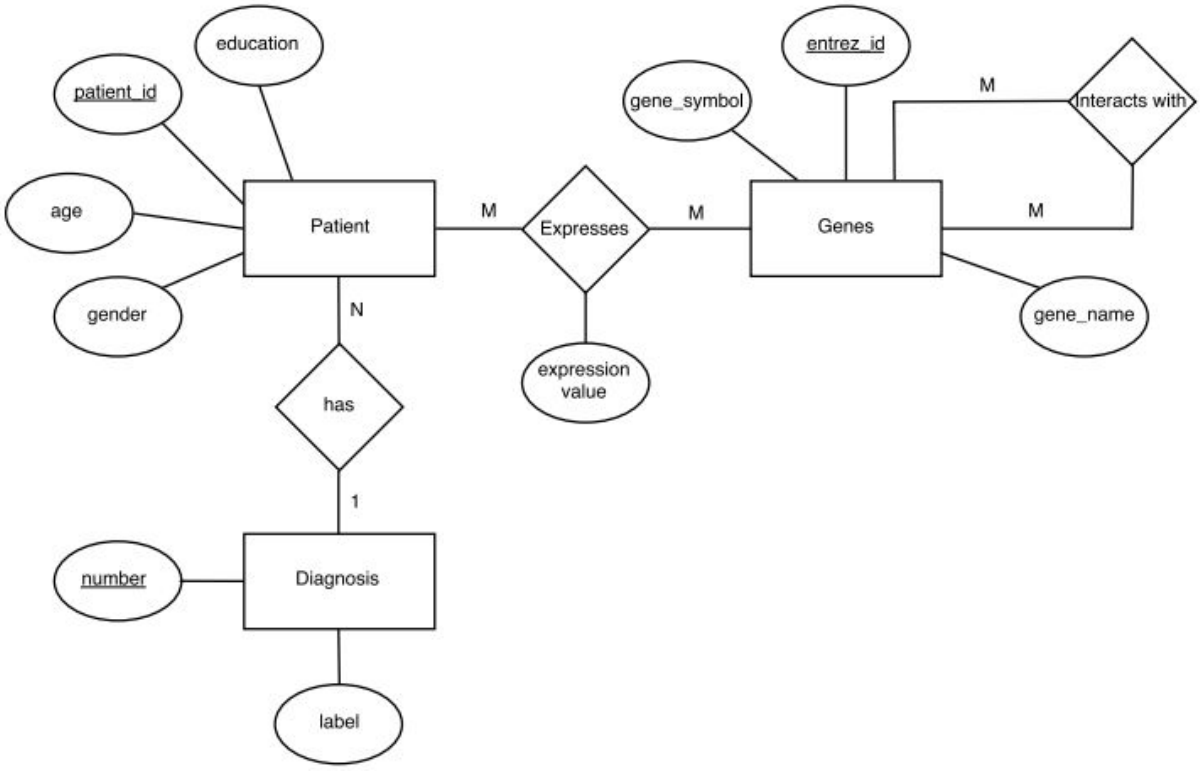
AD Knowledge Base

The AD Knowledge base attempts to correlate patient's age, gender, education, and genetic makeup with the onset of Alzheimer's. To do that genetic and patient data need to be merged in order to run analysis the dataset to see which gene when expressed at what threshold leads to an onset of Alzheimer's.

Design

We used three different DBMS which each suits a different need for different problems that needed to be solved.

The first choice was obviously going to be a traditional relational database. Most medical databases use relational databases as it is an industry standard, and our database would be supported by their legacy systems. While slower, and less optimal for large data sets, they do have advantages for smaller problems. MySQL is secure which is a plus for keeping patient information secure, because of its rigid schema it is also easy to link information from multiple tables.



While less optimal for graph traversal such as in the problem of finding all the n-order genes due to the amount of queries it needs to do, it can still do it in a timely manner for a non graph based system. For queries that don't need much power such as: finding information on a patient however which is just linking two tables together, MySQL is a cost effective and effective DBMS to use.

MongoDB was another choice, it has well-documented file import and basic-statistics functions that makes it a suitable choice to analyze gene expression values grouped by disease type and is highly scalable. With just one command it is possible to upload the whole dataset and analyze it later with intuitive API.

Patient Document Template
Patient ID : value
Disease : value
entrez #1 : value
entrez #2 : value
entrez #... : value
entrez #n : value

Cassandra allows for flexibility in columns which is optimal for gene annotations. Certain genes may have more information than others stored in additional columns. Hence, Cassandra is a natural choice for this part of the project.

The data is stored in a table where each row can have different number of columns. Each row represents a gene, while columns represent its properties.

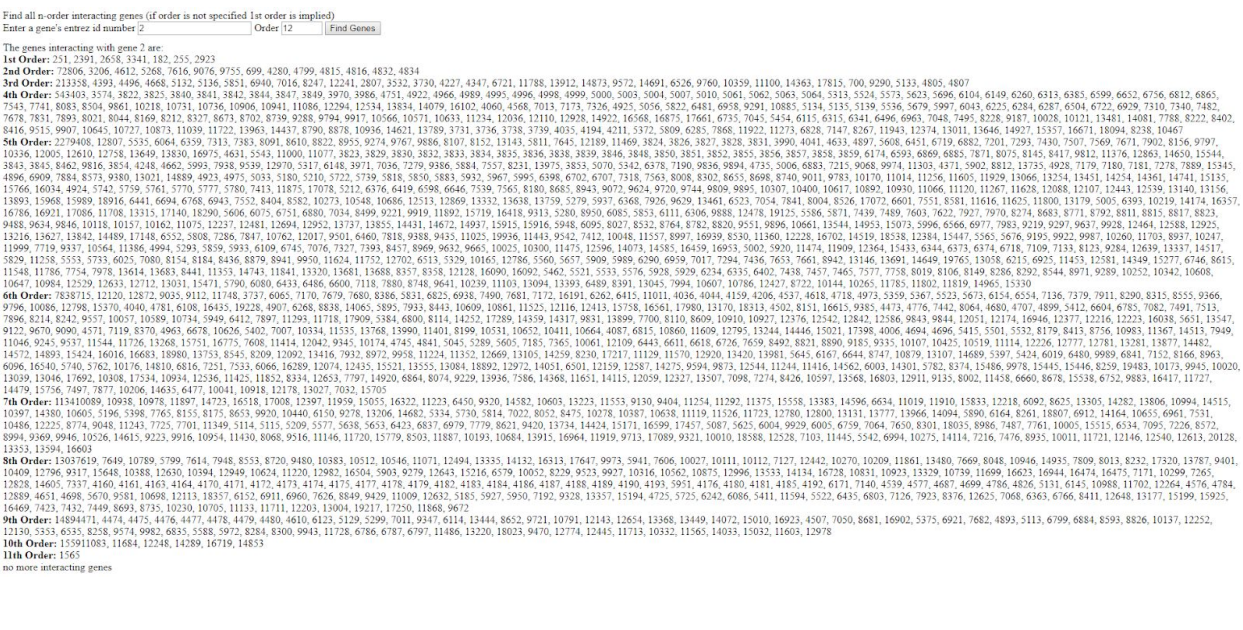
UNIPROT	ENTREZ	PROTEIN_NAME	ATTRIBUTE(1)	...	ATTRIBUTE(n)
Value	Value	Value	Value 1	**NULL**	**NULL**
Value	Value	Value	**NULL**	Value	Value
NULL	Value	**NULL**	Value	Value	Value
Value	Value	Value	**NULL**	Value	Value

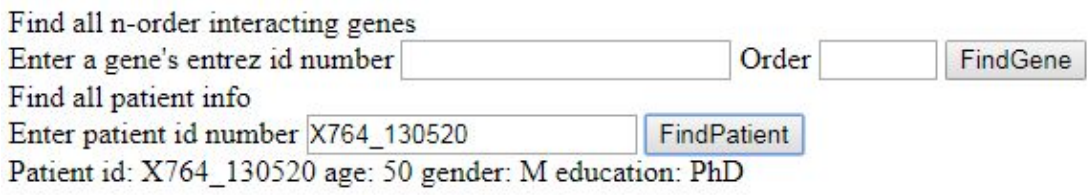
Implementation

For the MySQL database, all the data from the csv was loaded into their corresponding table using python. While a solution would be to also use python to create simple scripts that would run specific queries, a better solution would be to create an interface that integrate them all into one clean easy to use graphical user interface. AD Knowledge Base is hosted on a LAMP (Linux, Apache, MySQL, PHP) stack on a remote web server. The interface itself is simple and barebones, which makes it easy to use. When the user inputs data, it sends an AJAX request to the backend to run a query depending on which button was pressed. Each button sends a different value to the back end so it can differentiate between clicks.

Getting Patient Data was a simple query to get the row that corresponded to a `patient_id`. Getting n-order gene interactions, however was slightly more complicated. The implementation decided on was an array based dynamic programming algorithm. The initial node is the queried gene. First first order interactions are found through the initial query. The results are thrown into an array that represents the connecting nodes. Then those nodes are queried/visited and their connections thrown into the array. After all the connecting nodes are visited, we use `array_unique()` which removes duplicates from an array, to make sure we don't visit a node more than once. We repeat this until we have either completed the final order, or there are no more unvisited connecting nodes.

For example in the diagram below, we want to get the 12th order genes for gene 2. We get 11 orders before there are no more unique interacting genes and the program prints out "no more interacting genes"


Find all n-order interacting genes (if order is not specified 1st order is implied)
Enter a gene's entrez id number 2 Order 12 Find Genes
The genes interacting with gene 2 are:
1st Order: 251, 2391, 2658, 3341, 182, 255, 2923
2nd Order: 72806, 3206, 4612, 5268, 7616, 9076, 9755, 699, 4280, 4799, 4815, 4816, 4832, 4834
3rd Order: 213358, 4393, 4496, 4668, 5132, 5136, 5851, 6940, 7016, 8247, 12241, 2807, 3532, 3730, 4227, 4347, 6721, 11788, 13912, 14873, 9572, 14691, 6526, 9760, 10359, 11100, 14363, 17815, 700, 9290, 5133, 4805, 4807
4th Order: 543403, 3574, 3822, 3825, 3840, 3841, 3842, 3844, 3847, 3849, 3970, 3986, 4751, 4922, 4966, 4989, 4995, 4996, 4998, 4999, 5000, 5003, 5004, 5007, 5010, 5061, 5062, 5063, 5064, 5313, 5524, 5573, 5623, 5696, 6104, 6149, 6260, 6313, 6385, 6599, 6652, 6756, 6812, 6865, 7543, 7741, 8083, 8504, 9861, 10218, 10731, 10736, 10906, 10941, 11086, 12294, 12534, 13834, 14079, 16102, 4060, 4568, 7013, 7173, 7326, 4925, 5056, 5822, 6481, 6958, 9291, 10885, 5134, 5135, 5139, 5536, 5679, 5997, 6043, 6225, 6284, 6287, 6504, 6722, 6929, 7310, 7340, 7482, 7678, 7831, 7893, 8021, 8044, 8169, 8212, 8327, 8673, 8702, 8739, 9288, 9794, 9917, 10566, 10571, 10633, 11234, 12036, 12110, 12928, 14922, 16568, 16875, 17661, 6735, 7045, 5454, 6115, 6315, 6341, 6496, 6963, 7048, 7495, 8228, 9187, 10028, 10121, 13481, 14081, 7788, 8222, 8402, 8416, 9515, 9907, 10645, 10727, 10873, 11039, 11722, 13963, 14437, 8790, 8878, 10936, 14621, 13789, 3731, 3736, 3738, 3739, 4035, 4194, 4211, 5372, 5809, 6285, 7868, 11922, 11273, 6828, 7147, 8267, 11943, 12374, 13011, 13646, 14927, 15357, 16671, 18094, 8238, 10467
5th Order: 2279408, 12807, 5535, 6064, 6359, 7313, 7383, 8091, 8610, 8822, 8955, 9274, 9767, 9886, 8107, 8152, 13143, 5811, 7645, 12189, 11469, 3824, 3826, 3827, 3828, 3831, 3990, 4041, 4633, 4897, 5608, 6451, 6719, 6882, 7201, 7293, 7430, 7507, 7569, 7671, 7902, 8156, 9797, 10336, 12005, 12610, 12758, 13649, 13830, 16975, 4631, 5543, 11000, 11077, 3823, 3829, 3830, 3832, 3833, 3834, 3835, 3836, 3838, 3839, 3846, 3848, 3850, 3851, 3852, 3855, 3856, 3857, 3858, 3859, 6174, 6593, 6869, 6885, 7871, 8075, 8145, 8417, 9812, 11376, 12863, 14650, 15544, 3843, 3845, 8462, 9816, 3854, 4248, 4662, 5993, 7938, 9539, 12970, 5317, 6148, 3971, 7036, 7279, 9386, 5884, 7557, 8231, 13975, 3853, 5070, 5342, 6378, 7190, 9836, 9894, 4735, 5006, 6883, 7215, 9068, 9974, 11303, 4371, 5902, 8812, 13735, 4928, 7179, 7180, 7181, 7278, 7889, 15345, 4896, 6909, 7884, 8573, 9380, 13021, 14889, 4923, 4975, 5033, 5180, 5210, 5722, 5739, 5818, 5850, 5883, 5932, 5967, 5995, 6398, 6702, 6707, 7318, 7563, 8008, 8302, 8655, 8698, 8740, 9011, 9783, 10170, 11014, 11256, 11605, 11929, 13066, 13254, 13451, 14254, 14361, 14741, 15135, 15766, 16034, 4924, 5742, 5759, 5761, 5770, 5777, 5780, 7413, 11875, 17078, 5212, 6376, 6419, 6598, 6646, 7539, 7565, 8180, 8685, 8943, 9072, 9624, 9720, 9744, 9809, 9895, 10307, 10400, 10617, 10892, 10930, 11066, 11120, 11267, 11628, 12088, 12107, 12443, 12539, 13140, 13156, 13893, 15968, 15989, 18916, 6441, 6694, 6768, 6943, 7552, 8404, 8582, 10273, 10548, 10686, 12513, 12869, 13332, 13638, 13759, 5279, 5937, 6368, 7926, 9629, 13461, 6523, 7054, 7841, 8004, 8526, 17072, 6601, 7551, 8581, 11616, 11625, 11800, 13179, 5005, 6393, 10219, 14174, 16357, 16786, 16921, 17086, 11708, 13315, 17140, 18290, 5606, 6075, 6751, 6880, 7034, 8499, 9221, 9919, 11892, 15719, 16418, 9313, 5280, 8950, 6085, 5853, 6111, 6306, 9888, 12478, 19125, 5586, 5871, 7439, 7489, 7603, 7622, 7927, 7970, 8274, 8683, 8771, 8792, 8811, 8815, 8817, 8823, 9488, 9634, 9846, 10118, 10157, 10162, 11075, 12237, 12481, 12694, 12952, 13737, 13855, 14431, 14672, 14937, 15915, 15916, 5948, 6095, 8027, 8532, 8764, 8782, 8820, 9551, 9896, 10661, 13544, 14953, 15073, 5996, 6566, 6977, 7983, 9219, 9297, 9637, 9928, 12464, 12588, 12925, 13216, 13627, 13842, 14489, 17148, 6552, 5808, 7286, 7847, 10762, 12017, 9501, 6460, 7818, 9388, 9435, 11025, 19936, 11443, 9542, 7412, 10048, 11557, 8997, 16939, 8530, 11360, 12228, 16702, 14519, 18538, 12384, 15447, 5565, 5676, 9195, 9922, 9987, 10260, 11703, 8937, 10247, 11999, 7719, 9337, 10564, 11386, 4994, 5293, 5859, 5933, 6109, 6745, 7076, 7327, 7393, 8457, 8969, 9632, 9665, 10025, 10300, 11475, 12596, 14073, 14585, 16459, 16953, 5002, 5920, 11474, 11909, 12364, 15433, 6344, 6373, 6374, 6718, 7109, 7133, 8123, 9284, 12639, 13337, 14517, 5829, 11258, 5553, 5733, 6025, 7080, 8154, 8184, 8436, 8879, 8941, 9950, 11624, 11752, 12702, 6513, 5329, 10165, 12786, 5560, 5657, 5909, 5989, 6290, 6959, 7017, 7294, 7436, 7653, 7661, 8942, 13146, 13691, 14649, 19765, 13058, 6215, 6925, 11453, 12581, 14349, 15277, 6746, 8615, 11548, 11786, 7754, 7978, 13614, 13683, 8441, 11353, 14743, 11841, 13320, 13681, 13688, 8357, 8358, 12128, 16090, 16092, 5462, 5521, 5533, 5576, 5928, 5929, 6234, 6335, 6402, 7438, 7457, 7465, 7577, 7758, 8019, 8106, 8149, 8286, 8292, 8544, 8971, 9289, 10252, 10342, 10608, 10647, 10984, 12529, 12633, 12712, 13031, 15471, 5790, 6080, 6433, 6486, 6600, 7118, 7880, 8748, 9641, 10239, 11103, 13094, 13393, 6489, 8391, 13045, 7994, 10607, 10786, 12427, 8722, 10144, 10265, 11785, 11802, 11819, 14965, 15330
6th Order: 7838715, 12120, 12872, 9035, 9112, 11748, 3737, 6065, 7170, 7679, 7680, 8386, 5831, 6825, 6938, 7490, 7681, 7172, 16191, 6262, 6415, 11011, 4036, 4044, 4159, 4206, 4537, 4618, 4718, 4973, 5359, 5367, 5523, 5673, 6154, 6554, 7136, 7379, 7911, 8290, 8315, 8555, 9366, 9796, 10086, 12798, 15370, 4040, 4781, 6108, 16435, 19228, 4907, 6268, 8838, 14065, 5895, 7933, 8443, 10609, 10861, 11525, 12116, 12413, 15758, 16561, 17980, 13170, 18313, 4502, 8151, 16615, 9385, 4473, 4776, 7442, 8064, 4680, 4707, 4899, 5412, 6604, 6785, 7082, 7491, 7513, 7896, 8214, 8242, 9557, 10057, 10589, 10734, 5949, 6412, 7897, 11293, 11718, 17909, 5384, 6800, 8114, 14252, 17289, 14359, 14317, 9831, 13899, 7700, 8110, 8609, 10910, 10927, 12376, 12542, 12842, 12586, 9843, 9844, 12051, 12174, 16946, 12377, 12216, 12223, 16038, 5651, 13547, 9122, 9670, 9090, 4571, 7119, 8370, 4963, 6678, 10626, 5402, 7007, 10334, 11535, 13768, 13990, 11401, 8199, 10531, 10652, 10411, 10664, 4087, 6815, 10860, 11609, 12795, 13244, 14446, 15021, 17398, 4006, 4694, 4696, 5415, 5501, 5532, 8179, 8413, 8756, 10983, 11367, 14513, 7949, 11046, 9245, 9537, 11544, 11726, 13268, 15751, 16775, 7608, 11414, 12042, 9345, 10174, 4745, 4841, 5045, 5289, 5605, 7185, 7365, 10061, 12109, 6443, 6611, 6618, 6726, 7659, 8492, 8821, 8890, 9185, 9335, 10107, 10425, 10519, 11114, 12226, 12777, 12781, 13281, 13877, 14482, 14572, 14893, 15424, 16016, 16683, 18980, 13753, 8545, 8209, 12092, 13416, 7932, 8972, 9958, 11224, 11352, 12669, 13105, 14259, 8230, 17217, 11129, 11570, 12920, 13420, 13981, 5645, 6167, 6644, 8747, 10879, 13107, 14689, 5397, 5424, 6019, 6480, 9989, 6841, 7152, 8166, 8963, 6096, 16540, 5740, 5762, 10176, 14810, 6816, 7251, 7533, 6066, 16289, 12074, 12435, 15521, 13555, 13084, 18892, 12972, 14051, 6501, 12159, 12587, 14275, 9594, 9873, 12544, 11244, 11416, 14562, 6003, 14301, 5782, 8374, 15486, 9978, 15445, 15446, 8259, 19483, 10173, 9945, 10020, 13039, 13046, 17692, 10308, 17534, 10934, 12536, 11425, 11852, 8334, 12653, 7797, 14920, 6864, 8074, 9229, 13936, 7586, 14368, 11651, 14115, 12059, 12327, 13507, 7098, 7274, 8426, 10597, 13568, 16803, 12911, 9135, 8002, 11458, 6660, 8678, 15538, 6752, 9883, 16417, 11727, 14479, 15756, 7497, 7877, 10206, 14635, 6477, 10041, 10918, 12178, 13027, 7032, 15705
7th Order: 113410089, 10938, 10978, 11897, 14723, 16518, 17008, 12397, 11959, 15055, 16322, 11223, 6450, 9320, 14582, 10603, 13223, 11553, 9130, 9404, 11254, 11292, 11375, 15558, 13383, 14596, 6634, 11019, 11910, 15833, 12218, 6092, 8625, 13305, 14282, 13806, 10994, 14515, 10397, 14380, 10605, 5196, 5398, 7765, 8155, 8175, 8653, 9920, 10440, 6150, 9278, 13206, 14682, 5334, 5730, 5814, 7022, 8052, 8475, 10278, 10387, 10638, 11119, 11526, 11723, 12780, 12800, 13131, 13777, 13966, 14094, 5890, 6164, 8261, 18807, 6912, 14164, 10655, 6961, 7531, 10486, 12225, 8774, 9048, 11243, 7725, 7701, 11349, 5114, 5115, 5209, 5577, 5638, 5653, 6423, 6837, 6979, 7779, 8621, 9420, 13734, 14424, 15171, 16599, 17457, 5087, 5625, 6004, 9929, 6005, 6759, 7064, 7630, 8301, 18035, 8986, 7487, 7761, 10005, 15515, 6334, 7095, 7226, 8572, 8994, 9369, 9946, 10526, 14615, 9223, 9916, 10954, 11430, 8068, 9516, 11146, 11720, 15779, 8503, 11887, 10193, 10684, 13915, 16964, 11919, 9713, 17089, 9321, 10010, 18588, 12528, 7103, 11445, 5542, 6994, 10275, 14114, 7216, 7476, 8935, 10011, 11721, 12146, 12540, 12613, 20128, 13353, 13594, 16603
8th Order: 13037619, 7649, 10789, 5799, 7614, 7948, 8553, 8720, 9480, 10383, 10512, 10546, 11071, 12494, 13335, 14132, 16313, 17647, 9973, 5941, 7606, 10027, 10111, 10112, 7127, 12442, 10270, 10209, 11861, 13480, 7669, 8048, 10946, 14935, 7809, 8013, 8232, 17320, 13787, 9401, 10409, 12796, 9317, 15648, 10388, 12630, 10394, 12949, 10624, 11220, 12982, 16504, 5903, 9279, 12643, 15216, 6579, 10052, 8229, 9523, 9927, 10316, 10562, 10875, 12996, 13533, 14134, 16728, 10831, 10923, 13329, 10739, 11699, 16623, 16944, 16474, 16475, 7171, 10299, 7265, 12828, 14605, 7337, 4160, 4161, 4163, 4164, 4170, 4171, 4172, 4173, 4174, 4175, 4177, 4178, 4179, 4182, 4183, 4184, 4186, 4187, 4188, 4189, 4190, 4193, 5951, 4176, 4180, 4181, 4185, 4192, 6171, 7140, 4539, 4577, 4687, 4699, 4786, 4826, 5131, 6145, 10988, 11702, 12264, 4576, 4784, 12889, 4651, 4698, 5670, 9581, 10698, 12113, 18357, 6152, 6911, 6960, 7626, 8849, 9429, 11009, 12632, 5185, 5927, 5950, 7192, 9328, 13357, 15194, 4725, 5725, 6242, 6086, 5411, 11594, 5522, 6435, 6803, 7126, 7923, 8376, 12625, 7068, 6363, 6766, 8411, 12648, 13177, 15199, 15925, 16469, 7423, 7432, 7449, 8693, 8735, 10230, 10705, 11133, 11711, 12203, 13004, 19217, 17250, 11868, 9672
9th Order: 14894471, 4474, 4475, 4476, 4477, 4478, 4479, 4480, 4610, 6123, 5129, 5299, 7011, 9347, 6114, 13444, 8652, 9721, 10791, 12143, 12654, 13368, 13449, 14072, 15010, 16923, 4507, 7050, 8681, 16902, 5375, 6921, 7682, 4893, 5113, 6799, 6884, 8593, 8826, 10137, 12252, 12130, 5353, 6535, 8258, 9574, 9982, 6835, 5588, 5972, 8284, 8300, 9943, 11728, 6786, 6787, 6797, 11486, 13220, 18023, 9470, 12774, 12445, 11713, 10332, 11565, 14033, 15032, 11603, 12978
10th Order: 155911083, 11684, 12248, 14289, 16719, 14853
11th Order: 1565
no more interacting genes



Mongo DB

Mongo has mongoimport utility that allows to upload files of various formats, including .csv. The documents have PATIENT_ID, DIAGNOSIS, and all entrez_ids as their field. This allows to avoid costly updates that would be caused by smaller documents (e.g., every gene value as a document). After loading the expression values of the genes with this utility, I used pymongo, a Mongo API for python. It is well-documented and has basic statistics functions that allow for quick and easy data analysis. The python script takes entrez_id as an input, aggregates documents by disease, and calculates average and standard deviation values for specified gene. In this way, we can see expression differences for all seven disease groups with one command. An example follows:

```
Yaroslavs-MacBook-Pro:mongo yaroslavmarkov$ python gene_stats.py
Enter entrez id of the gene: 143
Diagnosis: 1, average: 2.11672222222, std: 0.770247312777
Diagnosis: 2, average: 2.22281690141, std: 0.831239839457
Diagnosis: 3, average: 1.833, std: 0.584928200722
Diagnosis: 4, average: 2.2805, std: 0.773127253432
Diagnosis: 5, average: 2.1047826087, std: 0.632262119317
Diagnosis: 6, average: 2.289, std: 0.655918440052
Diagnosis: NA, average: 1.88, std: 0.46
```

Cassandra

Cassandra uses CQL which is bundled with itself. It allows to create and alter keyspaces, tables and upload .csv files straight to existing tables. Multiple documents may be uploaded to the same table using a primary key. However, in this case the documents were merged prior to uploading everything to the database to avoid issues with primary key, as UNIPROT is not indicated in all datasets. To upload a JSON or other similar formats a python script could be used similar to the one used in the first version of Mongo script. After the table is populated, the data can be retrieved with the following query:

```
SELECT ‘column’ from genes_info.proteins WHERE ‘conditions’ ALLOW FILTERING;
```

Which retrieves specified by ‘columns’ attributes of a gene similar to SQL.

Example:

```
cqlsh> SELECT gene_name from genes_info.proteins where entrez = 90639 ALLOW FILTERING;

 gene_name
-----------------------------------------------------
  COX19  cytochnome c oxidase assembly facton(COX19)
  COX19  cytochnome c oxidase assembly facton(COX19)

(2 rows)
```

In this case we retrieve gene_name attribute from every row where entrez id is 90639.

Drawbacks

Obviously one of the biggest downsides is using three different DBMS as that takes up more space than just one, it is also harder to set up as it requires all the technologies to be installed.

SQL is susceptible to cross site scripting and sql injection if the input is not sanitized. With Mongo and Cassandra because there is no set schema, it is harder for attackers to succeed with injection attempts. Also, for this Mongo implementation, retrieval of documents requires transferring large amounts of data as each documents stores expression data of all genes. This could be solved by creating a document for each gene-patient pair, but in this case updating disease data would require updating thousands of documents.