Cancer Subtype Classification

Ben Flügge
ben.fluegge@stud.uni-hannover.de

Meike Liedtke
meike.liedtke@stud.uni-hannover.de

Bao Tran Nguyen
bao.tran.nguyen@stud.uni-hannover.de

Hauke Schüle
hauke.schuele@stud.uni-hannover.de

Tatjana Wehrmann
t.wehrmann@stud.uni-hannover.de

I. Introduction

We used the TCGA Kidney Cancers Dataset [1] that contains transcriptome profiles of patients diagnosed with three different subtypes of kidney cancers. The dataset is used to make predictions about the specific subtype of kidney cancer. In the datasets patients with kidney chromophobe cancer are underrepresentated with only 8,85% of the total dataset. Samples of the kidney renal clear cell carcinoma represent the majority of the dataset with 59,73% and the remaining 31,42% are of the kidney renal papillary carcinoma cancer type.
After preprocessing the data, we performed a PCA. We performed the classification on the original and the PCA data both with a decision tree and a feed-forward neural network.

II. State of the art

Various approaches are applied to classify subtypes of cancer.
Rukhsar et al. [2] analyze eight different deep learning algorithms for classifying RNA-Seq data for different cancer types, finding that a Convolutional Neural Network achieved the best results in their testing. In particular for kidney cancer, Pirmoradi et al. [3] developed a deep-neuro fuzzy system to classify kidney cancer subtypes on selected miRNA. Marquardt et. al [4] used principal component analysis for data visualization and Random Forest Learning for classification. Shon et al. [5] proposed a deep learning approach for classification as a combination of an autoencoder and neural network. All approaches for kidney cancer classification used the data from TCGA [3]–[5].

III. Methodology

A. Data preprocessing

The TCGA Kidney Cancers Dataset provided 1028 files, representing the patients' transcriptome profiles (gene expression quantification). The normalized transcriptome profile data is given as TPM and FPKM. Since TPM is widely acknowledged and a preferred choice in many modern RNA-seq analysis workflows, this paper also chooses TPM for our machine learning models. We concatenated the TPM transcriptome profiles into one dataframe, with 60660 genes representing the features of the machine learning models. This new dataset contains 60661 columns because the last column is the cancer type label. The dataframe consists of 1028 rows, one for each patient, and was stored for future use as a csv file.

B. PCA

On the accumulated data we applied Principal Component Analysis (PCA) to reduce the number of features. As a first step, we standardized the data, using the StandardScaler from sklearn preprocessing. This measure is important because PCA is sensitive to the scale of features. On the standardized data, we performed the Principal Component Analysis with the PCA function of sklearns decomposition module. In our initial attempt we set the number of components to 1028, which resulted in a total explained variance ratio sum of 1.0. However, when we used that data in the machine learning process, we observed that the decision tree produced similar results to those using the original data, and the neural network produced no viable results, due to massive overfitting. We then tried the PCA again with only 200 components, achieving a total explained variance ratio sum of about 0.815. The second PCA data set had still no impact on the decision tree, but we achieved slightly better results with the neural network, the circumstances will be discussed in the results section.

C. Decision tree

As a first approach, we chose the Decision Tree Classifier from scikit-learn, because it is a simple and efficient supervised learning algorithm. The data was divided into training set (= 0.8) and test set (= 0.2). Scikit-learn also offers different parameters, e.g. min_samples_split to give a minimum threshold to stop the splitting, if the number is below the threshold. This can avoid overfitting. Similar to that the parameter min_samples_leaf gives a minimum threshold to the tree leaves. In addition, the Decision Tree classifier was also trained on the PCA dataset to see how the model works on the reduced features.

D. Neural Network

As a second approach, we chose to build a neural network as a more sophisticated machine learning model. We decided to work with pytorch and scikitlearn. The train-test-split of the data was the same as for the Decision Tree, however, the data was normalized to stabilize the training and improve convergence. We implemented a feed-forward neural network with 64 hidden layers, using Adam Optimizer and Cross

Entropy Loss. The network yielded the best results using a learning rate of 0.0005 and 50 epochs. The network consisted of 5 layers, starting with a dropout of 50% for regularization. Afterward, a Linear Layer, ReLu, a second Linear Layer, and finally a Softmax Layer to turn the logits into class probabilities. We decided against a Convolutional Network because our data did not contain images, but rather RNA-data.

IV. Results

As mentioned above the samples of cancer types in the dataset are not evenly distributed. This has effects on the results, as the model can't sufficiently learn to adapt to all classes. As a result, the accuracy of the classification for samples from this class is always lower than the other classes. For this reason, we chose the weighted average when calculating the f1 score for all models.
When using the PCA data for the two machine learning models, we received mixed results. For the Decision Tree Classifier it did not make a difference whether we used the original data, the PCA data with 100% variance, or the second version with 80% variance. With the Neural Network, using the first version of PCA data 100% variance led to massive overfitting. The model did not learn to generalize at all. Since this is a common effect of PCA data with a 100% variance we decided to reduce the variance. With this second PCA dataset, we were able to train the Neural Network to a slightly better f1 score than with the original data. However, we had to increase both the learning rate (to 0.001) and the number of epochs (to 80) for that. Therefore we needed more computational time than with the original data. Since the PCA data did not improve either of our machine learning models, all the following results and values were computed using the original dataset.

The weighted average F1 score for the Decision Tree over all three cancer types is 0.92, where class 1 performed best, which can be explained by the higher amount of data samples we have for that class. The resulting tree had a depth of 7 or 8 and the root node is the gene NDUFA4L2. This first tree had a very high granularity, with leaf nodes consisting often of only 1-3 samples. Therefore we tuned the model using the parameters min_samples_split=7, min_samples_leaf=7, or min_impurity_decrease=0.01 respectively for regularization. This reduces the tree structure and prevents overfitting. Each regularization measure led only to a slight increase in the F1 score.

After fine-tuning the Neural Network, using the learning rate, number of epochs, and hidden layers, the overall F1 score is 0.92. Similar to the results with the Decision Tree class 1 achieves the best F1 score. Higher learning grate or more epochs or layers lead to overfitting of the network. Lower values prevented the network from reaching its full potential. The number of epochs can be reduced in some cases, depending on the random seeds used for data-split and model initialization. 50 epochs was, however, the best overall setting, judging by the loss plots.

The overall results with both the decision tree and the neural network show a good performance. With the current settings, the Decision Tree is the most appropriate model, because of its simplicity and interpretability, while achieving similar results as the Neural Network. If the Neural Network can be improved to outperform the Tree, this would be a better choice when aiming for the best results without regard to the computation cost.

References

[1] M. LLC. (2023) Tcga kidney cancers. [Online]. Available: https://archive.ics.uci.edu/dataset/892/tcga+kidney+cancers

[2] L. Rukhsar, W. H. Bangyal, M. S. Ali Khan, A. A. Ag Ibrahim, K. Nisar, and D. B. Rawat, "Analyzing rna-seq gene expression data using deep learning approaches for cancer classification," *Applied Sciences*, vol. 12, no. 4, p. 1850, 2022.

[3] S. Pirmoradi, M. Teshnehlab, N. Zarghami, and A. Sharifi, "A self-organizing deep neuro-fuzzy system approach for classification of kidney cancer subtypes using mirna genomics data," *Computer Methods and Programs in Biomedicine*, vol. 206, p. 106132, 2021.

[4] A. Marquardt, A. G. Solimando, A. Kerscher, M. Bittrich, C. Kalogirou, H. Kübler, A. Rosenwald, R. Bargou, P. Kollmannsberger, B. Schilling *et al.*, "Subgroup-independent mapping of renal cell carcinoma—machine learning reveals prognostic mitochondrial gene signature beyond histopathologic boundaries," *Frontiers in oncology*, vol. 11, p. 621278, 2021.

[5] H. S. Shon, E. Batbaatar, K. O. Kim, E. J. Cha, and K.-A. Kim, "Classification of kidney cancer data using cost-sensitive hybrid deep learning approach," *Symmetry*, vol. 12, no. 1, p. 154, 2020.