Integrating genomics data with cancer registry records
Final Report

Jae Young Kim, Kevin Tsang

Project description

At the City of Hope Clinical Cancer Genomics Division, researchers collect patient medical records by administering Cancer Risk Questionnaires and the data is stored in a software called Progeny (Sybase). Meanwhile, patients in the registry also provide blood samples for the lab to examine their biological sequences, where outputs of mutation data are exported by the Bioinformatics pipeline.

In this project, we aim to connect the mutation data with the patient data in the cancer registry database. Our goal is to integrate these two data and construct a database system which can generate insight. We have created a query tool that allows researchers to answer research questions by correlating patient cancer history with specific gene mutations.

Description data sets (link to data sources, example contents and explain what data attributes mean)

Data Link :
https://drive.google.com/drive/folders/1e3KhRyGq5Ae5KPtccpnVDraB1rB1ooZo?usp=sharing

1.Cancer Registry data* (n = 28566)

Cancer Registry Data is extracted from Progeny and it contains Lab UPN, which is essentially the sample ID, Hormone Receptor status (Estrogen Receptor, Progesterone Receptor), HER2/neu status (A specific protein in cell growth) and up to 11 Cancer Diagnosis and Age of Diagnosis (Cancer types includes Breast, Colorectal, Lung, Skin, Ovarian, Gastric, Melanoma, Retinoblastoma, Thyroid). These are all tabular text data and each row represents one patient.

2.Genomics mutation data *

We have 20 spreadsheets worth of mutation data generated by Ingenuity Variant Analysis. The data is in a specific format that is ordered by variants. Each row represents a mutation / variant and it contains Chromosome position, Gene Symbol (Gene name), Transcript ID, Transcript Variant (mutation notation in DNA format), Protein Variant (mutation notation in Protein format) and Final Call (Annotation by genetics expert to determine if a mutation is Pathogenic, Likely Pathogenic, Uncertain Significance, Likely Benign or Benign). Each column after the main information has the sample ID embedded in a string and if it has “Het”, it means that sample contains the mutation. One sample can consist of multiple mutations.

* See example dataset for elaboration
https://drive.google.com/drive/folders/1e3KhRyGq5Ae5KPtccpnVDraB1rB1ooZo?usp=sharing

System design / architecture

After understanding the data, we have engineered a new data model by split the two data sets (genomics data and cancer registry data) into 1 table and 2 json files: mutation information json, patient mutation table and patient registry json. Below are the columns of each table:

I. Mutation information data (json file):
This json file shows the biological information of each mutation.
Columns: 'Mutation_ID'*, 'Chromosome', 'Position', 'Reference Allele', 'Sample Allele', 'Variation Type', 'Gene Region', 'Gene Symbol' (Gene name), 'Transcript ID' (transcript IDs from various biological databases, cannot use as key because it has multiple IDs for each mutation), 'Transcript Variant' (mutation notation in DNA format), 'Protein Variant' (mutation notation in Protein format)
* We have created Mutation_ID from Chromosome, Position, Reference Allele, and Sample Allele; this will truly create a unique ID for each mutation as we learned that more than one mutation can happen in a chromosome position. We will use Mutation_ID to connect with the Patient Mutation table.

II. Patient - Mutation table (csv / pandas dataframe):
This table shows which mutation the patient has. Note that Genotype, Compound Heterozygous, Read Depth, Allele Fraction depends on both Lab UPN (patient id) and Mutation id. Therefore, there is no redundancy for the mutation information described in the data. For each spreadsheet, there are "final call" and "all out" sheets; primarily, we extracted the majority of the data from the "all out" sheet, which comprises all variants found in a run. The "final call" includes final results manually curated by genetics experts. We have replaced the respective rows in the "all out" sheet with the "final call" data because they are more relevant to clinical management and patient care.

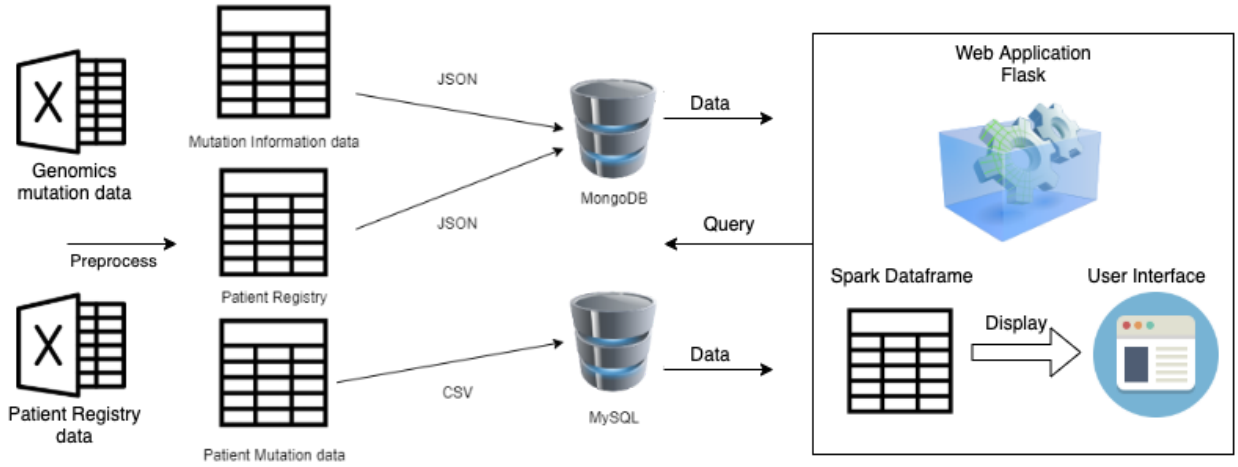
Columns : LabUPN (Sample unique ID), 'Mutation_ID*', 'Genotype', 'Compound Heterozygous', 'Read Depth', 'Allele Fraction', 'Classification', 'Curated' (Marked yes if it was manually curated by a genetics expert, marked no if the Classification was created by software algorithm.)

*Mutational_ID is used to connect with the Mutation information json and LabUPN connects the patient registry data.

III. Patient Registry data (json file)
This data shows the cancer history and various cancer biomarker statuses of patients in the City of Hope Registry.

'Patient_ID'*, 'Pedigree name', 'UPN', 'Lab UPN', 'ER', 'PR', 'Her2/Neu', 'Her2/Neu by FISH', 'Her2/Neu by IHC', "history" (cancer history: cancer type and age of diagnosis)

*Patient_ID is created from Pedigree name and UPN (Unique Patient Number); A pedigree is a family history diagram and since we have enrolled family members into the registry, we use UPN to identify each individual. Lab UPN is used to connect with Patient Mutation data.



Though the original plan of uploading Cancer Registry data and Genomics mutation data to two different databases is viable, we learned that we need to avoid redundancy in data storage. Also, due to the sparsity of the data, we have splitted the datasets and made it more accessible. As a result, we separated the mutation information (which is universal to the specific mutation) from the genomics mutation data into the mutation information table. Since many patients can share the same mutations, we avoided many data redundancy by just having a record of each mutation once. Patient mutation table then indicates how many mutations the patient has and its corresponding relevant statistics and information.

As we were preprocessing the data, we found that some of the genomics mutation data excel files are very large. Since we already have a considerable amount of data, we have decided to not use the data from excel spreadsheets Genomics_7, 10, 11, 12, 13, 14 to gather patient mutation data. This project is mainly a prototype for this data integration so we have decided to omit those data. For some of the files, the preprocessing would have taken more than 500 hours if we were to use our local computing power and memory (Time calculated by tqdm package). We acknowledge that we could have split those files into smaller files and used Hadoop MapReduce to perform data cleaning as a future improvement but we did not have the time to do so since we have learned it late in the semester.

After all the data cleansing, we have uploaded mutation information and patient registry data as JSON files to MongoDB while the patient mutation data is converted to a merged CSV file and uploaded to MySQL. We have connected the two databases to our web application written in Python Flask web development framework. As we query,

we use spark dataframe for data aggregation tasks such as group by and count. The responses we get from the databases are then displayed in the user interface (front-end) of the web application.

Data problems that have been addressed, evaluation and conclusion.

Data Cleaning : Removed unnecessary columns and created primary key by combining multiple columns. Cleansed unformatted columns. For example, changed Transcript ID, Transcript Variant to list format. Therefore, it was possible to search data including certain Transcript ID and Variant by using mongoDB query.

Transcript ID	Transcript Variant
NM_004565.2; ENST00000356607.8	c.84+10814C>T
NM_004565.2; ENST00000356607.8	c.84+10818T>C
NM_004565.2; ENST00000356607.8	c.84+10827A>G
NM_004565.2; ENST00000356607.8	c.84+10832C>A
NM_004565.2; ENST00000356607.8	c.84+10837A>G
...	...
NM_001139498.1; ENST00000441825.8; NM_00113950...	c.-39+58830C>T; c.50-222750C>T; c.50-73999C>T
NM_001139498.1; ENST00000441825.8; NM_00113950...	c.-39+58827G>T; c.50-74002G>T; c.50-222753G>T
NM_001139498.1; ENST00000441825.8; NM_00113950...	c.50-74002G>A; c.-39+58827G>A; c.50-222753G>A
NM_001139498.1; ENST00000441825.8; NM_00113950...	c.50-222754C>T; c.-39+58826C>T; c.50-74003C>T
NM_001139498.1; ENST00000441825.8; NM_00113950...	c.-39+58806A>G; c.50-222774A>G; c.50-74023A>G

⇒

Transcript ID	Transcript Variant
[NM_004565.2, ENST00000356607.8]	[c.84+10814C>T]
[NM_004565.2, ENST00000356607.8]	[c.84+10818T>C]
[NM_004565.2, ENST00000356607.8]	[c.84+10827A>G]
[NM_004565.2, ENST00000356607.8]	[c.84+10832C>A]
[NM_004565.2, ENST00000356607.8]	[c.84+10837A>G]
...	...
[NM_001139498.1, ENST00000441825.8, NM_0011395...	[c.-39+58830C>T, c.50-222750C>T, c.50-73999C>T]
[NM_001139498.1, ENST00000441825.8, NM_0011395...	[c.-39+58827G>T, c.50-74002G>T, c.50-222753G>T]
[NM_001139498.1, ENST00000441825.8, NM_0011395...	[c.50-74002G>A, c.-39+58827G>A, c.50-222753G>A]
[NM_001139498.1, ENST00000441825.8, NM_0011395...	[c.50-222754C>T, c.-39+58826C>T, c.50-74003C>T]
[NM_001139498.1, ENST00000441825.8, NM_0011395...	[c.-39+58806A>G, c.50-222774A>G, c.50-74023A>G]

[Picture] Example of data preprocessing

Data modeling : Splitted Genomics data into Patient mutation data and Mutation information data to reduce sparsity and redundancy of the data

Variant Fin	Case Sampl	Final Call	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her	28336_Her
130	1	Uncertain Significance																							
22	1	Uncertain Significance														Het									
20	1	Uncertain Significance																							
17	1	Uncertain Significance																							
16	1	Uncertain Significance																							
15	2	Uncertain Significance																							
14	1	Uncertain Significance																							
14	1	Uncertain Significance																				Het			
12	5	Uncertain Significance				Het	Het		Het																
12	3	Uncertain Significance					Het																		Het
11	10	Uncertain Significance											Het						Het				Het		
11	1	Uncertain Significance																							
11	3	Uncertain Significance																							
10	2	Uncertain Significance										Het													
10	2	Uncertain Significance																							
9	1	Uncertain Significance																							
9	2	Uncertain Significance																				Het			
9	5	Uncertain Significance											Het												
9	2	Uncertain Significance																							
9	2	Uncertain Significance																				Het			
8	1	Uncertain Significance																							
8	1	Uncertain Significance																							
8	1	Uncertain Significance																							
1	1	Pathogenic																							
1	1	Pathogenic														Het									
1	3	Pathogenic	Het						Het																
1	1	Likely Benign										Het													
1	1	Likely Benign									Het														
1	1	Pathogenic																					Het		
1	1	Likely Benign																							
1	1	Pathogenic						Het																	
1	1	Pathogenic																							
1	1	Likely Benign																							

[Picture] Sparsity of raw data

	Mutation_id	Allele_Fraction	Classification	Compound_Heterozygous	Curated	Genotype	Lab_UPN	Read_Depth
0	chr1:10566230_C-->A	2.67	Uncertain Significance	No	No	Het	CO22348-0257	150.0
1	chr1:10566272_C-->T	51.98	Benign	No	No	Het	CO22348-0257	177.0
2	chr1:45797154_G-->A	2.02	Uncertain Significance	No	No	Het	CO22348-0257	99.0
3	chr1:45797505_C-->G	47.12	Benign	No	No	Het	CO22348-0257	104.0
4	chr1:45797951_G-->X	0.81	Likely Pathogenic	No	No	Het	CO22348-0257	248.0
...	...	...	...	...	...	...	...	...
56425	chr22:40876274_G-->T	14.89	Uncertain Significance	No	No	Het	CO23882-0763	94.0
56426	chr22:40876278_X-->T	7.45	Benign	No	No	Het	CO23882-0763	94.0
56427	chr22:40876278_A-->T	7.45	Uncertain Significance	No	No	Het	CO23882-0763	94.0
56428	chr22:40876279_T-->X	17.02	Uncertain Significance	No	No	Het	CO23882-0763	94.0
56429	chr22:40876279_TT-->X	5.32	Uncertain Significance	No	No	Het	CO23882-0763	94.0

[Picture] After preprocessing

For example, the team changed data structure, so it became dense and relational.

Data storage : Used MongoDB and MySQL

Data integration : Integrated 10 files of raw data after preprocessing

Data Retrieval : Used spark dataframe linked with Flask(python web application) to search data and group by the result

Classification Count

Likely Benign	556
Benign	831
Uncertain Significance	7610
Pathogenic	583
Likely Pathogenic	206

[Picture] Example of the result of spark dataframe applying group by

Results - How did you meet the requirements?

For databases, we have used MySQL database (relational) to store patient mutation data and MongoDB (NoSQL) to store the mutation information and patient registry data. We have used spark to to perform data aggregation during our query process. We have loaded the mutation information and patient registry json files to MongoDB and patient mutation csv files to MySQL and used MySQL connector and MongoSpark connector to load data from MongoDB and MySQL to Spark by performing queries (such as selection, project, aggregation, etc). Specifically for classification count of genomics data, we use Spark to perform parallel processing in counting how many variants are in each category for patients using the group by function. Moreover, we have also used the same function to count the patient cancer history as well. These results utilize the fast cluster computing that Spark offers.

We have developed an intuitive interface for searching and exploring the data in the database using Python Flask web development framework. In our search tool, we included patient query, mutation information query and variant query. Exploration can be done with keyword search in all three of the queries. For patient query, we enter the LabUPN of the patient we are interested in and it outputs that patient's ID, age, cancer history and diagnosis age. Also we used Spark to calculate the number of mutations for that patient group by classification. Classification is the five-tier classification system recommended by The American College of Medical Genetics and Genomics (ACMG) to interpret sequence variants (Benign, Likely Benign, Variant of Uncertain Significance, Likely Pathogenic and Pathogenic.) Last but not least, it lists out the list of mutations the patient has and it contains characteristics of the mutation that pertains to the patient.

We have also embedded a hyperlink in the table to get more information of the mutation which essentially shares the same results as the mutation information query. That is, if users search for a specific mutation by its ID, they will get the same mutation information. The information includes gene symbol, gene region, chromosome number, protein variant and transcript variant. These serve as unique markers for a specific mutation. To elucidate the semantics of this, we can think of protein variant and transcript variant as addresses of a geographical location. Gene serves as the context or in this case, it can be the Country of the address. Just as multiple countries can have the same street name, multiple genes can have the same protein variant or transcript variant. Therefore, mutation information is very crucial in identifying the correct mutation.

Our most important query, variant query, utilizes all the above to generate insights for research. By using Protein Variant, Transcript ID, Transcript Variant and Gene, we can see how many patients have that specific mutation and see how that mutation is classified. Also, we have included the distribution of those patients' cancer history diagnosis to help us better understand the correlation between a specific mutation with various cancer types. In this search function, it is flexible as users do not have to input all four keywords to get a result. This feature does take longer since the search can be very broad and output huge amounts of data. Conventionally, researchers would query the specific variant by its gene and protein variant or transcript variant; this would significantly lower the time it takes to generate results.

How can we practically use this application in real life applications?

We can practically use this application for research purposes. First of all, data integration enriches the knowledge we have for the patients and can greatly improve researchers efficiency. With the query tool, we can see at a glance how many mutations a patient has and how many of them are pathogenic and need attention. With the variant query, we can use the distribution of the cancer types in a specific variant search and develop some statistics to characterize gene mutations. Potentially, we can identify new mutations that correlate with the diagnosis of specific cancer types. This is very beneficial for patient care since doctors and genetics counsellors can develop preventative treatment options for patients who can be pre-screened for those pathogenic mutations prior to them having cancer.

Explain unique features of your project.

Our project is very unique because it uses real company data at City of Hope and it has a very practical application to the research world in Clinical Cancer Genomics. With some specific modifications, we can see it being implemented at City of Hope as a research query tool. We also had to remodel the datasets to reduce the sparsity and redundancy to store the data into MongoDB and MySQL. The dataset is also very large, with over a million rows of mutation data and about 30000 patient data. It took three minutes to load one csv file into pandas dataframe and all the while for data cleansing, it takes significant time to perform each operation.

Discuss your learning experiences, collaboration efforts, challenges met on the project.

At first, since Kevin has domain knowledge about this complex dataset, Jae needed time to learn about the dataset from Kevin. After understanding data, the team could interpret the data structure. Since the structure of the raw data was not well-structured, we had to restructure the data to reduce sparsity and redundancy. In the process, the team could split genomics data into two parts; Mutation_information and Mutation_patient data. Moreover, the team did not have enough knowledge about web application, the team spent time learning how to build web application from scratch. The team worked very well together despite having geographical timezone limitations as Jae resides in South Korea and Kevin is in the United States. The team collaborates using Zoom and its screen sharing feature to conduct working meetings while other asynchronous messaging is done by Slack. This project was a huge opportunity for both to learn database management, data cleaning and modeling and communication. Kevin and Jae both benefited a lot from this project.

Team members and what each member has done

Background:
Jae Young Kim - B.S. in Mathematics
M.S. Applied Data Science
Background skills: Have several project experience in machine learning, Deep learning
https://www.linkedin.com/in/jae-young-kim

Kevin Tsang - B.S. in Biochemistry; working as a Clinical Research Assistant at City of Hope who has access to the data needed in this project
M.S. Applied Data Science

Background skills: Domain knowledge in Clinical Cancer Genomics, exposure to web development (Python Django REST framework)
https://www.linkedin.com/in/hkerkevin/

Clean Genomics Table and upload to MySQL - KT, JK
Clean Cancer Registry data and upload to MongoDB - KT, JK
Clean Mutation information data and upload to MongoDB - KT, JK
Integration of the two data - KT, JK
----------Midterm ----------
Build database and upload data - JK
Build webapp to show query using Flask - KT, JK

Data Link :
https://drive.google.com/drive/folders/1e3KhRyGq5Ae5KPtccpnVDraB1rB1ooZo?usp=sharing
Data after preprocessing Link:
https://drive.google.com/drive/folders/1cR-vOAWz4IYO-Y1mFmNm47COgVK4Iof9?usp=sharing
Web Demo Link:
https://www.youtube.com/watch?v=YSEwtz3yo0M
Github link for code
https://github.com/jeayoung114/DSCI-551-project

Appendix

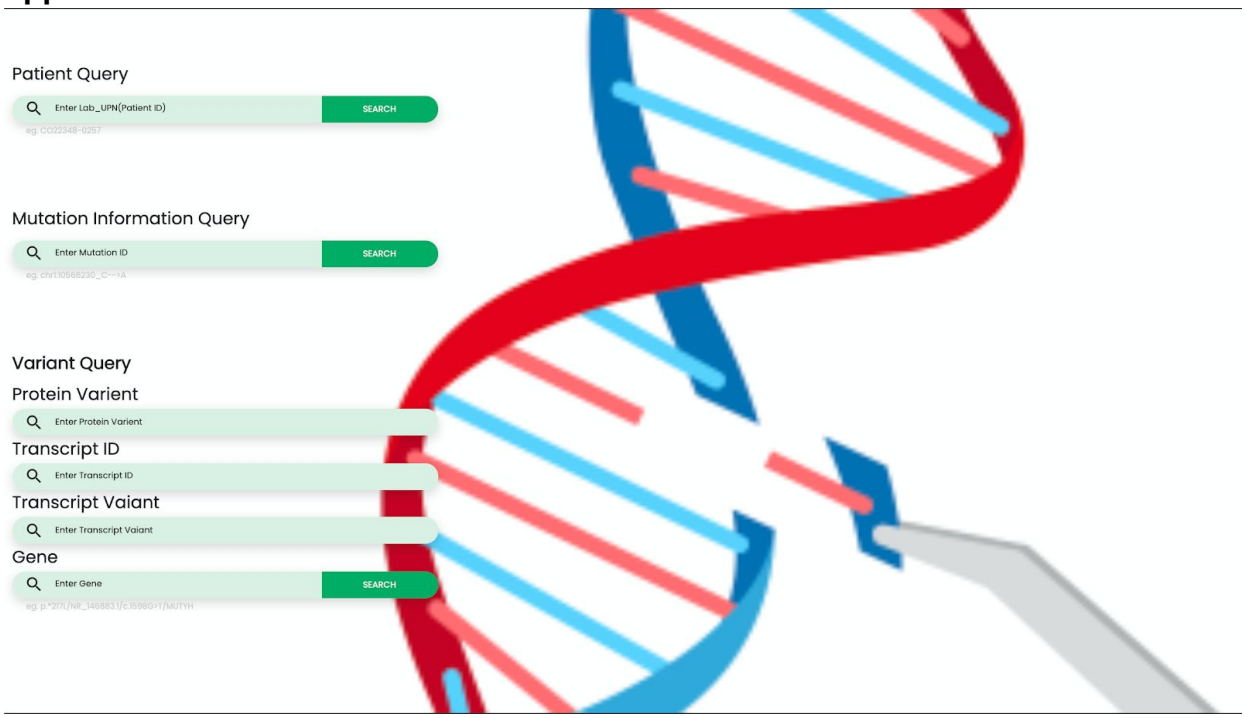


Image of web application user interface for patient, mutation information and variant queries.

Cancer Type Count

Thyroid	1
Uterine (Endometrial)	1
Breast	4

Mutation List

#	Mutation_id	Allele_Fraction	Classification	Compound_Heterozygous	Curated	Genotype	Lab_UPN	Read_Depth
1	chr1:45794979_C-->A	1.64	Uncertain Significance	Yes	No	Het	PE22545-0388	122.0
2	chr1:45794979_C-->A	0.96	Uncertain Significance	Yes	No	Het	ME17406-0972	208.0
3	chr1:45794979_C-->A	0.8	Uncertain Significance	Yes	No	Het	CO23892-0444	251.0
4	chr1:45794979_C-->A	1.16	Uncertain Significance	Yes	No	Het	ME17429-0449	258.0
5	chr1:45794979_C-->A	0.53	Uncertain Significance	Yes	No	Het	MT26139-0684	376.0
6	chr1:45794979_C-->A	1.08	Uncertain Significance	Yes	No	Het	MX26452-0190	278.0

Patient Information

#	ER	Her2_Neu	Her2_Neu by FISH	Her2_Neu by IHC	Her2_Neu by unknown method	ID	Lab_UPN	PR	Pedigree_name	UPN
1	0	0	0	0	0	56242-0388_27	PE22545-0388	0	56242-0388	27
2	0	0	0	0	0	62032-0972_1	ME17406-0972	0	62032-0972	1
3	Positive	Negative	0	0	0	73435-0190_1	MX26452-0190	Positive	73435-0190	1
4	0	0	0	0	0	57508-0444_1	CO23892-0444	0	57508-0444	1
5	0	0	0	0	0	62055-0449_1	ME17429-0449	0	62055-0449	1

Image of sample query output for protein variant : p.*217L, Transcript ID: NR_146883.1, Transcript Variant : c.1598G>T and Gene : MUTYH