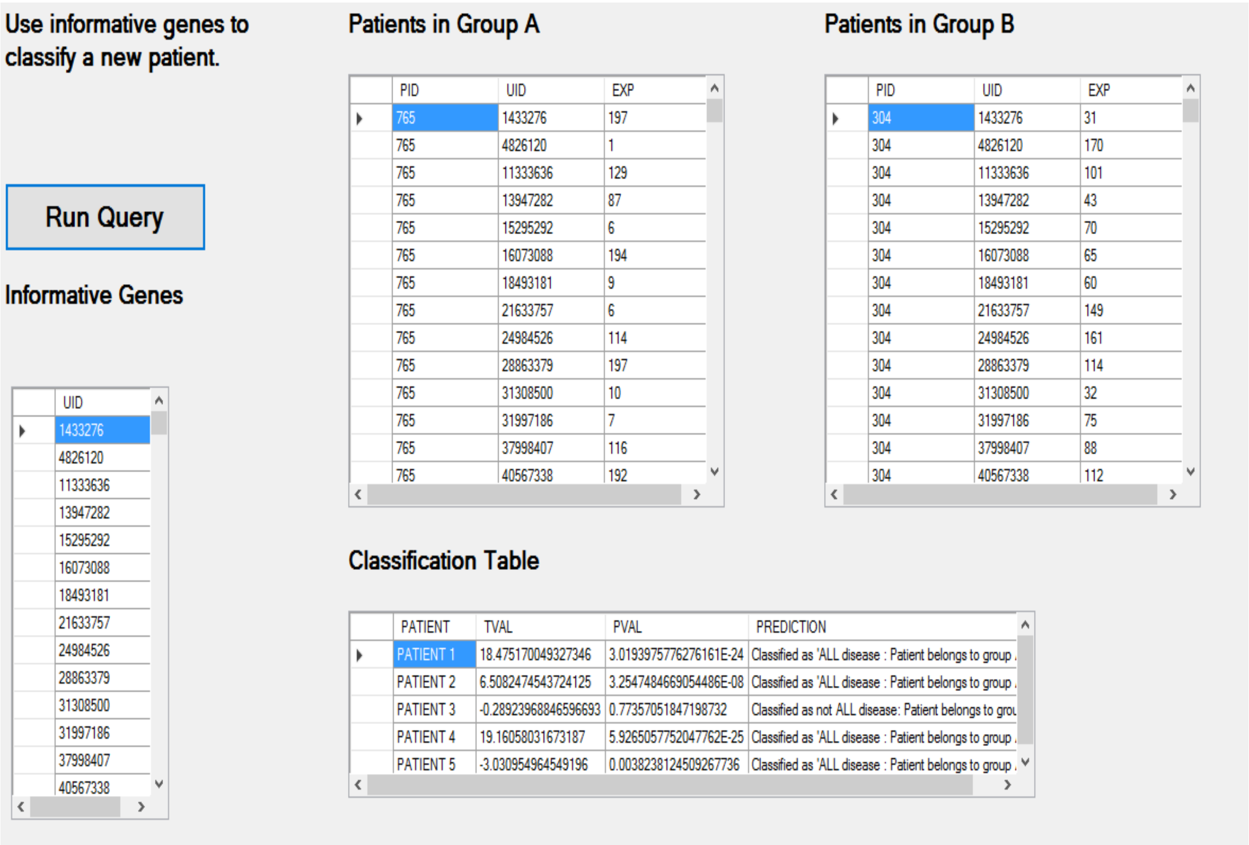
2: Use informative genes to classify a new patient.

The informative genes obtained from the previous section is used to classify whether new patients have the disease or not. For this we get UID and expression values from mRNAExpression table of patients who have the disease and patients who don't. Then we calculate the correlation between the new patients and each patient in the previously collected lists based on their expression values for UID's which represent the informative genes. This gives us 2 lists of correlation: 1 between new patient and patients who have the disease and another between new patient and patients who do not have the disease. T-Statistics is carried out on these 2 lists and if the obtained p-value is smaller than 0.01 then the new patient is classified as "has disease".

Query1: *SELECT patient.p_id, ap.u_id, mn.exp FROM mrnaexpression mn INNER JOIN arrayprobe ap ON mn.pb_id=ap.pb_id INNER JOIN (SELECT a.s_id, b.p_id FROM clinicalsample a, patientsample b WHERE a.s_id = b.s_id AND b.p_id IN (SELECT p_id FROM diagnosis WHERE ds_id IN (SELECT ds_id FROM disease WHERE name = 'ALL'))) patient ON mn.s_id = patient.s_id AND ap.u_id IN glob.dataglob.UID) ORDER BY patient.p_id,ap.u_id*

Query2: *SELECT patient.p_id, ap.u_id, mn.exp FROM mrnaexpression mn INNER JOIN arrayprobe ap ON mn.pb_id=ap.pb_id INNER JOIN (SELECT a.s_id, b.p_idn FROM clinicalsample a, patientsample b WHERE a.s_id = b.s_id AND b.p_id IN (SELECT p_id FROM diagnosis WHERE ds_id IN (SELECT ds_id FROM disease WHERE name != 'ALL'))) patient ON mn.s_id = patient.s_id AND ap.u_id IN glob.dataglob.UID) ORDER BY patient.p_id,ap.u_id*

Here glob.dataglob.UID is a variable which has the list of UID's of informative genes.



Informative Genes

UID
1433276
4826120
11333636
13947282
15295292
16073088
18493181
21633757
24984526
28863379
31308500
31997186
37998407
40567338

Patients in Group A

PID	UID	EXP
765	1433276	197
765	4826120	1
765	11333636	129
765	13947282	87
765	15295292	6
765	16073088	194
765	18493181	9
765	21633757	6
765	24984526	114
765	28863379	197
765	31308500	10
765	31997186	7
765	37998407	116
765	40567338	192

Patients in Group B

PID	UID	EXP
304	1433276	31
304	4826120	170
304	11333636	101
304	13947282	43
304	15295292	70
304	16073088	65
304	18493181	60
304	21633757	149
304	24984526	161
304	28863379	114
304	31308500	32
304	31997186	75
304	37998407	88
304	40567338	112

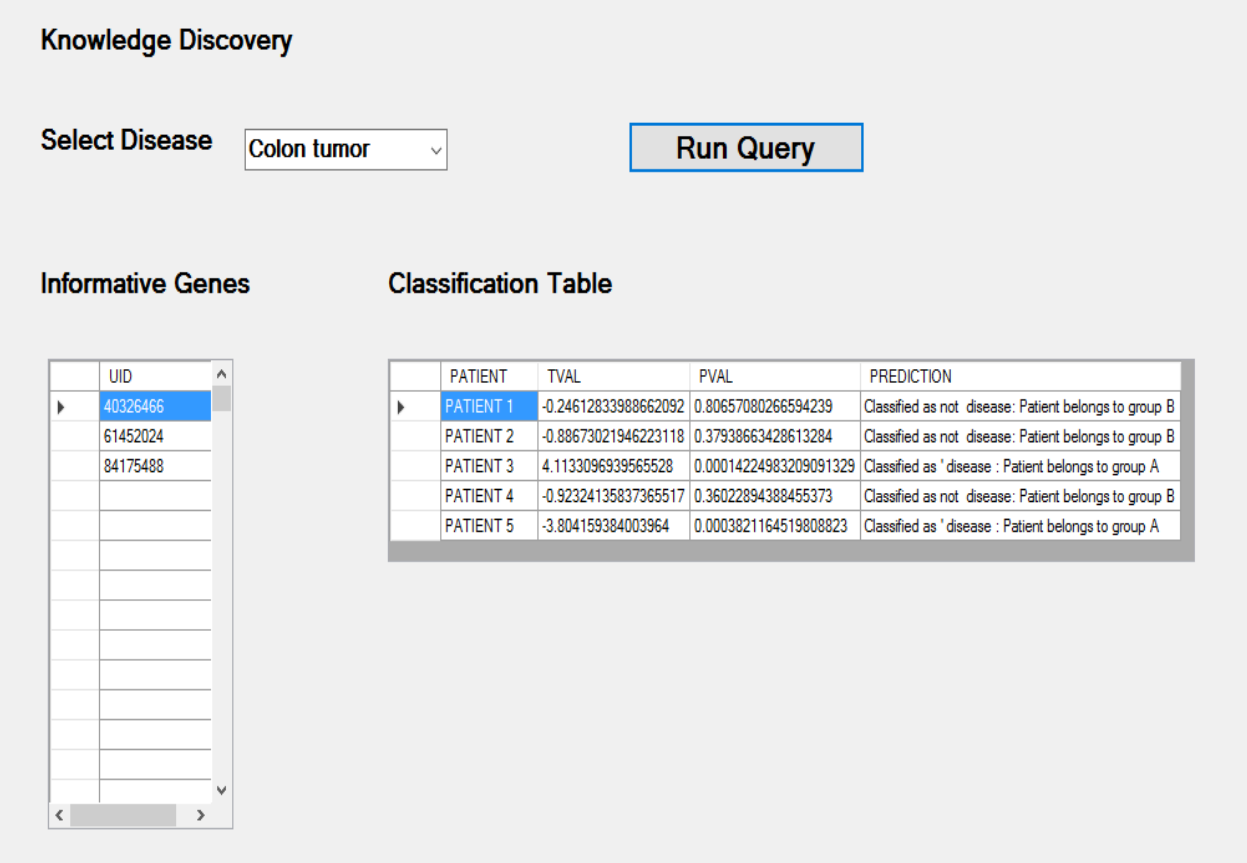
Classification Table

PATIENT	TVAL	PVAL	PREDICTION
PATIENT 1	18.475170049327346	3.0193975776276161E-24	Classified as 'ALL disease : Patient belongs to group .
PATIENT 2	6.5082474543724125	3.2547484669054486E-08	Classified as 'ALL disease : Patient belongs to group .
PATIENT 3	-0.28923968846596693	0.77357051847198732	Classified as not ALL disease: Patient belongs to grou
PATIENT 4	19.16058031673187	5.9265057752047762E-25	Classified as 'ALL disease : Patient belongs to group .
PATIENT 5	-3.030954964549196	0.0038238124509267736	Classified as 'ALL disease : Patient belongs to group .

Here except for Patient 2 all other patients have been classified as has “ALL” disease based on the informative genes obtained.

Knowledge Discovery

Classification of new patients for disease selected: Colon tumor based on the information genes obtained.



UID
40326466
61452024
84175488

PATIENT	TVAL	PVAL	PREDICTION
PATIENT 1	-0.24612833988662092	0.80657080266594239	Classified as not disease: Patient belongs to group B
PATIENT 2	-0.88673021946223118	0.37938663428613284	Classified as not disease: Patient belongs to group B
PATIENT 3	4.1133096939565528	0.00014224983209091329	Classified as ' disease : Patient belongs to group A
PATIENT 4	-0.92324135837365517	0.36022894388455373	Classified as not disease: Patient belongs to group B
PATIENT 5	-3.804159384003964	0.0003821164519808823	Classified as ' disease : Patient belongs to group A

Conclusion

We have efficiently implemented a biomedical data warehouse and an OLAP layer which carries out many OLAP and Statistical operations. Using these we were able to gain knowledge about the informative genes for any particular disease and classify if new patients have the disease or not. Also by data cleaning, indexing and efficient UI, we have optimized query retrieval.