

Predicting clinical features based on RNA-protein correlations in cancer patients

Hannah Boekweg, Corbin Day, Caleb Lindgren

Department of Biology
Brigham Young University

Abstract

Cancer patients have abnormal RNA/protein correlations, compared to healthy patients. We use various metric of aberrant RNA/protein correlations as features to predict clinical features (e.g survival) of cancer patients. We assess the accuracy achieved by three different models: MLP, KNN, and Naive Bayes. For survival and recurrence, all models achieve the 80 percent accuracy. For other features Naive Bayes outperformed MLP and KNN

1 Introduction

There are several molecular process that occur in order for proteins, which perform the function of cells, to be made. It begins with DNA (the instructions for the cell), which gets transcribed to RNA (the messenger), which is then able to be translated to proteins. Proteins are often referred to as the 'workhorses' of the cell, and are the molecules that actually carry out the instructions encoded in the DNA. The correlation between RNA and protein has been previously investigated [Waldbauer *et al.*, 2012] [Payne, 2015] [Han *et al.*, 2021]. We can use RNA/protein correlations to help us predict clinical outcomes of cancer patients. Cancer patients have an abnormal RNA/protein correlation when compared to healthy patients. We will use these abnormal correlations to predict patient survival, recurrence status, histology, pathology, and measure of patient recovery.

2 Methods

2.1 Data generation

Inputs

Our original proteomic and transcriptomic data was obtained from the Cancer Proteomics Tumor Analysis Consortium (CPTAC) [Rodriguez *et al.*, 2021] using the `cptac` software package [Lindgren *et al.*, 2021]. Previous work (not yet published) calculated the change between tumor and normal RNA-protein correlation, or delta correlation, for each protein across all patients in each cancer type. These correlations were calculated using the Spearman method. Orthogonal residuals for tumor and normal for each patient in each protein's correlation plot were calculated using the `scipy.odr` module.

We took the delta correlations and generated several features from them for training our models. This required some adaptation, as the delta correlations look at one protein across all patients, while we were interested in looking at one patient at a time. We thus sought to summarize the most important delta correlation information for each patient. First, we found the 25 proteins whose delta correlations had the greatest absolute value across all cancer types. Our reasoning was the the proteins with the greatest delta correlation were more likely to be informative. Then, for each patient, we calculated the change between tumor and normal residuals of the patient for each of those 25 proteins and saved these protein tumor-normal residual differences for each patient. We also noted whether that patient was above or below the tumor and normal regression lines for each protein.

After calculating these tumor-normal residual differences for the overall most important proteins in each patient, and the above or below regression line data for each protein in each patient, we calculated correlation between tumor and normal residuals for each patient in across all proteins. This used the Spearman method again, and provided with a single new feature for each patient, based on the delta correlation data.

Our final feature was not derived from the delta correlations. Instead, for each protein in each patient, we calculated the ratio of tumor RNA abundance to normal RNA abundance and the ratio of tumor protein abundance to normal protein abundance, then found the correlation for each patient between this RNA tumor-normal ratio and this protein tumor-normal ratio across all proteins. The goal was to generate a feature for each patient that reflected differences between RNA and protein data, but, unlike the delta correlation-derived data, did not depend on data across multiple patients, and thus could be calculated for a single new patient without needing to find how that data fit in with data from other patients.

Targets

Clinical data was obtained in the same manner as proteomics and transcriptomics data. For targets, we selected six categorical features that had lower numbers of missing values and had potential to be predictable based on our input features. The targets chose were survival status, recurrence status, and treatment success status of the patient at last follow-up, as well as histologic grade and type, and tumor stage.

2.2 Data cleaning

For boolean input features (the above or below regression line features), missing values were filled randomly with true and false values in such a way that the imputed data would have the same proportion of true to false values as the original data. Missing values in numerical features were filled with the mean of the feature, and all numerical features were normalized using the formula $(x - \min(x)) / (\max(x) - \min(x))$.

2.3 Scoring

We used three scoring metrics: accuracy, precision, and recall. Accuracy is simply the proportion of samples that were correctly scored. Precision and recall are based on the confusion matrix for the scoring results. For a single class, precision is calculated as $\frac{TP}{TP+FP}$, where TP is the true positive rate and FP is the false positive rate. It tells us what proportion of samples assigned the given class were actually in that class. Recall for a single class is calculated as $\frac{TP}{TP+FN}$, where TP is the true positive rate and FN is the false negative rate. This tells us what proportion of all members of the given class were successfully detected as members of that class. Thus, if we have high precision we know that anything assigned the given class is probably actually in that class; if we have high recall, we know that anything that is actually in the given class was probably assigned to that class.

In our data, some targets have more than two output classes. In these cases, precision and recall are calculated separately for each output class, and then averaged across all output classes.

3 Results

3.1 Baseline accuracy

To establish a baseline accuracy for the targets, we trained the SciKit Learn multi-layer perceptron (MLP) using the default hyperparameters, except reducing the hidden layers to one layer of 16 nodes, supposing that this simple model would give us a good idea of the complexity of the problem. The cancer type input feature, as well as all targets, were one-hot encoded. Testing with 10-fold cross-validation yielded these baseline results for each target:

| Target | Accuracy | Precision | Recall |
|------------------------|----------|-----------|--------|
| histologic_grade | 0.3309 | 0.3371 | 0.3434 |
| histologic_type | 0.1949 | 0.1949 | 0.1949 |
| recurrence_status | 0.7702 | 0.7733 | 0.7764 |
| success_last_follow-up | 0.2570 | 0.2585 | 0.2600 |
| survival_status | 0.8067 | 0.8113 | 0.8159 |
| tumor_stage | 0.3578 | 0.3761 | 0.3945 |

Table 1: Baseline scores for each target.

3.2 Multi-layer perceptron

After establishing the baseline accuracy using an MLP, we initially tried to optimize all hyperparameters at once using the SciKit Learn randomized cross-validation search.

This method draws random combinations of hyperparameters from user-provided distributions and tests them, returning the best combinations it finds. This did lead to improvements in all metrics for the "histologic_grade" and "success_last_follow-up" targets, and a marginal improvement in the recall for "recurrence_status", as shown below:

| Target | Accuracy | Precision | Recall |
|------------------------|---------------|---------------|---------------|
| histologic_grade | 0.3669 | 0.3822 | 0.4012 |
| histologic_type | 0.1411 | 0.1441 | 0.1599 |
| recurrence_status | 0.7653 | 0.7711 | 0.7769 |
| success_last_follow-up | 0.3085 | 0.3208 | 0.3473 |
| survival_status | 0.7990 | 0.8005 | 0.8021 |
| tumor_stage | 0.2059 | 0.2170 | 0.2282 |

Table 2: MLP scores for each target after initial random hyperparameter search.

However, for the rest of the targets this random search actually led to decreased scores. We believe this is because it takes too many iterations to happen upon good parameter combinations if all parameters were being randomly selected at once. So, we instead decided to only use the random search to optimize one hyperparameter at a time, holding all others constant. Using this method, we obtained the following results for each target. This led to significant score improvements for "histologic_grade", "histologic_type", "success_last_follow-up", and "tumor_stage".

| Target | Accuracy | Precision | Recall |
|------------------------|-----------------|-----------------|-----------------|
| histologic_grade | 0.521496 | 0.521496 | 0.521496 |
| histologic_type | 0.593371 | 0.600994 | 0.608617 |
| recurrence_status | 0.7702 | 0.7733 | 0.7764 |
| success_last_follow-up | 0.517803 | 0.517803 | 0.517803 |
| survival_status | 0.8067 | 0.8113 | 0.8159 |
| tumor_stage | 0.366288 | 0.409233 | 0.452178 |

Table 3: MLP scores for each target after initial random hyperparameter search.

3.3 k-nearest neighbors

The first parameter that was optimized was k. We found that accuracy ceased to improve once $k > 20$. As with MLP, we tested a combination of hyperparameters. Due to the speed of KNN we were able to test a wide range of parameters. We were able to test every possible input for weights (the weight function used in prediction), algorithm (the algorithm used to compute the nearest neighbors), and p (the parameter for the Minkowski metric). n_neighbors and leaf_size tested a range of numbers, 10-100 and 2-50 respectively. The optimal scores for each target are reported in the table below.

| Target | Accuracy | Precision | Recall |
|------------------------|----------|-----------|--------|
| histologic_grade | 0.8126 | 0.8126 | 0.8126 |
| histologic_type | 0.8250 | 0.8250 | 0.8250 |
| recurrence_status | 0.3882 | 0.3882 | 0.3882 |
| success_last_follow-up | 0.0164 | 0.0164 | 0.0164 |
| survival_status | 0.3066 | 0.3066 | 0.3066 |
| tumor_stage | 0.1043 | 0.1043 | 0.1043 |

Table 4: KNN scores for each target.

3.4 Naïve Bayes

SciKit Learn does not currently have a Naïve Bayes classifier that takes both continuous data and nominal data. We separated the continuous data from the nominal and then ran SciKit’s GaussianNB classifier on the continuous data and SciKit’s MultinomialNB on the nominal data. Again, we used 10-fold cross validation and took the average accuracy, precision, and recall scores. The MultinomialNB classifier scored similarly to the MLP, however the GaussianNB classifier performed poorly across all labels with accuracies ranging between 10 and 50 percent.

| Target | Accuracy | Precision | Recall |
|------------------------|----------|-----------|--------|
| histologic_grade | 0.3499 | 0.3499 | 0.3499 |
| histologic_type | 0.4776 | 0.4776 | 0.4776 |
| recurrence_status | 0.2570 | 0.2570 | 0.2570 |
| success_last_follow-up | 0.1435 | 0.1435 | 0.1435 |
| survival_status | 0.4482 | 0.4482 | 0.4482 |
| tumor_stage | 0.3997 | 0.3997 | 0.3997 |

Table 5: Scores for GaussianNB on continuous data.

| Target | Accuracy | Precision | Recall |
|------------------------|----------|-----------|--------|
| histologic_grade | 0.4512 | 0.4512 | 0.4512 |
| histologic_type | 0.5581 | 0.5581 | 0.5581 |
| recurrence_status | 0.8098 | 0.8098 | 0.8098 |
| success_last_follow-up | 0.5121 | 0.5121 | 0.5121 |
| survival_status | 0.8253 | 0.8253 | 0.8253 |
| tumor_stage | 0.5271 | 0.5271 | 0.5271 |

Table 6: Scores for MultinomialNB on nominal data.

4 Discussion

Our dataset is not large enough to successfully learn this problem. For example, the best accuracies we achieved were around 80% for the recurrence and survival status features; however, both features have about 80% of patients in one class and 20% in the other, so the models likely achieved this accuracy simply by guessing the majority class every time, and not really paying attention to any differences between samples. If we obtained a larger dataset, that would enable our models to learn more deeply about these problems.

Additionally, the performance differences of the two different Naïve Bayes Classifiers suggests that the nominal features are more indicative of outcomes than the continuous data.

References

- [Han *et al.*, 2021] Yu Han, Lauren Z. Li, Nikhitha L. Kas-tury, Cody T. Thomas, Maggie P. Y. Lam, and Edward Lau. Transcriptome features of striated muscle aging and predictability of protein level changes. *Molecular Omics*, 17(5):796–808, October 2021. Publisher: The Royal Society of Chemistry.
- [Lindgren *et al.*, 2021] Caleb M. Lindgren, David W. Adams, Benjamin Kimball, Hannah Boekweg, Sadie Tayler, Samuel L. Pugh, and Samuel H. Payne. Simplified and Unified Access to Cancer Proteogenomic Data. *Journal of Proteome Research*, 20(4):1902–1910, April 2021. Publisher: American Chemical Society.
- [Payne, 2015] Samuel H. Payne. The utility of protein and mRNA correlation. *Trends in Biochemical Sciences*, 40(1):1–3, January 2015.
- [Rodriguez *et al.*, 2021] Henry Rodriguez, Jean Claude Zenklusen, Louis M. Staudt, James H. Doroshov, and Douglas R. Lowy. The next horizon in precision oncology: Proteogenomics to inform cancer diagnosis and treatment. *Cell*, 184(7):1661–1670, April 2021.
- [Waldbauer *et al.*, 2012] Jacob R. Waldbauer, Sébastien Rodrigue, Maureen L. Coleman, and Sallie W. Chisholm. Transcriptome and Proteome Dynamics of a Light-Dark Synchronized Bacterial Cell Cycle. *PLOS ONE*, 7(8):e43432, August 2012. Publisher: Public Library of Science.