Machine Learning Classification of Sarcoma Types using Gene Expression Data

Jeffrey S. DeLong, MA, Mathematics

**Abstract**

*Soft tissue sarcomas are a diverse set of cancers that are challenging to diagnose and treat. We present a machine learning classification analysis of six distinct types of adult soft tissue sarcomas utilizing gene expression data from The Cancer Genome Atlas (TCGA) with a goal of understanding how machine learning models might be used to diagnose the sarcoma type. We explore several approaches, including dimensionality reduction followed by supervised machine learning algorithms as well as a multi-layer perceptron, feed forward, artificial neural network. The analysis was challenging due to the limited availability and unbalanced nature of sample data. Notwithstanding these limitations, we were able to demonstrate models that are able to classify 5 of 6 types of sarcoma with a fairly high degree of accuracy. The one exception is likely due to the similarity between two of the sarcoma types.*

**Introduction**

Soft tissue sarcomas are a diverse family of mesenchymal malignancies. They start in the connective tissue of the body, which include muscle, fat, nerves, tendons, veins, and the lining of joints, and account for about 1% of all adult solid tumors. SEER ([https://seer.cancer.gov](https://seer.cancer.gov/)) estimates there will be 12,750 new cases of soft tissue sarcoma in 2019, and that 5270 people will die as a result of soft tissue sarcoma. Soft tissue sarcoma is slightly more prevalent in men than women, and the median age at diagnosis is 60 years old. Soft tissue sarcomas are distinct from other solid tumors, which occur in epithelial tissue which line the organs of the body. There are more than 50 types of soft tissue sarcoma, and they are classified based on the connective tissue they most resemble. The same type of sarcoma can be found in different locations within the body, e.g., limbs, trunk, neck, retroperitoneum, etc. Due to the many types of sarcomas and similarities among them, diagnosis based on histology can be difficult, and information about the sarcoma genesis process and the impact of genomic factors on survival or treatment options is only recently being discovered.

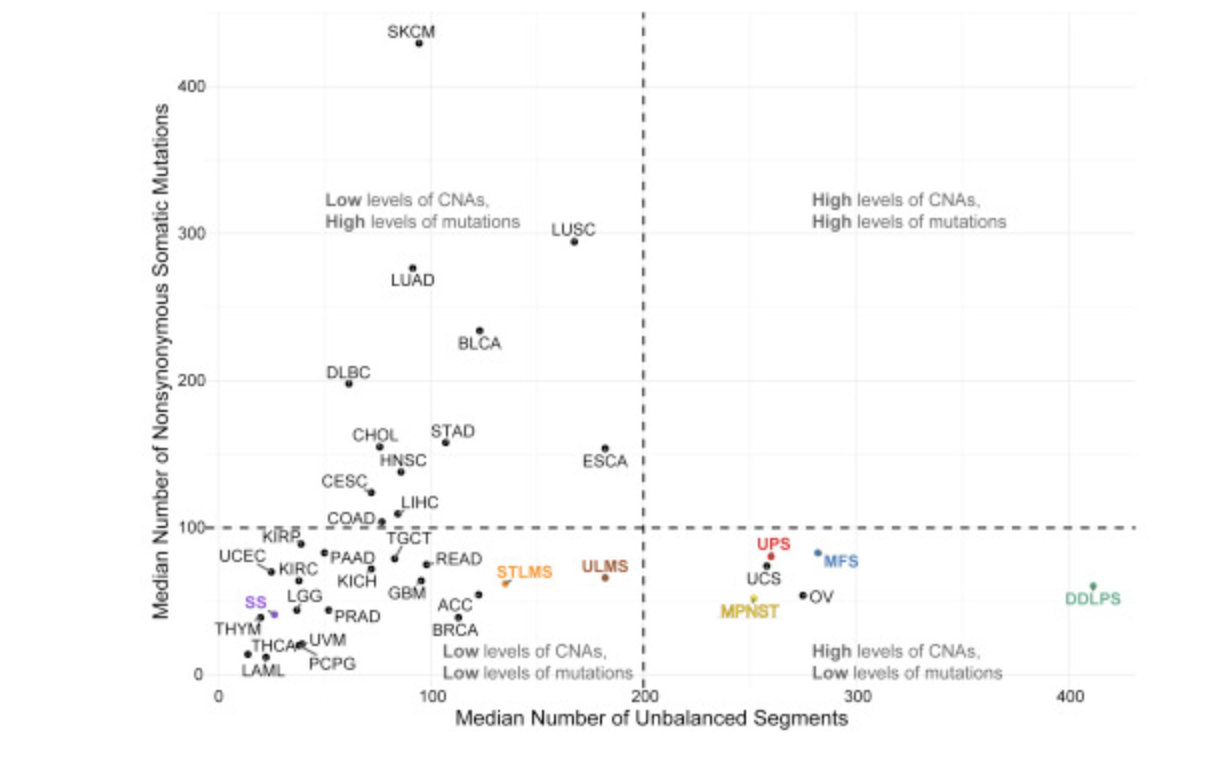
As genomic data has become available many different machine learning algorithms have been utilized to analyze cancer types and better understand their biological implications. In particular, Support Vector Machines and Artificial Neural Networks have exhibited good performance in dealing with the large number of complex features found with genomic data1. In 2012 the National Cancer Institute Genomic Data Commons established the Cancer Genome Atlas (TGCA), a database of genomic data, with the goals to generate, quality control, merge, analyze and interpret molecular profiles at the DNA, RNA, protein and epigenetic levels for hundreds of clinical tumors representing various tumor types and their subtypes.2 Gene expression data is particularly interesting as it a good indicator of how the cell is actually performing. Gene expression data has over 20000 features, which is beyond the capabilities of many machine learning algorithms to analyze. Available options to address the magnitude of features are unsupervised feature reduction techniques in combination with machine learning algorithms such as Support Vector Machines and Logistic Regression as well as Artificial Neural Networks. To demonstrate the viability of Artificial Neural Networks, TCGA gene expression data was used to build a deep learning cancer classification model in the book Java Deep Learning Projects3. This model was used to classify five types of cancer: prostate, lung breast, kidney, and colon using TCGA gene sequencing data downloaded from the UCI Machine Learning site <https://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq>. It contains over 800 cases with over 20,000 features per case. The deep learning model described in the book exhibits accuracy of over 99%.

In 2017 TCGA researchers prepared a comprehensive analysis of adult soft tissue sarcomas4, in which they studied 206 adult soft tissue sarcomas: 80 Leiomyosarcoma (LMS), 50 Dedifferentiated liposarcoma (DDLPS), 44 Undifferentiated pleomorphic sarcoma (UPS), 17 Myxofibrosarcoma (MFS), 10 Synovial sarcoma (SS), and 5 Malignant peripheral nerve sheath tumor (MPNST). See Table 1.

|  |  |
| --- | --- |
| **Sarcoma Type** | **Sample Count** |
| LMS (ULMS and STLMS) | 80 |
| DDLPS | 50 |
| UPS | 44 |
| MFS | 17 |
| SS | 10 |
| MPNST | 5 |

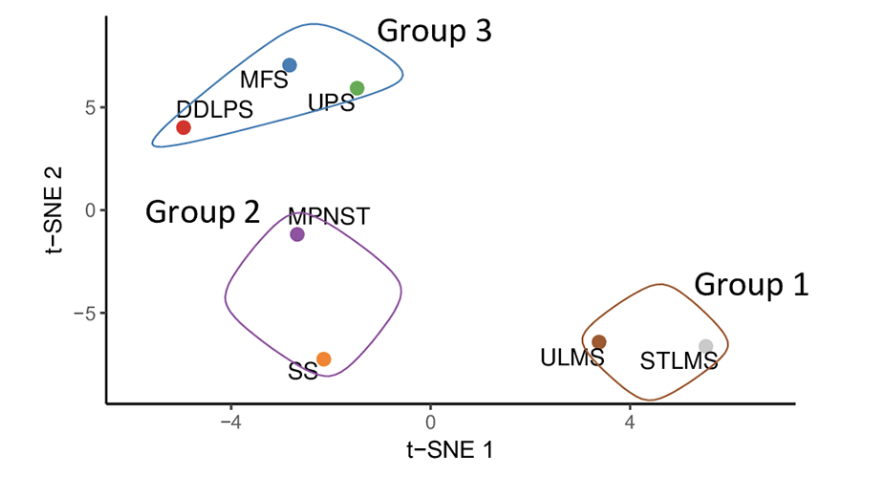
**Table 1. Sarcoma Types**

These cases were collected and processed as part of the TCGA-SARC project. The study used a variety of statistical and machine learning techniques (predominantly unsupervised hierarchical clustering) to provide an integrated genome-scale analyses of gene expression, microRNA, protein, and alterations of DNA sequence, methylation, and copy number to understand the genomic diversity of oncogenic drivers, to refine clinical risk stratification, and to identify potential therapeutic targets. The results of the study show that the overall somatic mutation burden in these 206 sarcomas was relatively low compared to other cancers, with only three genes, TP53, RB1, and ARTX, highly recurrently mutated across sarcoma types. Instead, adult soft tissue sarcomas exhibit frequent somatic copy-number alterations (SCNAs) relative to other forms of cancer (see Figure 1). They utilized unsupervised cluster analysis of somatic copy-number alterations (SCNAs) to divide the cases into 6 major clusters C2 with relatively few unbalanced segments, consisting mostly of DDLPS and SS; C3, consisting mostly of DDLPS with complex copy-number alterations; C4 and C5, dominated by LMS; and C1 and C6 consisting mostly of UPS and MFS.”



**Figure 1.** Landscape of Genomic Alterations in 206 Sarcomas

More recently, the TCGA-SARC RNA-seq count data has been analyzed with a combination of machine learning technologies5. Gene expression samples for soft tissue sarcomas from TCGA were combined with gene expression samples from tissue without malignancies from the Genotype Tissue Extension project and analyzed using a deep learning neural network to identify similarities between normal tissue and tumors. They also performed unsupervised t-Distributed Stochastic Neighbor Embedding on the gene expression data to identify 3 main clusters: STLMS and ULMS (both forms of LMS); DDLPS, UPS, and MFS; and SS and MPNST (see Figure 2). Finally, they utilized a random forest algorithm to identify diagnostic markers for soft tissue sarcoma that distinguished between synovial sarcoma and MPNST, as well as a k-nearest neighbor algorithm to identify prognostic genes that are strong predictors of disease outcome.



**Figure 2.** t-SNE analysis of all soft tissue sarcoma samples

In this study we present a machine learning classification of the six types of adult soft tissue sarcomas using gene expression data. We first reduce the number of features by eliminated data that is primarily zero, and then z score standardized the remaining data. We then use Principal Component Analysis to select the feature that explain almost all of the variance, and evaluate a series of models using the principal components. From the is exercise we identify Logistica Regression and Support Vector Machine as the best models. We also analyzed a multi-layer perceptron model. We achieved an overall accuracy of approximately 80%, with many of the errors the result of misclassification of MFS as UPS, most likely due to the molecular similarity between UPS and MFS. The overall accuracy is increased to over 90% when excluding the MFS samples. After excluding those samples, we built a BaggingClassifier model with the Logistic Regression, with additional synthetic data created with SMOTE, to build a model that achieved 97% accuracy.

**Materials and Methods**

*Hardware and Software Environment*

For this study we performed data pre-processing and standardization using Knime Analytics Platform. Machine learning models were created using Python, skikit-learn, and Keras. For the Keras deep learning models we used PlaidML as the backend. The hardware platform is a MacBook Pro with 2.2 GHz Intel Core i7 processor, 32GB RAM, and Radeon Pro 560X GPU. PlaidML was utilized because it supports OpenCL across various platforms and Apple Metal on Mac with various GPUs, including Radeon models. PlaidML performed about 50 times as fast as TensorFlow on the CPU, however it was exhibited some instability with long running tests.

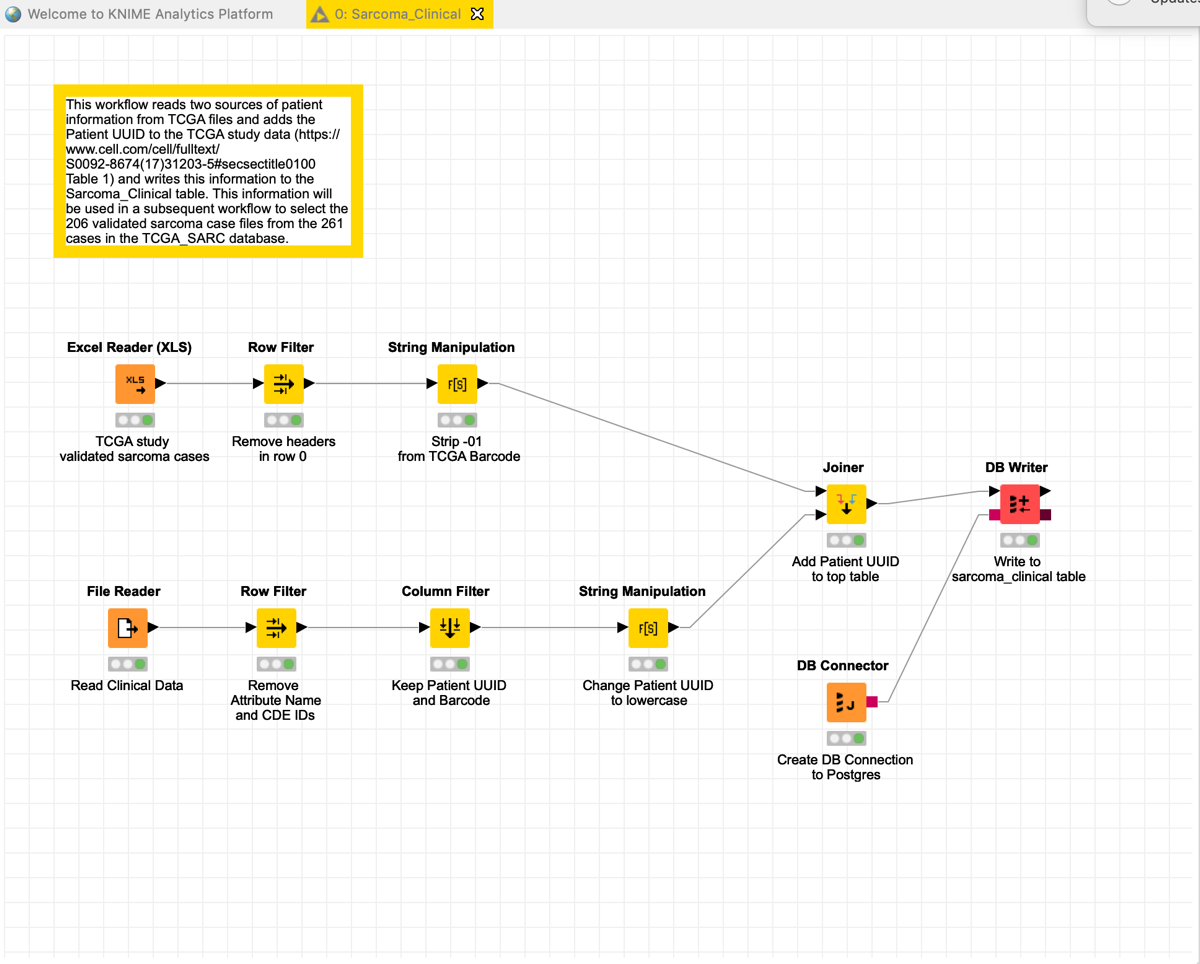
*Data Selection and Setup*

The process of transcribing a gene’s DNA sequence into the RNA that serves as a template for protein production is known as gene expression. A gene’s expression level indicates the approximate number of copies of that gene’s RNA produced in a cell. This is considered to be correlated with the amount of corresponding protein made. Cancer can be described as a disease of altered gene expression. There are many proteins that are turned on or off (gene activation or gene silencing) that dramatically alter the overall activity of the cell. A gene that is not normally expressed in that cell can be switched on and expressed at high levels. Or conversely a gene that is normally expressed (e.g., a tumor suppressor such as TP53) might be silenced. These changes can be the result of a single mutation, additional copies or deletions, or changes in gene regulation (epigenetic, transcription, post-transcription, translation, or post-translation).

*Download and Store Clinical Data*

For this study we performed data pre-processing and standardization using Knime Analytics Platform. The primary source of data for this analysis is TCGA-SARC project cases found at <https://portal.gdc.cancer.gov/>. In particular the 206 cases analyzed in [4] and [5] are utilized here. The associated patient data was downloaded from

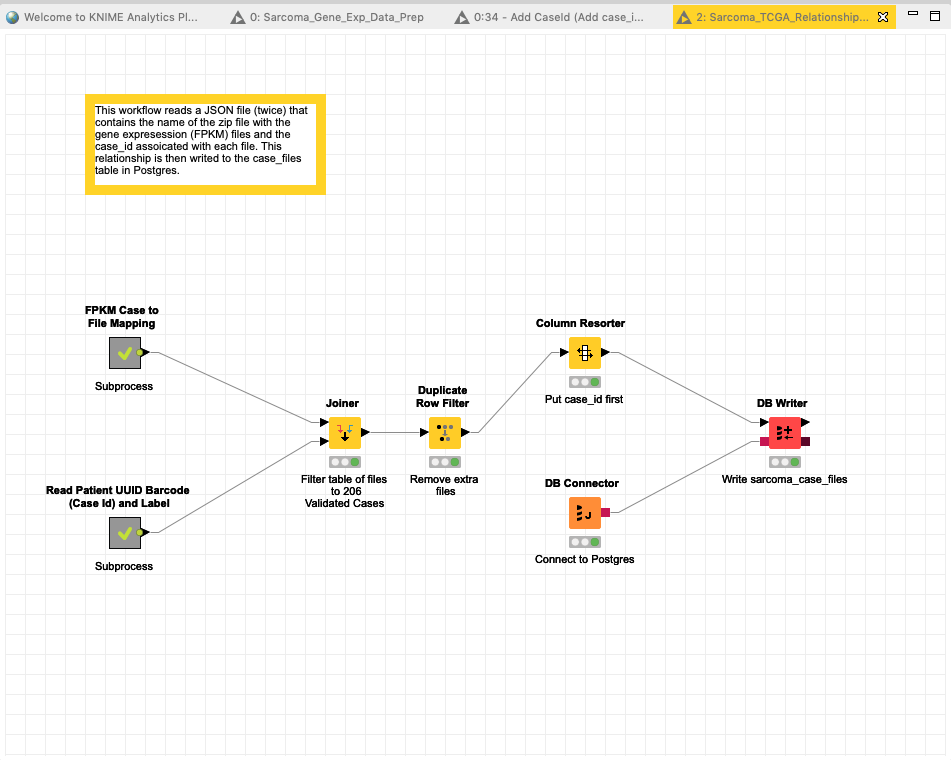
<https://www.cell.com/ell/fulltext/S0092-8674(17)31203-5#secsectitle0100> Table 1. This data was used to identify the 206 cases (from the total 261 cases available in TCGA-SARC). We also downloaded nationwidechildrens.org\_clinical\_patient\_sarc.txt from TCGA-SARC. This file also contains patient information and in particular the Patient UUID which is used to identify the specific gene sequencing data files. The pertinent information from both files was extracted and persisted into a Postgres table, sarcoma\_clinical, using the Sarcoma\_Clinical workflow. See Figure 3.



**Figure 3.** Sarcoma Clinical Workflow

*Create mapping of FPKM files to Case Id*

We downloaded a JSON file from <https://portal.gdc.cancer.gov/> that associates each gene expression file with a Case Id, and processed this file in the Sarcoma\_TCGA\_Relationships\_FPKM workflow to create a database table, sarcoma\_case\_files, with the caseid, file name, and label (sarcoma type). See Figure 4.



**Figure 4.** Sarcoma TCGA Relationships Workflow

*Download FPKM Files*

The gene expression data provided in FPKM files were downloaded from GDC web site using the GDC Client tool. Fragments Per Kilobase of transcript per Million (FPKM) files are available from GDC as tab delimited files with the Ensembl IDs in the first column and the expression values in the second. The files contain 60,484 rows each, though many of the values are zero.

*Data Preprocessing and Standardization*

Initial pre-processing of the FPKM files included transposing each file, adding each case as a row in a table, and removing features where the genes expression values are 0 in more than 90% of the cases. This reduced the number of features to 20619. After preparing the environment and reading in the data, we used Knime to normalize the data using z score standardization, and prepared CSV files both normalized and non-normalized, with and without labels, in the Sarcoma\_Gene\_Exp\_Data\_Prep workflow. See Figure 5.

## 

**Figure 5.** Sarcoma Gene Expression Data Preparation Workflow

*Descriptive Analysis and Machine Learning Algorithms*

We used the data (both non-standardized and standardized) to perform several unsupervised learning experiments using Python skikit-learn in order to better understand the data and identify opportunities to reduce the number of features. In particular we performed Chi Squared analysis (using non-standardized data), K-means Clustering, Hierarchical Clustering, Principal Component Analysis, and t-Stochastic Neighbor Embedding. We then executed several machine learning models, including K Nearest Neighbor, Support Vector Machine, Random Forest, and Logistic Regression on the various feature reduced subsets identified by the unsupervised techniques. Before executing the models, we split the dataset between training and test 75/25. We then used Kfold Cross Validation to split the training set into 4 training folds and 4 cross-validation folds and ran each model with baseline (typically default) parameters. Finally, we executed each model using Grid Search with one or two of the most commonly adjusted hyperparameters varied as part of the search. The best model was identified from the four models, re-fit with the training data, and tested with the test data. The associated statistics and confusion matrix were then displayed.

We also built a Multilayer Perceptron deep learning model in Keras with the PlaidML backend. This model consisted of an input layer, two hidden layers, and output layer with 5000 neurons in each hidden layer. We selected a batch size of 32, Stochastic Gradient Decent optimizer, and ReLU activation function. Most tests were initially run with 10 epochs, and training and validation loss and accuracy were plotted and reviewed.

For all supervised models we used Accuracy as our evaluation metric as this is a multi-class classification problem with no particular bias as to the type of classification error. That is, unlike a binary classification problem, say for example a diagnosis of malignant vs benign where a false negative of benign (not malignant) might be considered worse than a false positive of malignant, we consider all misclassifications to be of similar impact. For clustering we used purity as label data was available for this calculation.

For the Multilayer Perceptron model, we created and reviewed the plots comparing training and validation loss as well as training and validation accuracy in Python to understand the bias and variance associated with the model. Once we established a baseline model, we ran several experiments in Python adjusting the size of the neural network, l1 and l2 regularization, and dropout before executing a hyperparameter tuning exercise using Talos.

We also experimented with using SMOTE in an attempt to address our unbalanced data problem.

**Results**

*Models with Reduced Feature Set*

Chi Squared

Chi squared analysis was performed on the non-standardized feature data using Python skikit-learn with KBest set to 200. These 200 best features were then z-score standardized and a grid search was performed using the four classifiers. The best models were SVM with C=1, Gamma=0.001, and RBF kernel and Logistic Regression with C=2.78. Both models showed a test accuracy of 79%. The list of Ensemble Ids for the 200 best features was also captured and the associated gene symbols identified for subsequent analysis.

Principal Component Analysis

Principal Component Analysis was performed on the z-score standardized FPKM data using Python skikit-learn with variance set to .95. This resulted in 163 components. I.e., 163 components preserve 95% of the variance in the 20000 plus gene expression levels. A grid search was performed with the 163 components using the 4 classifiers. The best models were Logistic Regression with C=1 and a test accuracy of 83%, and SVM with C=1, Gamma=0.0001, RBF kernel and test accuracy 79%.

Clustering – 6 clusters

We performed hierarchical agglomerative clustering with affinity='euclidean' and linkage='war' using Python skikit-learn with the number of with number of clusters set to 6. This identified the following clusters: C1 dominated by LMS, C2 consisting of UPS and MFS, C3 consisting of primarily UPS, C4 dominated by DDLPS, C5 with a mix of DDLPS and others, and C6 dominated by SS. This is similar to what [4] reported for SCNA clustering. The purity of this cluster is .73. We then added the cluster number to the PCA component feature set and ran a grid search using the 4 classifiers. The best models were Logistic Regression with C=1 and a test accuracy of 81%, and SVM with C=1, Gamma=0.0001, RBF kernel and test accuracy 79%.

We also performed k-means clustering using Python skikit-learn with the number of number of clusters set to 6 and identified similar clusters as with hierarchical clustering. The purity of the k-means cluster was .62.

Clustering – 3 clusters

The two clustering algorithms were also run with number of clusters equal to 3. Both approaches identified the following three clusters: MPS, UPS, and DDLPS clustered together, ULMS and STLMS clustered together, a mix of DDLPS, SS and MPNST. These results are similar to the findings reported in [5].

T-Stochastic Neighbor Embedding

T-Stochastic Neighbor Embedding was performed on the z-score standardized FPKM data with the number of components set to 2, a perplexity of 60 and an angle of 0.5 using Barnes-Hut algorithm. After the two components were calculated, the label data was added, the data was grouped based on label values (sarcoma types), the mean of each group was calculated, and the means plotted. The results are similar to those reported in [5].

Labels were then added to these new features, and Grid Search was run with the 4 classifiers. The best model was SVM using the RBF kernel, C=1, and Gamma = Scale. This model exhibits .54 accuracy on the test data. This result is inferior to SVM using PCA transformed data or Chi Squared reduced data.

In all cases examination of the confusion matrix shows a strong tendency to misclassify MFS as UPS.

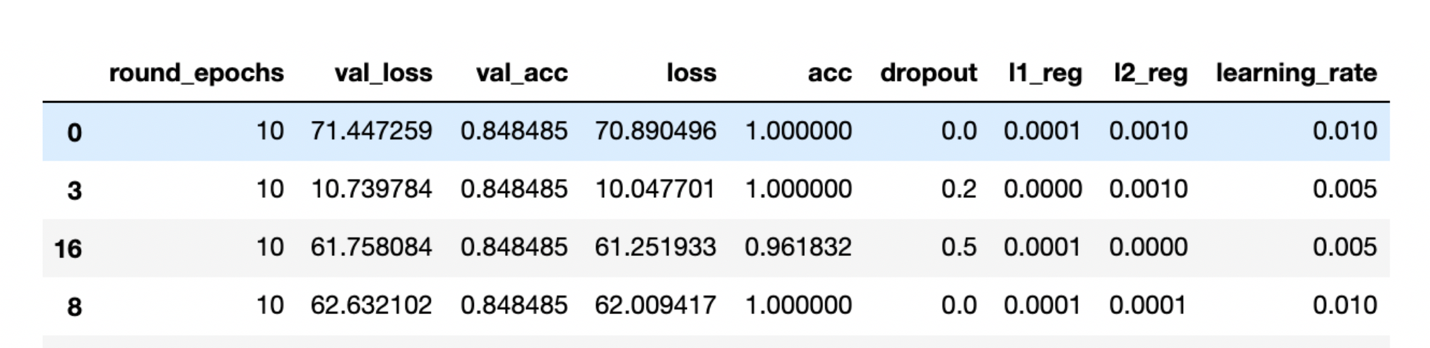
*Multi-layer Perceptron*

Baseline

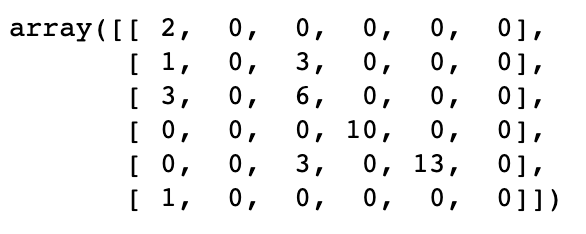
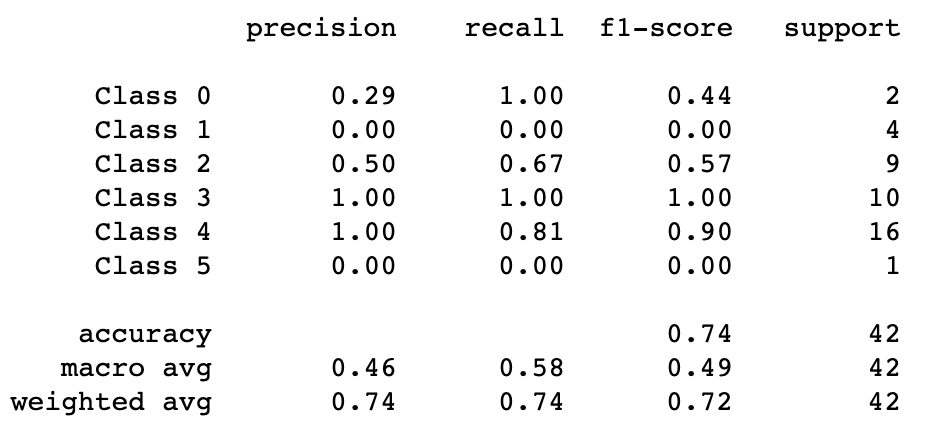
A baseline Multilayer Perceptron model was created using Python with Keras and the PlaidML backend. These baseline models were used to classify sarcoma types with no hyperparameter tuning or feature extraction. The model data is the 206 sarcoma cases and z-score normalized 20605 features. The labels were categorized using the Keras to\_category method. Data was split 60/20/20 between training, validation, and test. Accuracy was used to evaluate the models. The model consisted of one input layer, two 5000-unit hidden layers with ReLU activation function, and an output layer with Softmax activation function and Multiclass Cross Entropy as the loss function. The initial model used a batch size of 32, 10 epochs, Stochastic Gradient Descent optimizer, and the Keras default learning rate (0.01). This model had a training accuracy of 73%.

### Hyperparameter Tuning

Hyperparameter tuning was conducted using Talos. The hyperparameters tuned included: Learning Rate, L1 Regularization, L2 Regularization, and Dropout. See Table 2. The top four models all had a validation accuracy of 85%. Testing this model with the test data resulted in an accuracy score of .74. See Table 3



**Table 2. Hyperparameter Tuning results from Talos. Best 4 out of 16 models.**



**Table 3. Metrics and Confusion Matrix from best MLP model evaluated against test data**

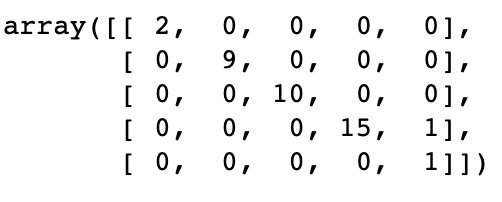
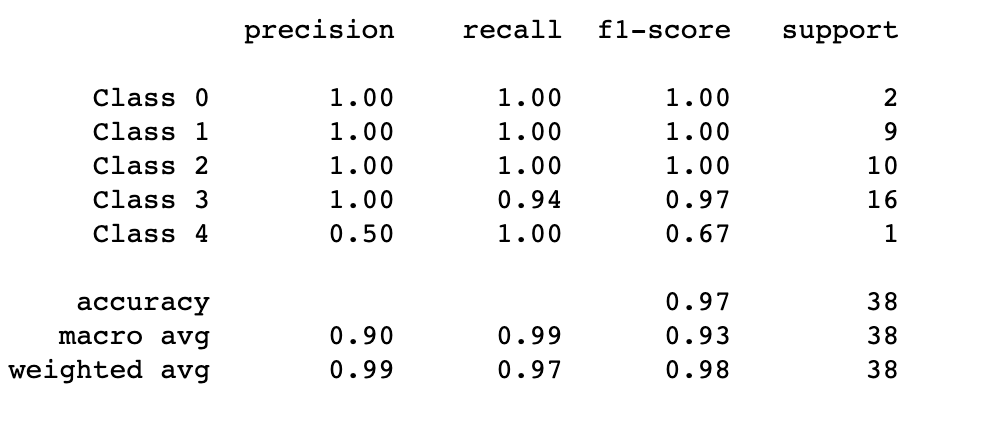
**Discussion**

The various feature reduction and unsupervised learning techniques did show some similarities to the results reported in [4] and [5]. The machine learning models exhibited good performance with some of the classifiers we tested. SVM and Logistic Regression were the best classifiers when utilized with the reduced data set, whereas Random Forest and KNN consistently underperformed. The Multilayer Perceptron model results were not quite as good as the SVM and Logistic Regression results, even with hyper-parameter tuning.

All models consistently mis-classified MFS. See Table 3 Confusion Matrix second column from the left. Most of the MFS sample were misclassified as UPS. This result is explained by the similarity between UPS and MFS, which are some people consider the same disease, the only significant difference being larger amounts of myxoid stroma in MFS tumors [4].

We removed the MFS samples from out data set, re-computed the Principal Components, and re-ran the four machine learning models on the new set of principal components. Logistic Regression was the best model. With SAG solver and C of 1 we achieved an accuracy of .88. We then created a BaggingClassifier with the Logistic Regression classifier with n\_estimators=100 and max\_samples=100, and achieved an accuracy of 92%

As a final step we utilized SMOTE with a sampling strategy of ‘not majority’ to provide 48 samples of each label with the Bagging Classifier with Logistic Regression to achieve an accuracy of 97%. See Table 4.



**Table 4. Metrics and Confusion Matrix from best LR model Boosting and SMOTE evaluated against test data**

**Conclusions**

The results show that sarcoma types can be classified based on their gene expression data, with the caveat that we were not able to distinguish between MFS and UPS. The small number of samples in general, and MPNST in particular, was a challenge, particularly for the deep learning models. The additional accuracy gained using SMOTE lends credence to the notion that more data would result in more accurate predictions.

Areas of interest for future research include doing more research into the difference between UPS and MFS, using the gene expression data to predict some measure of outcome such as local recurrence, distant metastasis, or survival, and utilizing other forms of genomic data (e.g., methylation data) to perform similar classification exercises. Outcomes are available with the case data that we captured and methylation data is available for download from TCGA. Futher analysis of MFS and UPS would require identification of additional data sources and re-creating all FPKM files to have a consistent basis for comparison. We would also like to explore how we might gain insight from the models into the specific genes that are driving the differentiation between the sarcoma types. As a starting point we plan to examine the genes that resulted from our Chi Squared analysis.

**References**

1. Huang, S, Cai, N, et al. Applications of Support Vector Machine (SVM) Learning in Cancer Genomics. Cancer Genomics & Proteomics 15: 41-51. 2018. doi:10.21873/cgp.20063.
2. Weinstein, John N., et al. The Cancer Genome Atlas Pan-Cancer Analysis Project. Nature Genetics. 2013. 45(10): 1113-1120. <https://doi.org/10.1038/ng.2764>.
3. Karim, Md Rezaul. Java Deep Learning Projects: Implement 10 real-world deep learning applications using Deeplearning4j and open source APIs. Packt Publishing. 2018.
4. Abeshouse A, Adebamowo C, Adebamowo SN, Akbani R, Akeredolu T, Ally A, et al. Comprehensive and Integrated Genomic Characterization of Adult Soft Tissue Sarcomas. The Cancer Genome Research Network, Cell 171, 950-965. 2017. <https://doi.org/10.1016/j.cell.2017.10.014>
5. van IJzendoorn DGP, Szuhai K, Briaire-de Bruijn IH, Kostine M, Bovee JVMG. Machine learning analysis of gene expression data reveals novel diagnostic and prognostic biomarkers and identifies therapeutic targets for soft tissue sarcomas. PLoS Computational Biology 15(2): e10006826. 2019. <https://doi.org/10.1371/journal.pcbi.1006826>.