Identifying Cancer Using Genome Deep Learning

Nima Azbijari
University of Hawai'i at Mānoa
nazbijari@hawaii.edu

Kaitlyn Jacobs
University of Hawai'i at Mānoa
kjacobs8@hawaii.edu

Katrina Turner
University of Hawai'i at Mānoa
khj@hawaii.edu

Billy Troy Wooton
University of Hawai'i at Mānoa
bwooton@hawaii.edu

I. Introduction

Many domains, including healthcare, can benefit from well-designed applications of artificial intelligence. One example is explored in the paper, "Identification of 12 Cancer Types through Genome Deep Learning." (Sun, et al., 2019) Being able to identify cancer with a high amount of certainty through genome deep learning will enable doctors and scientists to catch cancer cases earlier than physicians are able to currently. It will also allow scientists to broaden our understanding of how the genome plays a role in cancer development and can lead to more specialized research on treatments. It can be a vital tool in the development of more targeted gene therapy - an increasingly studied treatment method (Dunbar, et al., 2018). Cancer is still the second leading cause of death in the United States (CDC Cancer Data and Statistics, n.d.) and applications to utilize machine learning will expedite the aforementioned research. While the paper by Sun et. al addresses an interesting problem, the framework proposed by the authors is ill-defined. We aim to address these weaknesses and produce a well-defined model that is useful in the field.

Genomic cancer research has increased rapidly in the last few decades, leading to the widely known relationship between mutations in the genome and cancer development. One big leap forward was the discovery of the involvement of p53, a proto oncogene that is involved in the majority of cancers, with mutations of the gene occurring in 40-45% of cancer patients (Soussi, 1994). Germline mutations in the tumor suppressor genes brca1 and brca2 were discovered as well to have a high correlation with breast cancer, with 30-50% of hereditary patients showing these mutations (Ferla, et al., 2007). These studies are just a glimpse into the research into cancer genomics, and we have found out a lot about specific genes that are involved heavily in cancer. However, much more research is needed on how these genes interact with one another, as well as somatic mutations that are frequently seen across a variety of cancers. Recent research has demonstrated that machine learning, and particularly deep learning can produce promising results in terms of leveraging individual sequence variations (such as those manifested in somatic mutations) to predict molecular traits (Angermueller, Parnamaa, Parts, & Stegle, 2016). The database COSMIC is a curation of these somatic mutations, compiling mutations for 83 different cancer related genes (Forbes, et al., 2011). Together with this data, as well as the ease and frequency of human genome sequencing, a deep learning model can be trained to pick up these mutations quickly and with increasing accuracy, giving doctors and researchers another tool to battle the increasing cases of cancer seen around the world.

II. Data

In an effort to mirror the paper by Sun et. al, similar data sets were collected. Somatic mutation data was collected from two main sources, the cancer data from the International Cancer Genome Consortium (Zhang, et al., 2011) and the healthy data from the 1,000 Genomes project. (Auton, Abecasis, Altshuler, & et al., 2015) The cancer data included 5,987 donors with 12 different cancers and the healthy data included 2,504 donors. The 12 cancer types are urothelial bladder carcinoma (BLCA), breast adenocarcinoma (BRCA), colon adenocarcinoma (COAD), glioblastoma multiforme (GBM), kidney renal clear cell carcinoma (KIRC), low grade glioma (LGG), lung squamous cell carcinoma (LUSC), ovarian carcinoma (OV), prostate adenocarcinoma (PRAD), skin cutaneous melanoma (SKCM), thyroid carcinoma (THCA) and uterine corpus endometrial carcinoma (UCEC). Table 1 shows the distribution of data.

Table 1. Distribution of Data by Donor Type

Donor Type	***Count***
Healthy	2504
Urothelial Bladder Carcinoma (BLCA)	411
Breast Adenocarcinoma (BRCA)	1020
Colon Adenocarcinoma (COAD)	402
Glioblastoma Multiforme (GBM)	388
Kidney Renal Clear Cell Carcinoma (KIRC)	361
Low Grade Glioma (LGG)	508
Lung Squamous Cell Carcinoma (LUSC)	485
Ovarian Carcinoma (OV)	426
Prostate Adenocarcinoma (PRAD)	497
Skin Cutaneous Melanoma (SKCM)	466
Thyroid Carcinoma (THCA)	492
Uterine Corpus Endometrial Carcinoma (UCEC)	531

Two encodings were created from the data sets. The first was a simple binary encoding for mutations, 1 if the given donor had that mutation and 0 if they did not. This created a very large, sparse matrix with over a million parameters. Filtering out mutations shared by less than 5 donors brought it down to about 33,000 parameters, but still a very sparse matrix. This approach was similar to the one in the Genome Deep Learning paper (Sun,

et al., 2019), but since we chose a multi-class model, we kept more parameters.

The second encoding contained frequencies and was based on the genes affected rather than the mutations. In this encoding, the number of mutations affecting each gene were counted for each donor and put in the matrix. Once again, this resulted in a fairly sparse matrix, but not nearly as sparse as the first. After filtering out genes only affecting a few donors, the number of parameters remained about 30,000. Ultimately, we opted to use the second encoding to feed into our models since upon preliminary evaluation, they gave better results.

The 1000 genomes data was gathered from the phase 3 project download site (1000 Genomes, n.d.) as a single VCF file containing the integrated structural variation map for all 2504 individuals included in the study. Mutation data was parsed from the VCF file using the scikit-allel package (Scikit-allel, 2019), which includes numerous functions to enable the extraction of data from VCF files. The PyEnsembl package (PyEnsembl, n.d.) was then used to map the position and chromosome of each structural variant call to an ensembl gene ID, thus converting the data from its raw form into the same gene frequency encoding used for the cancerous individuals.

III. Visualizations and Exploratory Analytics

In order to better understand our data, we produced a series of visualizations, both to understand the biological functions as well as the shape of the data. We also ran some preliminary analytics to help predict what our results might look like.

A. Pathway Enrichments

First, we wanted to see which functional pathway was being enriched for each of the cancer types. If there is a certain functional group being enriched for only one cancer, researchers can provide specialized treatments and possibly catch on to cancer earlier, when paired with the model. We used Cytoscape version 3.8.0, with the application BiNGO (Maere, Heymans, & Kuiper, 2005) that analyzes over-represented gene ontologies for a given gene symbol list. We used the list of genes affected from our data, separated by cancer type, to create a visualization of the affected pathways for each cancer. Low grade gliomas was the data set with the least amount of genes affected, and thus had the least amount of enriched pathways, making it the easiest to visualize. (Figure 1)

Pathway enrichment charts for the other eleven cancers are located in Appendix A. This is a valuable way to look at how the functional ontologies of these affected genes are all connected, and how related they all are. However, due to the extremely high number of represented pathways (440 at the most), it can be difficult to interpret. Not only because of the size, but because enriched pathways at the top of the hierarchy (more specific) could be affecting pathways at the bottom, leading to a misinterpretation of broader ontologies.

In order to make the functional processes of these genes easier to visualize, we plotted the top ten enriched pathways (from the Panther classification data) for each cancer in a stacked bar chart below. (Figure 2) Using this layout, we can more easily see which pathways are common among many

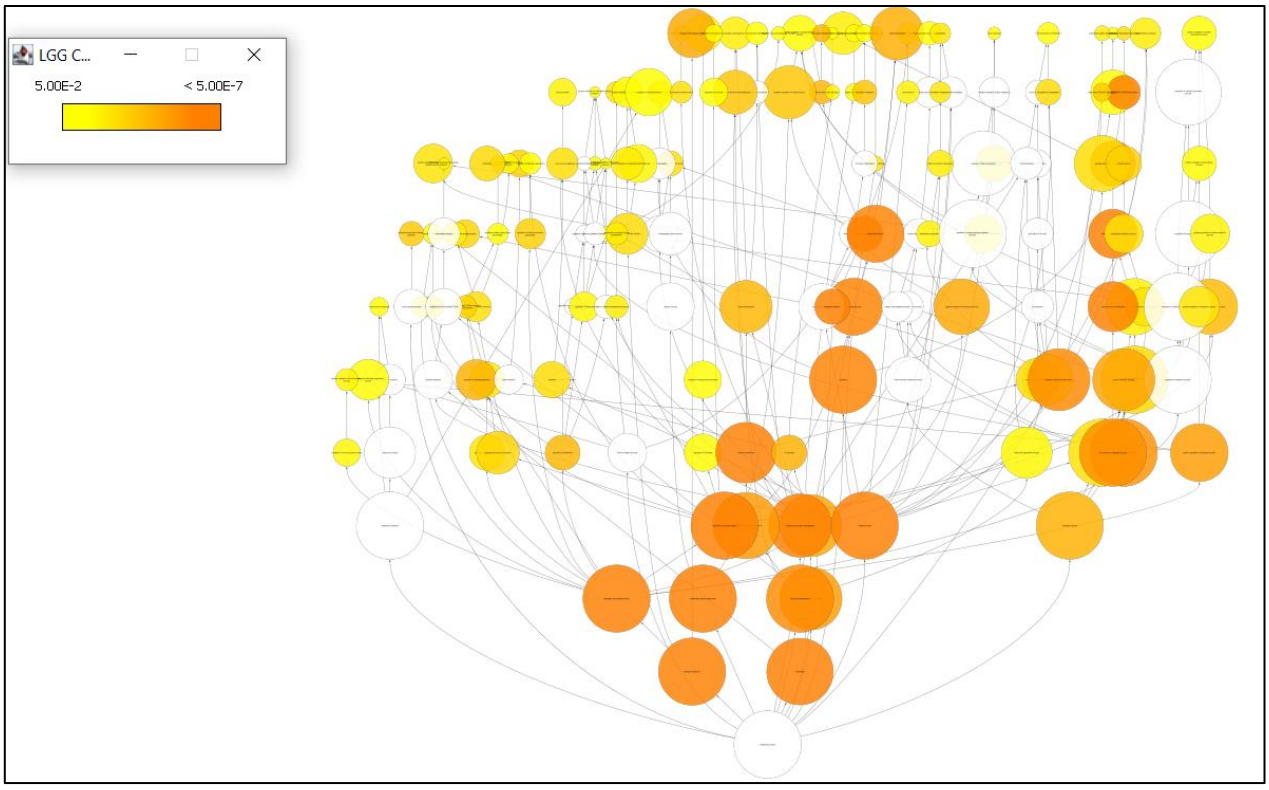

Figure 1. A pathway enrichment chart for low grade gliomas, representing 177 gene ontologies. Color on this chart represents the p-value of the Fisher's exact test for over representation, and the size represents how many genes are included in the pathway.

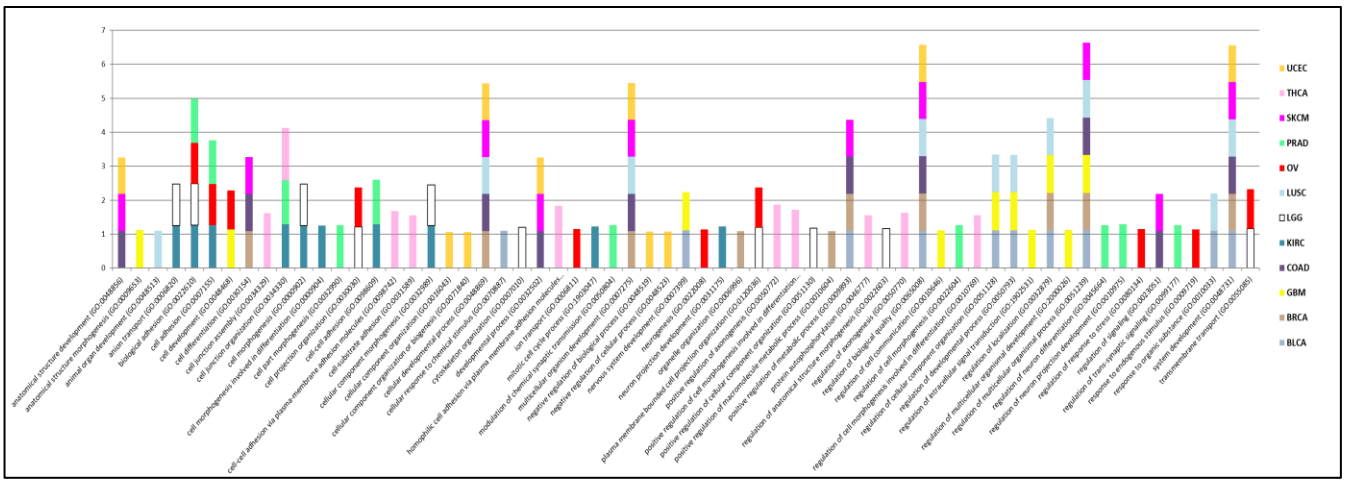

Figure 2. This bar plot shows each of the twelve cancer's top ten gene ontologies. The y axis represents enrichment and the x axis lists the pathways. Each cancer has anywhere from 1 to 3 fold enrichment.

cancers: regulation of biological quality (GO:0065008), regulation of multicellular organismal processes (GO:0051239), and system development (GO:0048731); and which cancers have pathways that are unique to them: thyroid carcinoma (THCA) and prostate adenocarcinoma (PRAD). Part of this difference is that THCA and PRAD were two of our biggest datasets, so those cancers are going to have much more data to analyze, and therefore can be grouped into more specific ontologies. These cancers also target a specific part of the body, the thyroid and prostate, respectively. This may be the cause of a more specialized functional pathway being affected than cancers in our model like skin cutaneous melanoma (SKCM) which is widely distributed across the body as well as functional pathways.

B. UMAP

Since the data set is sparse and very highly dimensional, it can be difficult to visualize. First we ran UMAP (McInnes, Healy, & Melville, 2018) on the full data set to reduce it to 2 dimensions. The resulting graph (Figure 3; left-side) shows the clear distinction between the healthy and cancerous data sets. This would indicate that a binary classifier trained to distinguish between cancer and healthy individuals should perform very well on this dataset, as a clear decision boundary is present in 2 dimensions. However, there is still a lot of overlap between the cancers. In order to get a better visualization of the relationships within the cancer data in high-dimensional space, we then re-ran UMAP on just the cancer data, again reducing it from 30,000 to 2 dimensions. (Figure 3; right-side) This plot shows that while

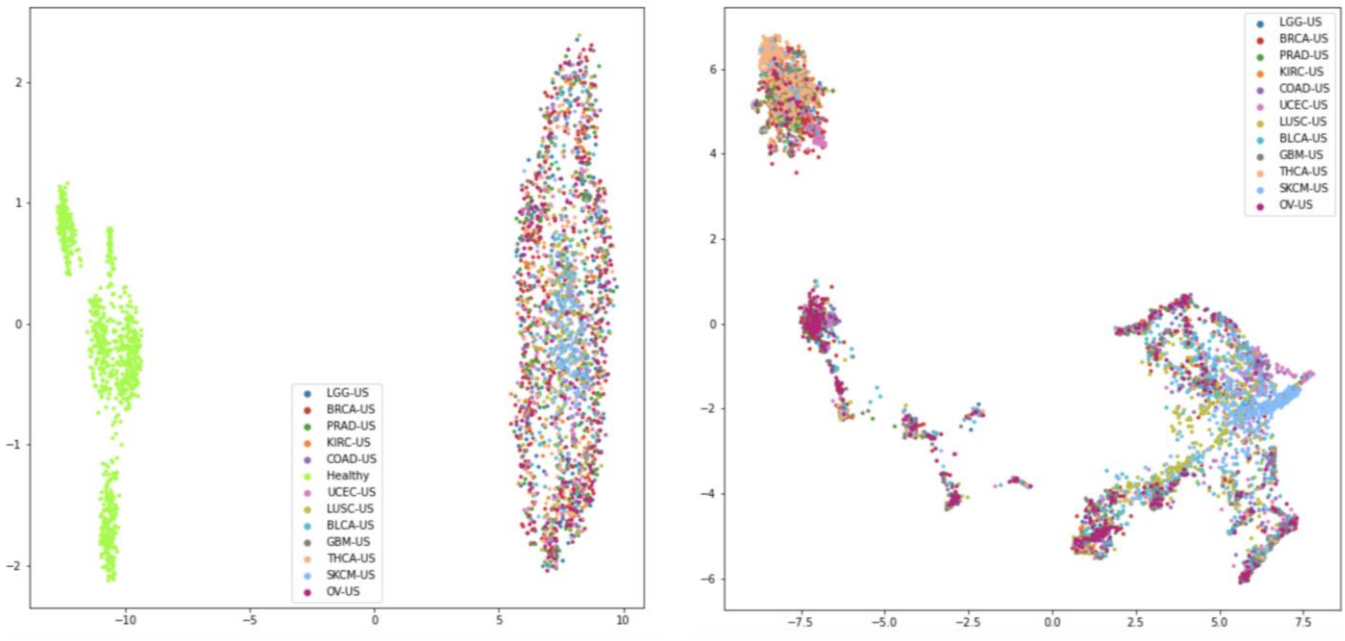

Figure 3. Visualization of 2-dimensional embeddings generated by UMAP models fit to the entire dataset (left) and only the cancer data (right).

the UMAP model is able to elucidate separation between some of the classes (e.g. THCA, top-left and SKCM, bottom-right), a large amount of overlap between the various classes still exists. This would indicate that learning a decision boundary that can distinguish between all 12 classes will require that the model learn a high-dimensional hyperplane that can divide the classes, and that this decision boundary is difficult to capture in two dimensions.

IV. Methods

Knowing the cancer data has high overlap, we took two approaches to finding a good classifier for the data. The first splits the process into two steps, first identifying if a patient has cancer, or not, and then attempting to classify the cancer. The second attempts to use an autoencoder and classifier together to predict healthy or cancer type all together.

To parse the ensembl gene ids to gene symbols, we wrote a parser in Jupyter notebook using MyGene located in Appendix B. MyGene is a database used to retrieve gene information and annotation data. After that, the gene symbols were used as an input on the gene ontology website, which uses the Panther classification database (Panther Classification System, 2020) to get gene ontology and functional information from the gene list.

A. *Two-Model Approach*

One approach to training a DNN model to perform classification of healthy vs. the twelve cancer types is to use a two model approach. In this approach, two fully connected neural network models are trained in a supervised fashion on the same input data, but with different optimization objectives. The first model is trained to perform binary classification to distinguish healthy examples from cancerous examples. To train this model, we label the data as follows. Let D = X x Y denote the set of labelled data. Let X denote the set of inputs, where each x $\in$ X is a vector of approximately 33,000 integer values, encoding the frequency of mutations affecting each of the approximately 33,000 genes reported in the dataset. Each x_i represents the mutation data for a separate individual donor. Let Y denote the set of labels, where $y_i = 1$ if x_i represents an individual with cancer (any one of BRCA, BLCA, THCA, etc.), and $y_i = 0$ if x_i represents an individual who is healthy. We then train the following binary classifier to predict a label $\hat{y}_i$ for each x_i in the training data, and we minimize the standard binary cross-entropy between the predicted label $\hat{y}_i$ and actual label y_i via backpropagation during training. A schematic diagram of the architecture binary classifier is depicted in Figure 4.

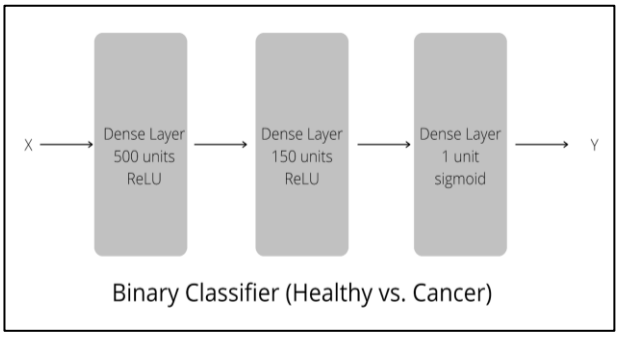

Figure 4. The model architecture of the binary classification model, consisting of three dense layers.

The second of the two models is trained to perform multiclass classification on individuals with cancer, further labelling them with a predicted type of cancer, such as BLCA or BRCA. To train this model, we use the same input encodings X as those used to train the binary classifier, but only train the model on cancerous examples. We produce multi-class labels Y, where each y_i is a one-hot vector of length 12, indicating which type of cancer individual x_i has. A schematic diagram of the architecture of the multi-class cancer type classifier is depicted in Figure 5.

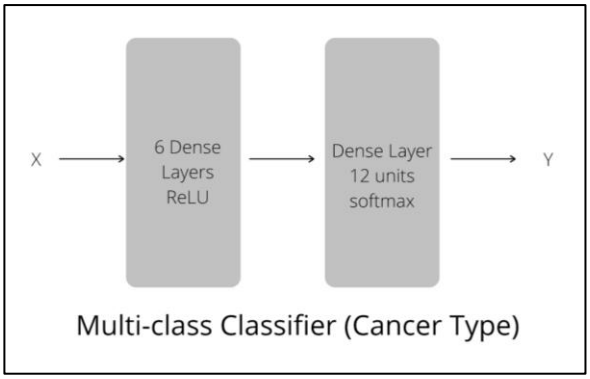

Figure 5. Simplified schematic of the model architecture of the cancer type classifier, consisting of 7 dense layers.

B. *Multi-Output Model*

The second approach attempted is the training of an autoencoder and classifier together. We allow the model to have a single input layer, but a middle layer branches out to a classifier module and a reconstruction module. Essentially, an autoencoder is constructed but the encoder output is linked to the classifier input and the decoder input so that the classifier and the autoencoder are trained together. This allows the two losses to be optimized in parallel and proved to provide better results in comparison to training the autoencoder first and transferring the encoder to a classifier to be trained separately. The test set is pulled from the shuffled full data (500 test samples). A basic diagram of this model is depicted in Figure 6.

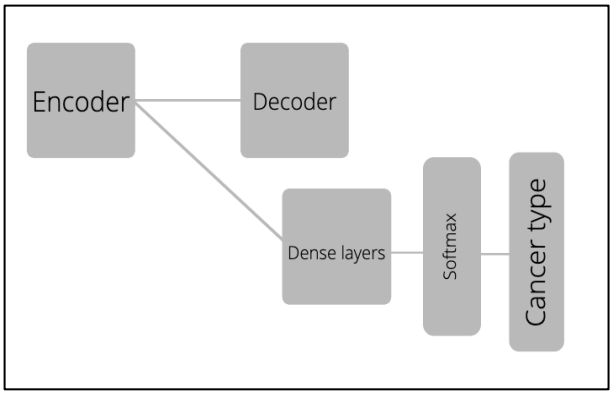

Figure 6. Simplified schematic of the model architecture of the multi-output model.

V. Discussion of Results

A. *Two-Model Approach*

The initial binary classifier was trained on 4200 samples for 4 epochs, achieving a final validation accuracy of 100% on 2800 validation samples in the 4th epoch. On a hold-out set of 1491 samples, this classifier obtained a final test accuracy of 100%. A confusion matrix showing the binary classifier's predictive performance on each of the two class labels on the test dataset is depicted in Figure 7.

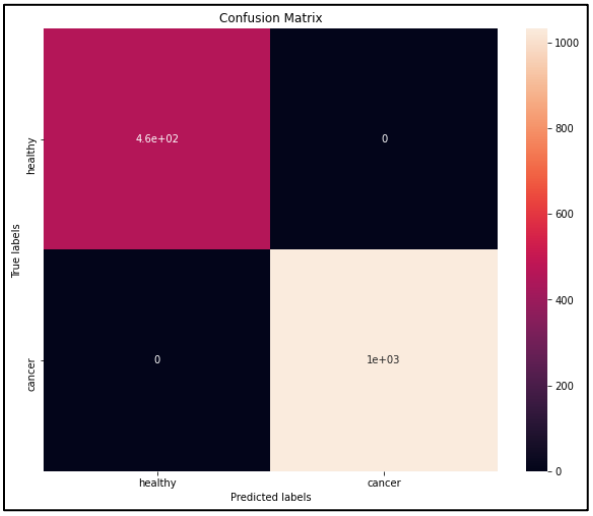

Figure 7. Confusion matrix showing the distribution of binary classifier predictions on the test set.

The cancer type (multi-class) classifier was trained exclusively on the 5,987 cancer samples, which were split into a training set of 5,000 and a test set of 987 samples. The training set was further split into training and validation sets, with a validation split of 1,250 samples out of 5,000. After training for 15 epochs, and minimizing the categorical cross-entropy between the predicted class labels and actual class labels, the multi-class classifier achieved a final validation accuracy of 50.64% in the 15th epoch. On the hold-out set of 987 samples, the classifier achieved a test accuracy of 50%. A confusion matrix depicting the multi-class classifier's predictive performance on the hold-out set is depicted in Figure 8. While the performance of the multi-class classifier alone leaves much to be desired, in order to evaluate the merits of the two-model approach, it is helpful to calculate the overall accuracy of the predictions made by the two models as a whole. To do this, we take the 460 healthy examples from the binary hold-out set, and combine them with the multi-class hold-out set of 987 samples to form a single test set of 1,447 samples. We then add up the number of correct predictions in the two confusion matrices depicted in figures 7 and 8, yielding a total of 982 correctly predicted samples. This amounts to an overall test accuracy of 67.86%, which is significantly worse when compared to the single-model approach described in section VI.*B*. Confusion matrices showing the model predictions on the full-dataset are available in Appendix C.

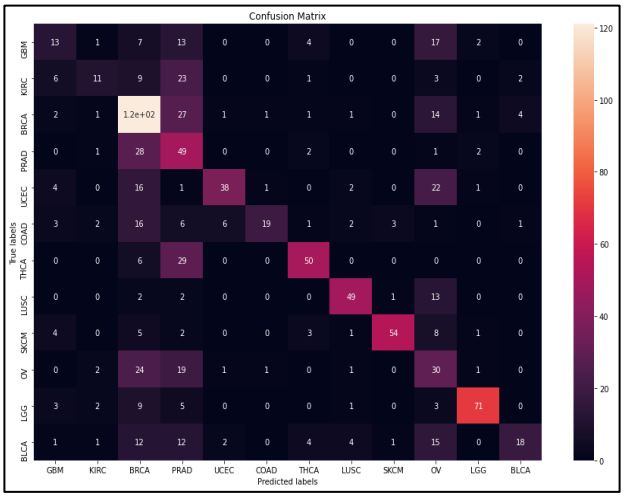

Figure 8. Confusion matrix showing the distribution of class label predictions generated by the cancer type classifier on the test set.

Table II. Scores for Two-Model Approach

Class	*Precision*	*Recall*	*F1 Score*
Healthy	1.0	1.0	1.0
GBM	0.36	0.23	0.28
KIRC	0.52	0.2	0.28
BRCA	0.47	0.69	0.56
PRAD	0.26	0.59	0.36
UCEC	0.79	0.45	0.57
COAD	0.86	0.32	0.46
THCA	0.75	0.58	0.65
LUSC	0.8	0.73	0.76
SKCM	0.91	0.69	0.78
OV	0.23	0.38	0.28
LGG	0.89	0.75	0.81
BLCA	0.72	0.26	0.39

While the two-model approach did not perform as well as the autoencoder/multi-class classifier hybrid approach, there is a notable correspondence between classes which the two-model classifier performed well on (measured in terms of F1 score) and the classes shown to be more separated from others in the UMAP visualizations. Clearly, the two-model approach is excellent at distinguishing healthy individuals from cancerous individuals, as was expected based on the UMAP visualization in Figure 3(left), which showed a clear separation between healthy and cancer data points. However, the model also performs relatively well in classifying LUSC and SKCM samples, with F1 scores of 0.76 and 0.78 respectively. These samples, represented by the olive and light-blue colored points in the UMAP visualization in Figure 3(right), were shown to be considerably better separated from the other classes. The fact that these samples are more separated from the other classes in

high-dimensional space explains why the model performs better in classifying LUSC and SKCM inputs.

B. Multi-Output Model

The second model was trained for 100 epochs with batch sizes of 32. RMSprop was used as the optimizer with an initial learning rate of 0.001, a discount factor of 0.8, and momentum of 0.3. A schedule was used to decrease the learning rate. The scheduler monitors the validation loss by waiting for an improvement for five epochs. If no improvement occurs then the learning rate is reduced by a factor of 0.2. This resulted in a test accuracy of 82.6% and validation accuracy of 79%. Due to the imbalance in the dataset we also measured the F1 score which was 0.89. The confusion matrix is shown in Figure 9 and includes both the healthy and cancer classifications.

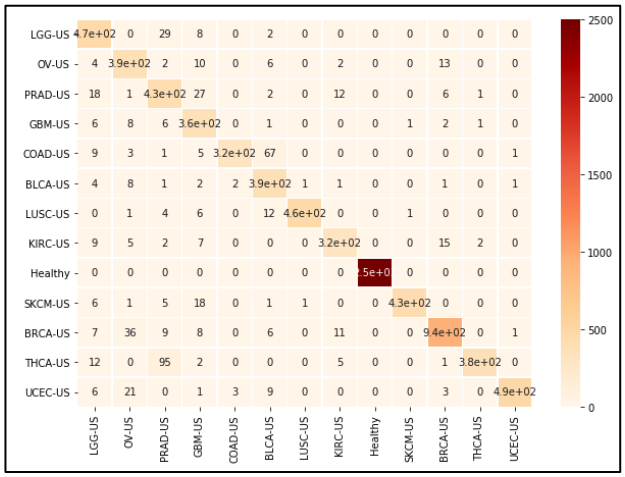

Figure 9. Confusion matrix showing the distribution of binary classifier predictions on the test set.

VI. Next Steps and Conclusion

The results we got from our two models were promising and show that deep learning can definitely be applied to help with identification of cancer at a gene level. There is still much research to be done in this area and we have included a few of our ideas here.

One idea we had to get better results when classifying which cancer a donor has, is to derive specific tissues that contain the enriched genes. By figuring out areas of local enrichment, medical professionals will be able to more precisely target predictive tests like biopsies and cell cultures. To do this, we can use the R package TissueEnrich (Jain & Tuteja, 2018) that takes a list of input genes and determines if any of them are enriched, and from those if there is any tissue-specific enrichment.

The lack of adoption of complex models in healthcare is mainly due to the lack of interpretability. Our second area of exploration would be to address this problem. By establishing a pipeline that provides information about what the model is learning and how inputs are contributing to the outputs would be useful. One approach would be to use DeepLIFT to see how each gene is contributing to the cancer predictions. This could then motivate exploration into gene pathways. Another idea would be to reduce the input dimension space and maintain interpretability. The reason we trained the model with a high input dimension was to be able to use current interpretation methods for deep learning models. A method to reduce to dimensionality and maintain human level interpretability when computing contributions from this input would be extremely useful.

References

1000 Genomes. (n.d.). Retrieved from ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/phase3/integrated_sv_map/

Angermueller, C., Parnamaa, T., Parts, L., & Stegle, O. (2016, July). Deep learning for computational biology. *Molecular Systems Biology, 12*(7). doi:10.15252/msb.20156651

Auton, A., Abecasis, G., Altshuler, D., & et al. (2015). A global reference for human genetic variation. *Nature, 526*, 68-74. doi:10.1038/nature15393

CDC Cancer Data and Statistics. (n.d.). Retrieved from CDC Centers for Disease Control and Prevention: https://www.cdc.gov/cancer/dcpc/data/index.htm

Dunbar, C., High, K., Joung, J., Kohn, D., Ozawa, K., & Sadelain, M. (2018, January). Gene therapy comes of age. *Science*. doi:10.1126/science.aan4672

Ferla, R., Calo, V., Cascio, S., Rinaldi, G., Badalamenti, G., Carreca, I., . . . Russo, A. (2007). Founder mutations in BRCA1 and BRCA2 genes. *Annals of Oncology, 18*. doi:doi:10.1093/annonc/mdm234

Forbes, S., Bindal, N., Bamford, S., Cole, C., Kok, C., Beare, D., . . . Futreal, P. (2011, January). COSMIC: mining complete cancer genomes in the Catalogue of Somatic Mutations in Cancer. *Nucleic Acids Research, 39*, D945-D950. doi:10.1093/nar/gkq929

ICGC Data Portal. (2019). Retrieved from https://dcc.icgc.org/

Jain, A., & Tuteja, G. (2018). TissueEnrich: Tissue-specific gene enrichment analysis. *Bioinformatics*. doi:10.1093/bioinformatics/bty890

Maere, S., Heymans, K., & Kuiper, M. (2005). BiNGO: a Cytoscape plugin to assess overrepresentation of Gene Ontology categories in biological networks. *Bioinformatics, 21*. doi:10.1093/bioinformatics/bti551

McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*. doi:arXiv:1802.03426

Panther Classification System. (2020). Retrieved from http://www.pantherdb.org/

PyEnsembl. (n.d.). Retrieved from GitHub - openvax: https://github.com/openvax/pyensembl

Scikit-allel. (2019, June). doi:10.5281/zenodo.3238280

Soussi, L. e. (1994). Multifactorial analysis of p53 alteration in human cancer: A review. *International Journal of Cancer, 57*.

Sun, Y., Zhu, S., Ma, K., Liu, W., Yue, Y., Hu, G., . . . Chen, W. (2019). Identification of 12 Cancer Types through Genome Deep Learning. *Scientific Reports, 9*. doi:doi:10.1101/528216

Zhang, J., Baran, J., Cros, A., Guberman, J., Haider, S., Hsu, J., . . . Kasprzyk, A. (2011). International Cancer Genome Consortium Data Portal—a one-stop shop for cancer genomics data. *Database*. doi:10.1093/database/bar026

APPENDIX

APPENDIX A: PATHWAY ENRICHMENTS FOR ALL 12 CANCERS

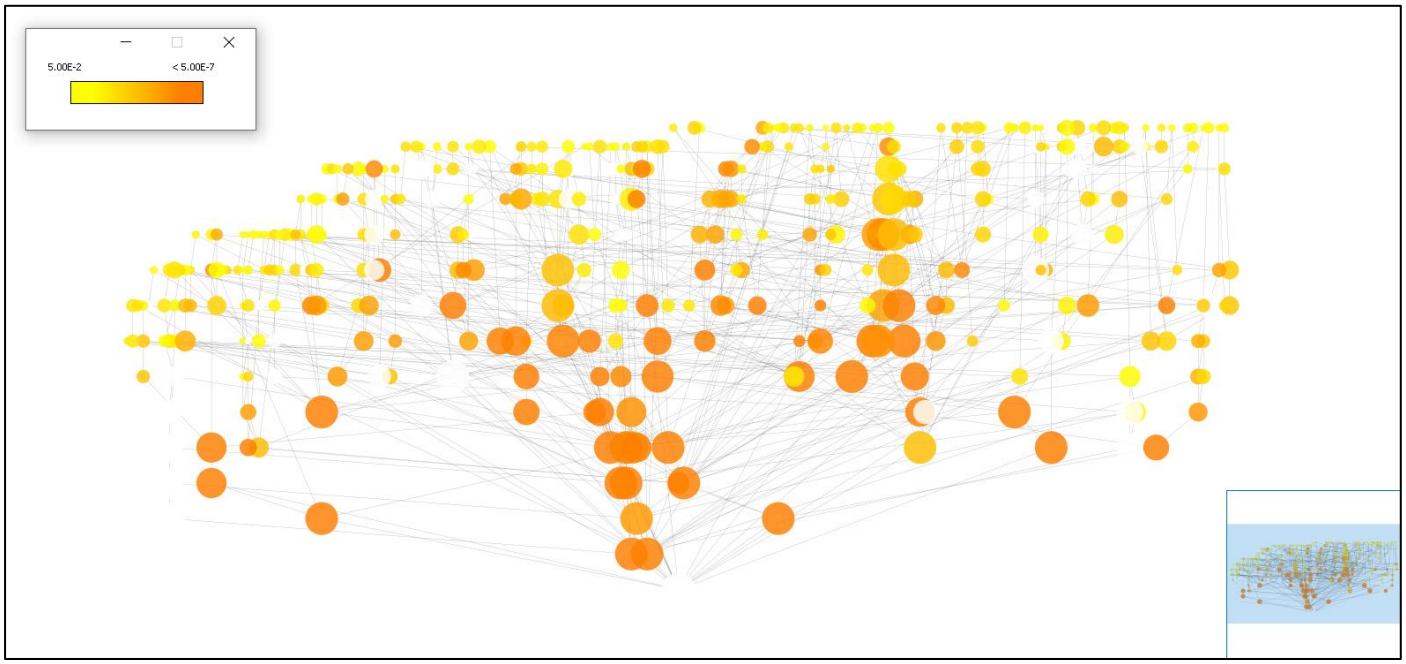

Figure 1. Bladder carcinoma (BLCA), 419 ontologies represented.

Figure 2. Breast carcinoma (BRCA), 314 ontologies represented.

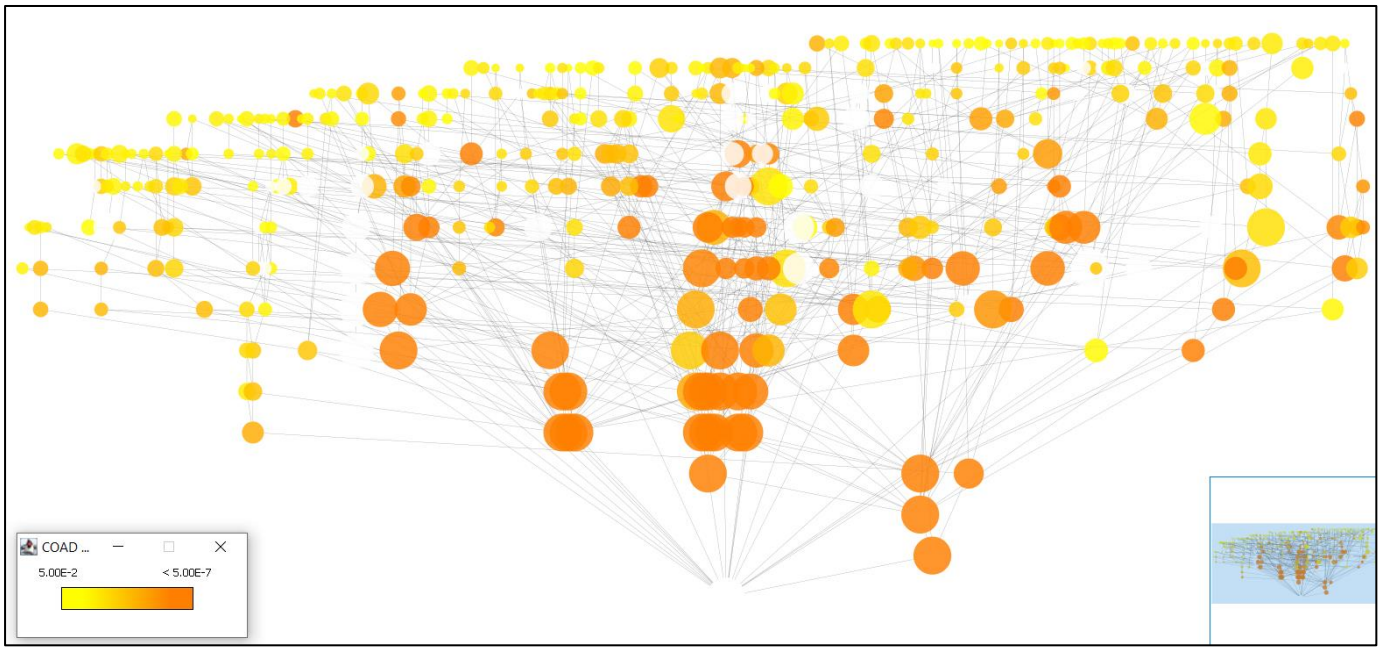

Figure 3. Colon adenocarcinoma (COAD), 414 ontologies represented.

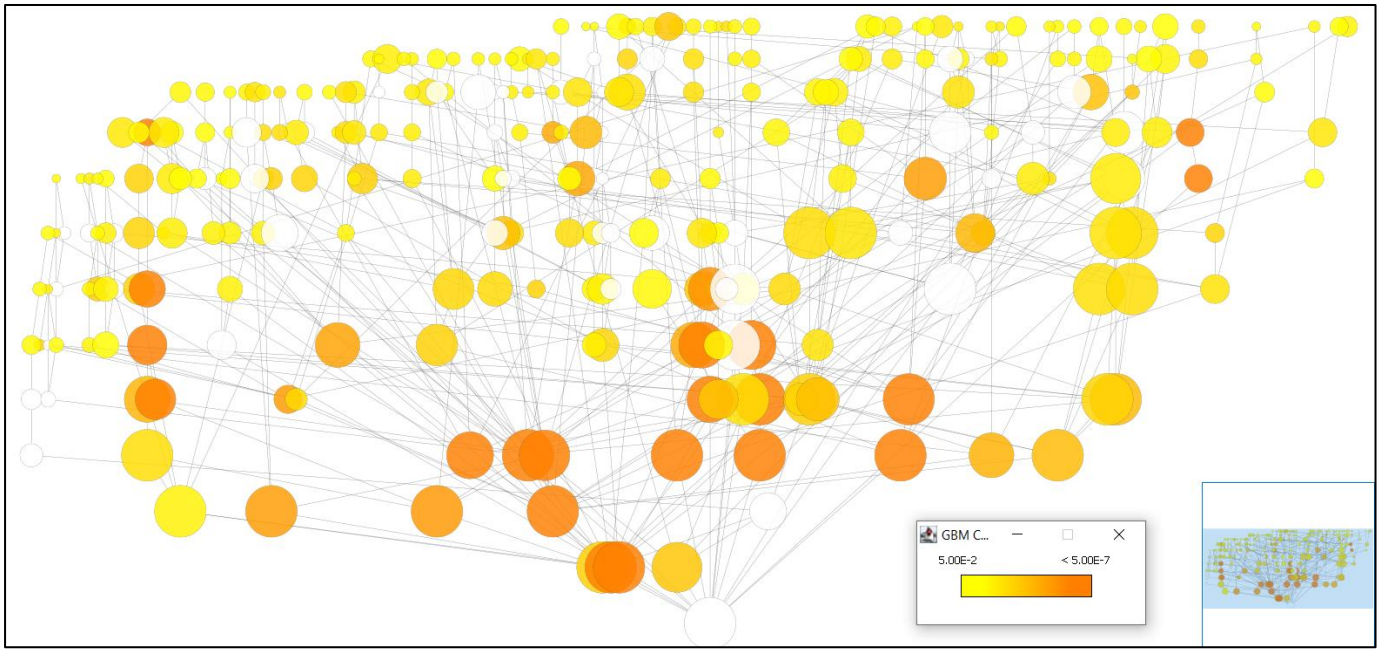

Figure 4. Glioblastoma multiforme (GBM), 303 ontologies represented.

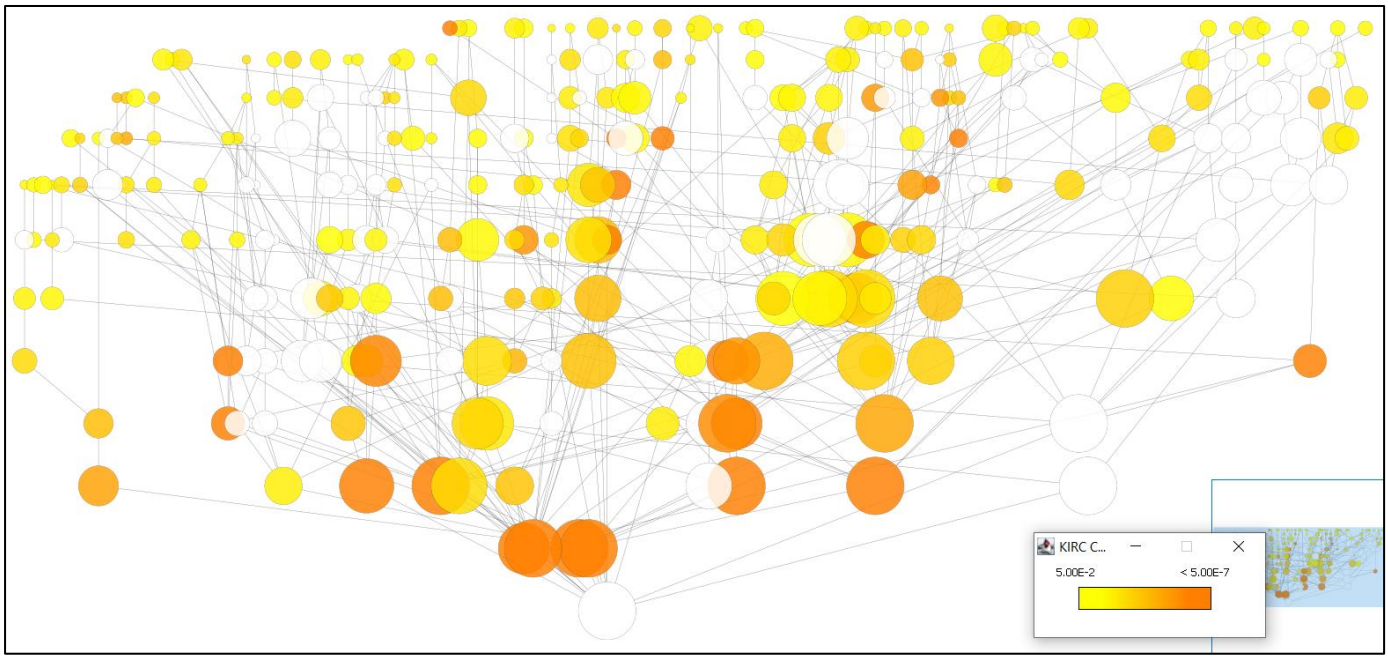

Figure 5. Kidney renal clear cell carcinoma (KIRC), 301 ontologies represented.

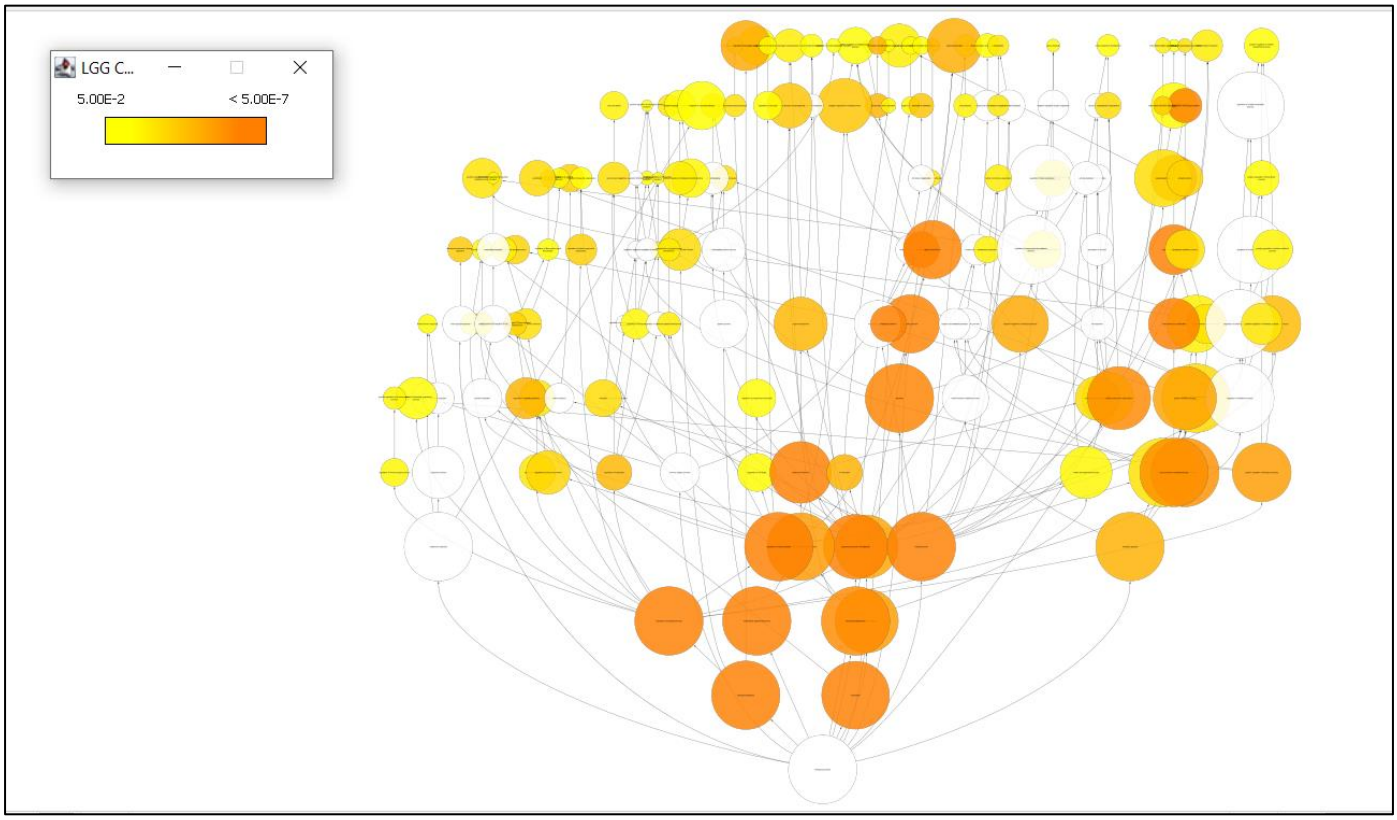

Figure 6. Low Grade Gliomas (LGG), representing 177 gene ontologies

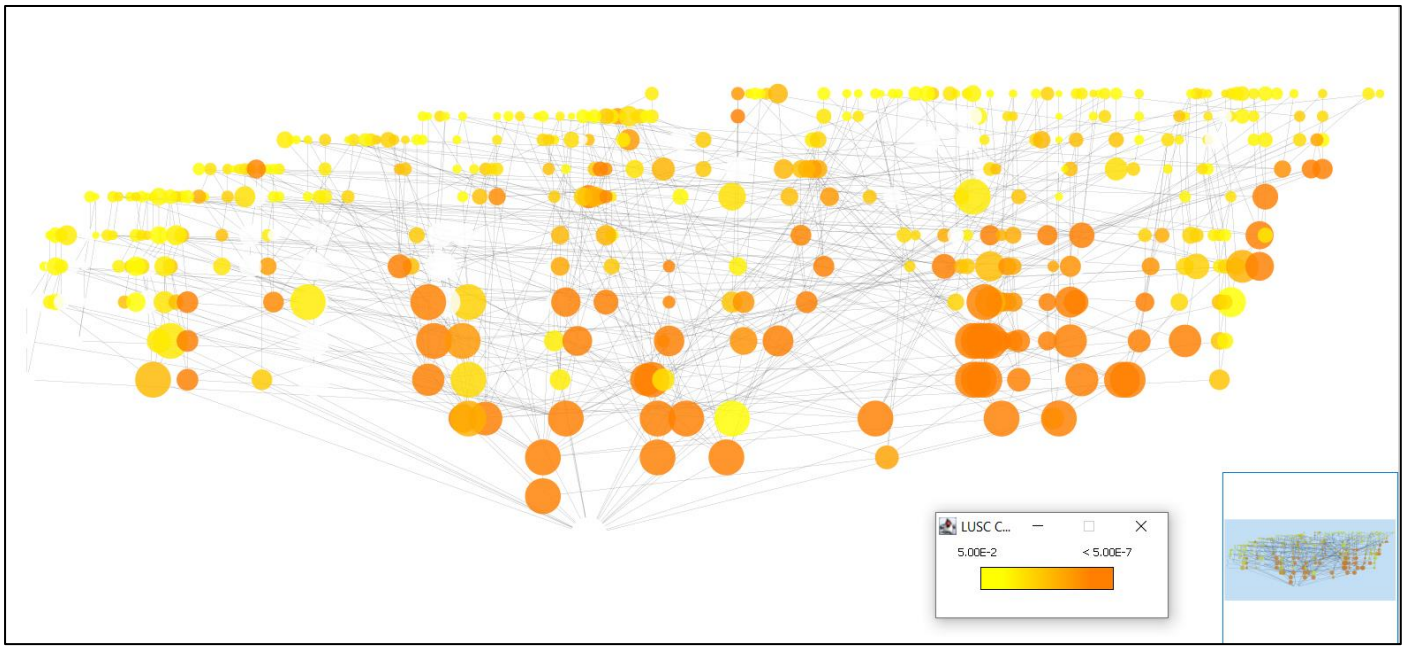

Figure 7. Lung squamous cell carcinoma (LUSC), 440 ontologies represented, the highest number in our dataset.

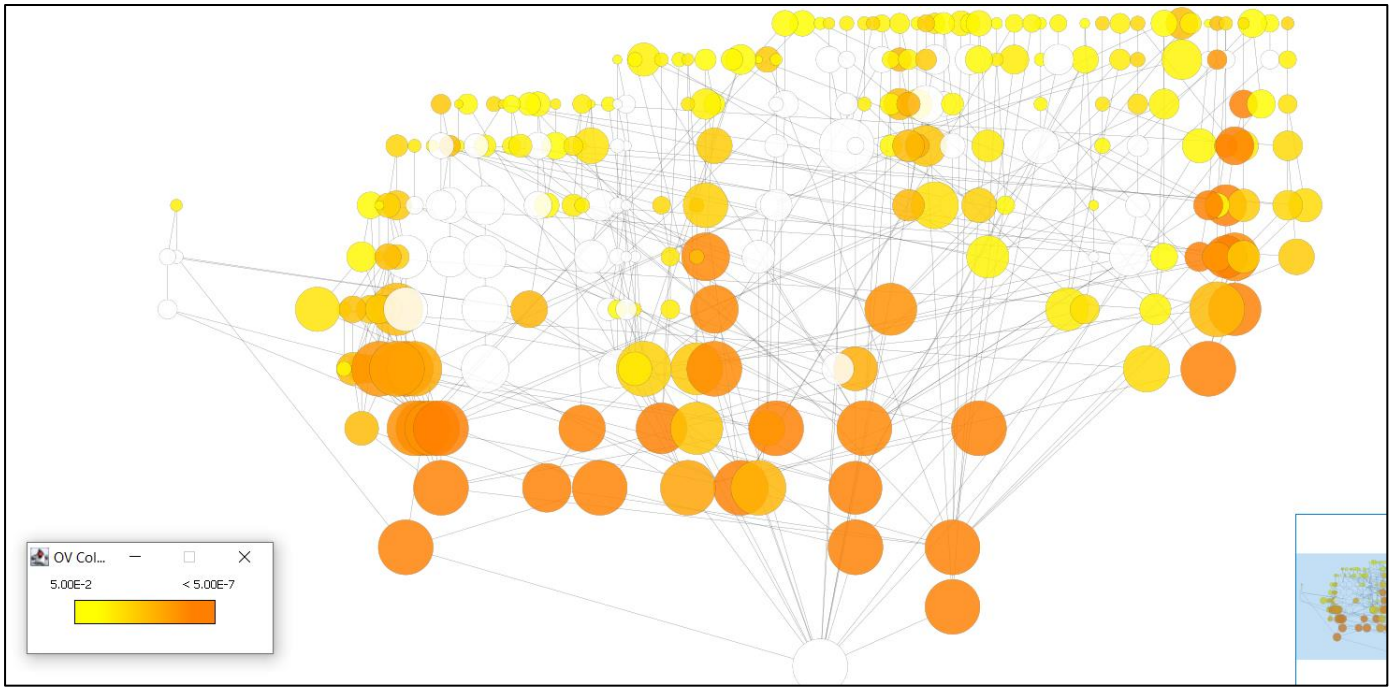

Figure 8. Ovarian cancer (OV), 268 ontologies represented.

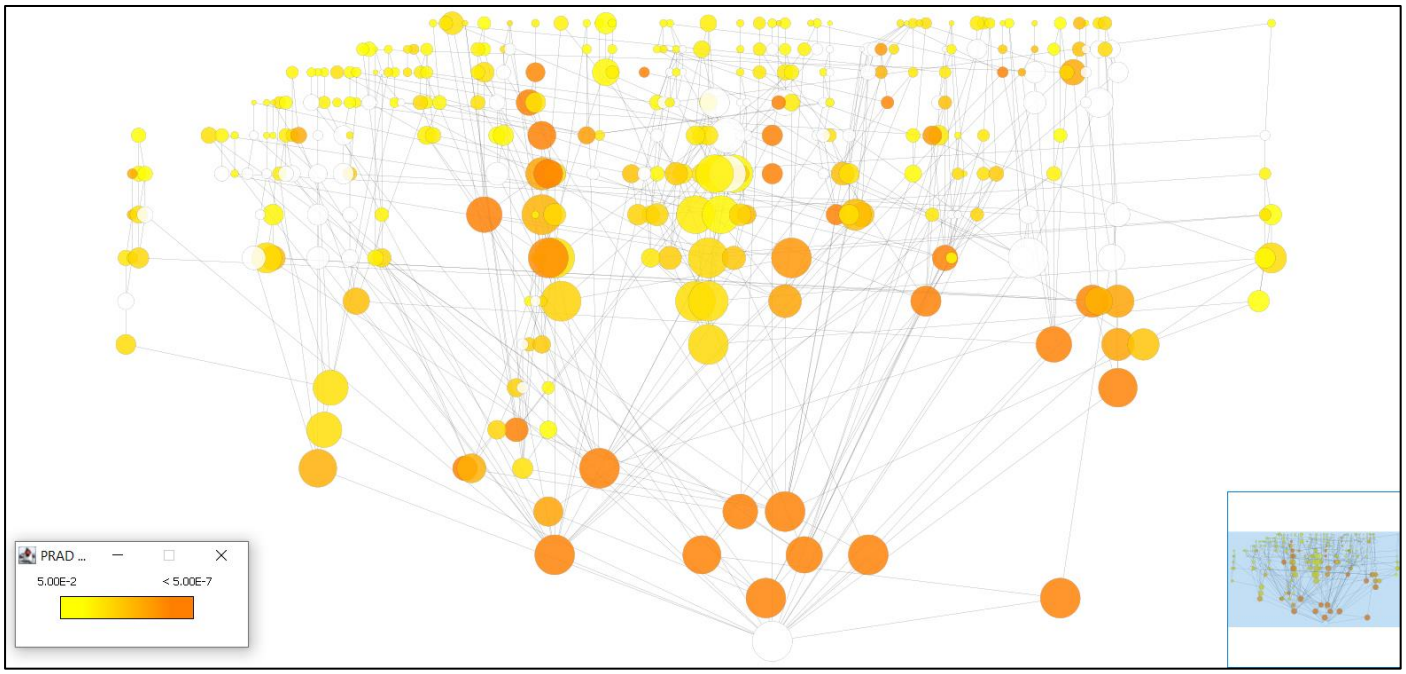

Figure 9. Prostate adenocarcinoma (PRAD), 349 ontologies represented.

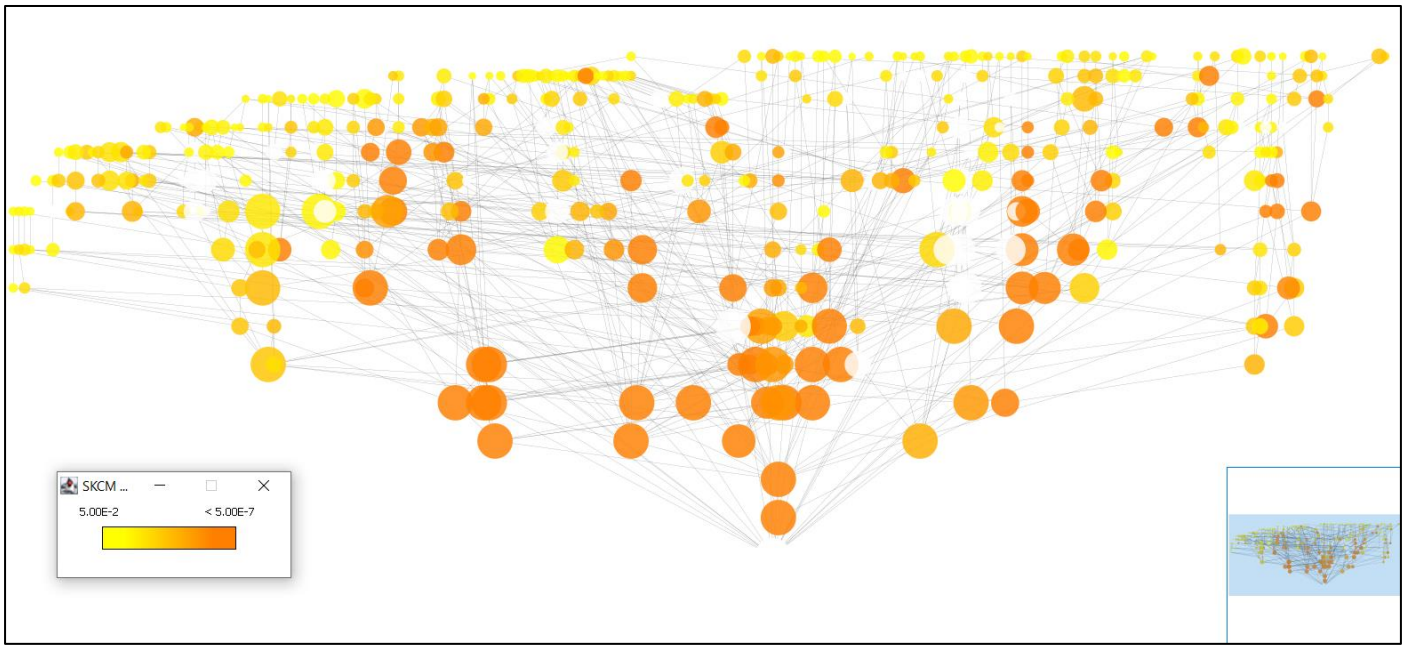

Figure 10. Skin cutaneous melanoma (SKCM), 430 ontologies represented.

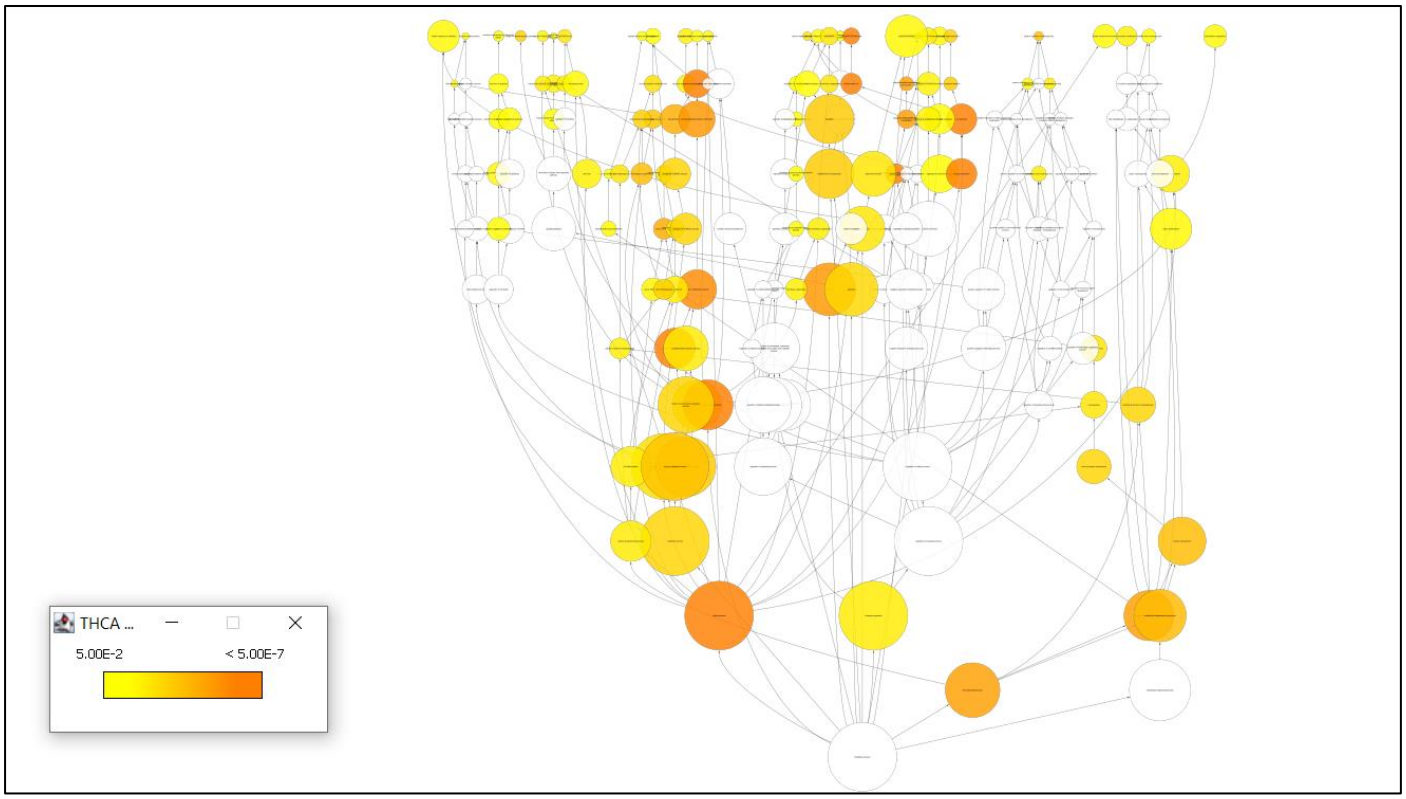

Figure 11. Thyroid carcinoma (THCA), 187 ontologies represented.

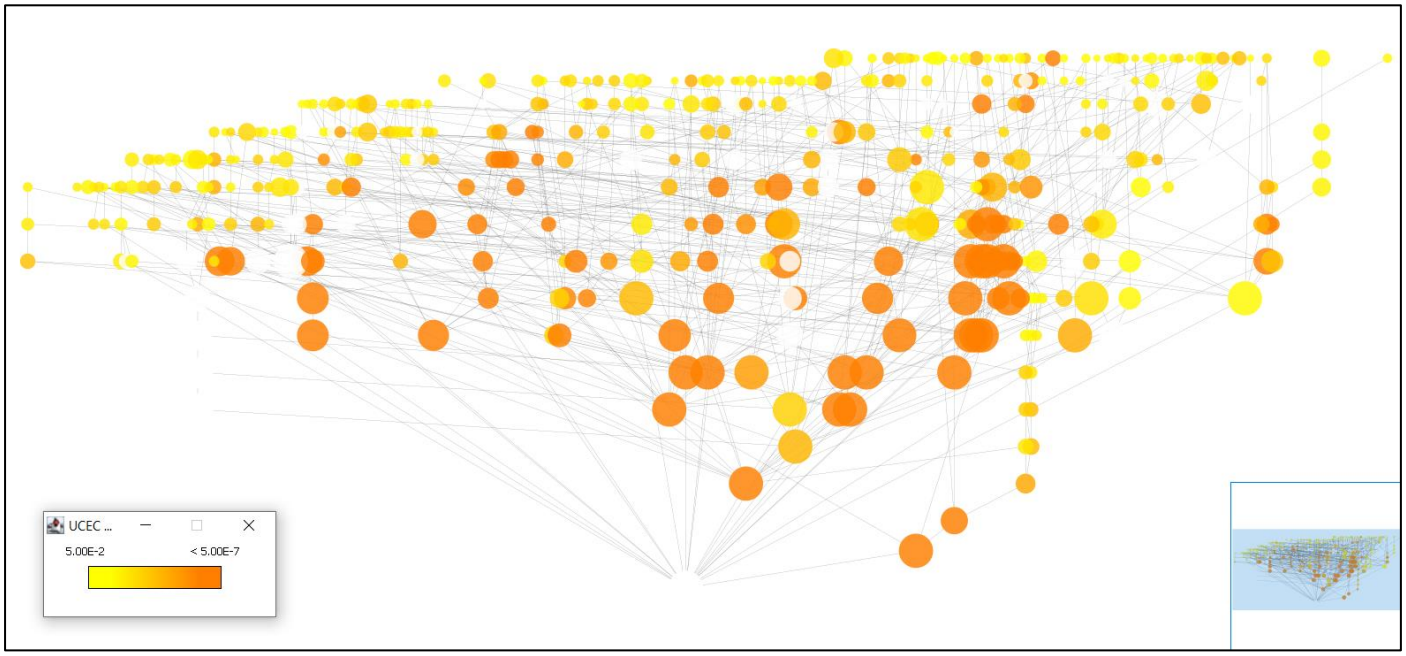

Figure 12. Uterine corpus endometrial carcinoma (UCEC), 433 ontologies represented.

APPENDIX B: GENE SYMBOL PARSER

Python parser to translate ensembl gene ID's to gene symbols using MyGene database.

```
import sys
from mygene import MyGeneInfo

import sys

stdoutOrigin=sys.stdout
sys.stdout = open("UCEC.txt", "w")

mg = MyGeneInfo()

genes = data.iloc[:,1]

results = mg.querymany(genes, scopes=["ensembl.gene"],
fields=["symbol"], species="human", verbose=False)

for res in results:
    q = res['query']
    s = 'NA'
    if 'symbol' in res:
        s = res['symbol']
    sys.stdout.write('{}\t{}\n'.format(q, s))

sys.stdout.close()
sys.stdout=stdoutOrigin
```

APPENDIX C: ADDITIONAL CONFUSION MATRICES

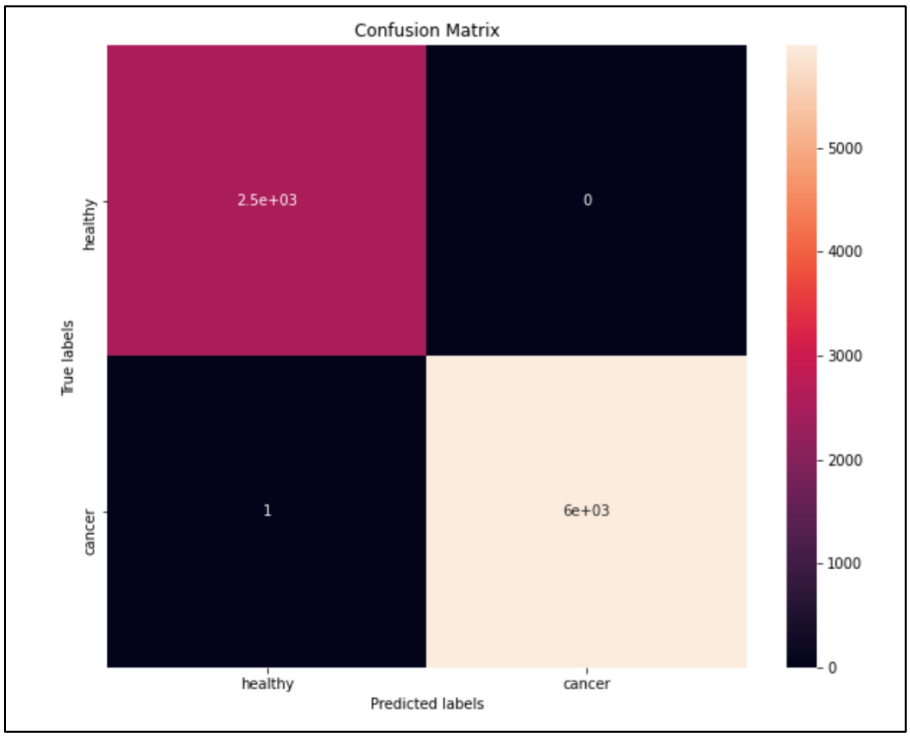

Figure 1. Confusion matrix for binary classifier on full dataset of 8491 healthy and cancer samples.

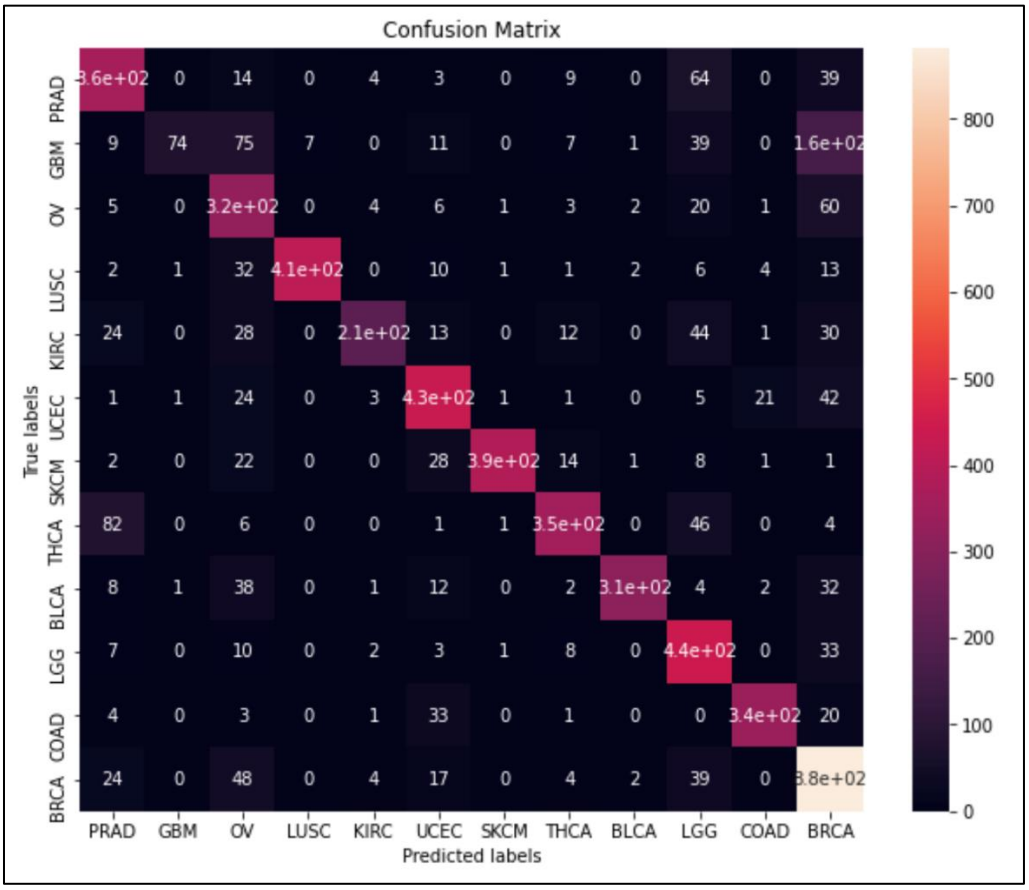

Figure 2. Confusion matrix for cancer type classifier on full set of 5987 cancer samples.