Protein Covariance: Predicting Phenotypes Based On Amino Acid Sequences

Lara Gokcelioglu

Insitute For Computing In Research Santa Fe, NM, United States Of America

July 28th 2024

Abstract

This paper analyzes the covariance of proteins following a research that is a mix of biology and computing. It aims to predict phenotypes ”em_max”, ”ex_max” and ”states_0_brightness” using amino acid sequences in two representations ”pc_coords” and ”proteins_projected_pc_coords. For this purpose the research trains models using K-Fold Cross Validation and LASSO. The goal of the project is to better understand functions and phenotypes of proteins on a genetic level; hence, the research uses the amino acid sequences to make these predictions. This document aims to analyze the relation between certain amino acids and the proteins' phenotypes. The analysis is conducted through the comparison between two representations' R^2 value for each phenotype, linearity of the relationship between the test set and predictions, probability plots and residual distributions. This research concluded that in order to make such predictions more parameters should be observed and taken into account.

1 Introduction

Covariance is defined as a measure of how much two variables change together[1]. In the context of this research the two variables are the amino acid sequences and the three phenotypes observed. The three phenotypes ”em_max”, ”ex_max” and ”states_0_brightness” are observable characteristics or traits of the proteins based on the outcome of their amino acid sequences. ”em_max” stands for the emission maximum of the state in nanometers, while ”ex_max” stands for the excitation maximum of the state in nanometers. ”states_0_brightness” on the other hand, represents molecule brightness and is calculated as the product of extinction coefficient and quantum yield. However the brightness of the protein relies on additional characteristics such as folding and maturation efficiency and pKa. The research cultivates models to predict these phenotypes based on two representations ”pc_coords” and ”proteins_projected_pc_coords. ”pc_coords” bases its data solely in flourescent proteins while ”proteins_projected_pc_coords” bases its data on various kinds of proeins. The purpose of using these different representations is to evaluate the effects of the source of the proteins on the phenotypes. The comparison of two distinct representations will answer the questions regarding the efficiency of the extra information on proteins in explaining the functions and phenotypes of proteins compared to the efficiency of working with a specific and contained group of proteins.

2 Background

Proteins–the main focus of this research–are large, complex molecules that play many critical roles in the body. They do most of the work in cells and are required for the structure, function, and regulation of the body's tissues and organs[2]. They consist of hundreds of building blocks chained to each other called amino acids. These chains, also called amino acid sequences, contain many of 20 different amino acids in nature, and their unique sequencing assign the proteins their shapes and functions. Another function of proteins is determining which genes will be expressed. The expression of genes called genotype affect how an organism looks leading to the formation of its phenotype. However, predicting the phenotypes of known sequences is no easy task. The foremost method to predict he phenotypes is looking at protein representations. Protein representations are vector representation, called contextualized embeddings, of the protein sequence that captures its important features. These representations are used as input to computational learning models to predict the phenotypes as is done on this research with the phenotypes ”em_max”, ”ex_max” and ”states_0_brightness” in representations of ”pc_coords” and ”proteins_projected_pc_coords.

In order to get the aforementioned representations from vectors, certain transformations have to be applied to the amino acid sequences. As most datasets are a split between categorical and numerical data, a method to transform that categorical data is required considering most machine learning models only work with numerical data. A popular method is one hot encoding which is a technique used to represent categorical variables as numerical values in a machine learning models. One hot encoding transforms categorical data by assigning two distinct features 1's and 0's, such as 1 for carbon and 0 for hydrogen. Hence the categorical data gets transformed into numerical values. Another efficient tool to transform data in machine learning is singular value decomposition which essentially reduces dimensions by dividing a certain matrix into smaller matrices with minimal data loss. The reduction in dimensions provides an easier pathway to data analysis and to train a machine learning model. When creating a machine learning model, you define the answer that you would like to capture and set parameters for the model to work within and learn from[4]. The model trained for the sake of this research predicts the phenotypes using the aforementioned representations using K-Fold cross validation for training the linear regression model and LASSO for predicting the output that is the phenotypes based on the previously provided representations.

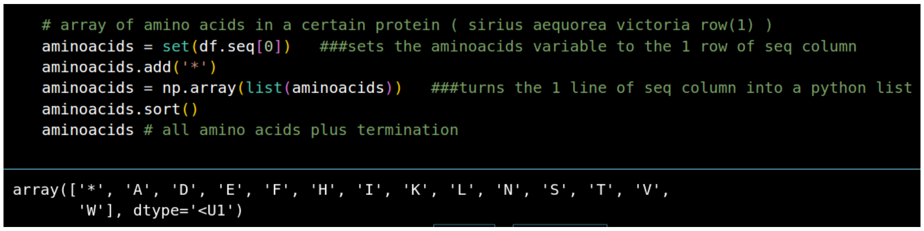
"Cross validation is a technique used in machine learning to evaluate the performance of a model on unseen data. It involves dividing the available data into multiple folds or subsets, using one of these folds as a validation set, and training the model on the remaining folds. This process is repeated multiple times, each time using a different fold as the validation set. Finally, the results from each validation step are averaged to produce a more robust estimate of the model's performance. Cross validation is an important step in the machine learning process and helps to ensure that the model selected for deployment is robust and generalizes well to new data.[5]"

K-Fold cross validation divides the dataset into k folds and uses k-1 models as the test split while keeping one as the test set. The process is repeated k times, hence every fold gets used as test data. K-Fold method's nature of every fold being used as a a test split produces more accurate results compared to other methods as random errors from the dataset are mostly avoided. Additionally, Lasso which stands for Least Absolute Shrinkage and Selection Operator is frequently used in machine learning to handle high dimensional data as it facilitates automatic feature selection with its application[6] and is used for predictions in this research's linear model.

In this research the coefficient of determination also called the R^2 value is used as numerical data to determine the fitness of the model's prediction. To give it a better definition in the context of regression, it is a statistical measure of how well the regression line approximates the actual data[7]. The R^2 value is always a number between 0 and 1. Generally, the closer to 1 the value the better the predictive capacity of the model, consequently a lower value would indicate a less accurate predictive model. However, this is not always the case due to the data transformations, different units or the nature of the variables. Another indicator of accuracy is residuals which stands for the difference between the actual value and the predicted value. An accurate linear model aims to have the least value of residuals. Other analytical indicators of a good predictive model are mean squared error, which stands for the average squared values between observed and predicted values, mean absolute error which measures the average of the absolute distance between the observed and predicted value and root mean squared error which measures the average difference between a statistical model's predicted values and the actual values.

3 Method

The data used is imported from a csv file and read using the pandas library. Then under the aminoacids variable the amino acids in the sequences are determined and turned into an array under the leftjustified_seqs variable. The leftjustified_seqs variable uses the "ljust" function to set the amino acid sequnces to a fixed 512 positions. The goal is to determine to what capacity these phenotypes can be predicted by the R^2 predictive metric using a linear model of the above representations. The two representations are distinct as pc_coords

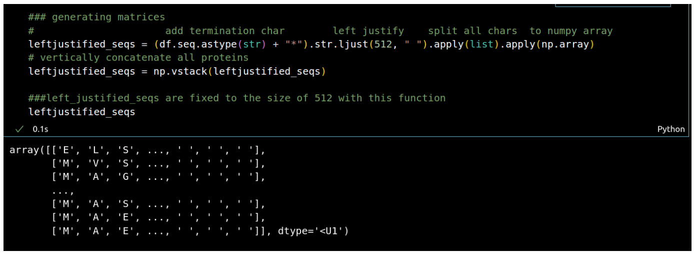


```
# array of amino acids in a certain protein ( sirius aequorea victoria row(1) )
aminoacids = set(df.seq[0])   ###sets the aminoacids variable to the 1 row of seq column
aminoacids.add('*')
aminoacids = np.array(list(aminoacids))   ###turns the 1 line of seq column into a python list
aminoacids.sort()
aminoacids # all amino acids plus termination
```

```
array(['*', 'A', 'D', 'E', 'F', 'H', 'I', 'K', 'L', 'N', 'S', 'T', 'V',
       'W'], dtype='<U1')
```

gets its data solely from fluorescent proteins, and proteins_projected_pc_coords looks at all kind of proteins from different organisms.

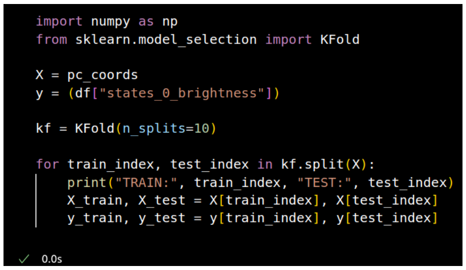
Under the leftjustified_seqs is now stored 227 proteins with 512 amino acid positions each. Following the variables creation, they are interpreted in match_aminoacids function. This function turns an array of 1s and 0s, one hot encodes, depending on the input protein sequences matching the 21 amino acids stored under "aminoacids" variable and this matrix is stored under the protein_ohe variable. As another form of data transformation singular value decomposition is performed on the protein_ohe variable in order to decrease dimensions of the matrix and stored under the final representation pc_coords. Now that all variables are assigned, the model can be trained. The model is trained using the K-fold cross validation method. The x value is assigned as pc_coords, while the y value is the "states_0_brightness" from the initial data frame. The data is split into 10 folds, and as it is in nature of K-fold cross validation all 10 folds are used to train and test which is why the method produces accurate results. It uses all data points rather than assigning a random 10% for the test set.



```
### generating matrices
#                     add termination char        left justify    split all chars  to numpy array
leftjustified_seqs = (df.seq.astype(str) + "*").str.ljust(512, " ").apply(list).apply(np.array)
# vertically concatenate all proteins
leftjustified_seqs = np.vstack(leftjustified_seqs)

###left_justified_seqs are fixed to the size of 512 with this function
leftjustified_seqs
```

```
array([['E', 'L', 'S', ..., ' ', ' ', ' '],
       ['M', 'V', 'S', ..., ' ', ' ', ' '],
       ['M', 'A', 'G', ..., ' ', ' ', ' '],
       ...,
       ['M', 'A', 'S', ..., ' ', ' ', ' '],
       ['M', 'A', 'E', ..., ' ', ' ', ' '],
       ['M', 'A', 'E', ..., ' ', ' ', ' ']], dtype='<U1')
```



```
import numpy as np
from sklearn.model_selection import KFold

X = pc_coords
y = (df["states_0_brightness"])

kf = KFold(n_splits=10)

for train_index, test_index in kf.split(X):
    print("TRAIN:", train_index, "TEST:", test_index)
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]
```

A very similar process is repeated for proteins_projected_pc_coords. However, proteins_projected_pc_coords gets its data from about 60 million different proteins in contrast to solely fluorescent proteins. Therefore, the dataframe gets divided into 2 matrices stored under the variables "evals" and "V_512". Then it is interpreted similarly to the "pc_coords".

4 Results And Discussion

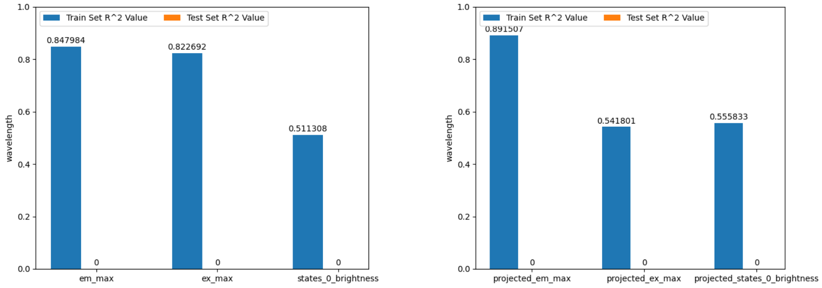
Phenotype	Representation	Train Set R^2 Value	Test Set R^2 Value
em_max	pc_coords	0.848	-37.9
ex_max	pc_coords	0.823	-26.9
states_0_brightness	pc_coords	0.511	-9.84
em_max	proteins_projected...	0.891	-34.4
ex_max	proteins_projected...	0.549	-26.9
sctates_0_brightness	proteins_projected...	0.556	-10.8

For this research 6 linear models were trained, and the tables above show that the models were overfitted–which occurs when an algorithm fits too closely or even exactly to its training data, resulting in a model that can't make accurate predictions or conclusions from any data other than the training data [8]. This explains the sensible R^2 values obtained for the train set and the below 0 values, which represent inaccuracy, for the test set. The data is collected in two representations for each phenotype in order to observe the effects of distinct proteins on the predictions.

The reason of overfitting might be interpreted as too many possible phenotypes but not enough data points. This causes a lack of generalization in the information obtained as it only fits the train set and not the test set. In case of the addition of more phenotypes the coefficients in the linear regression can be adjusted in a certain way to predict the phenotyes accurately.

5 Conclusion

After evaluating 6 models for 3 phenotypes "em_max", "ex_max" and "states_0_brightness" in terms of emission and excitation wavelengths and brightness features in two different representations pc_coords and proteins_projected_pc_coords one gathering data from solely fluorescent and the other from various proteins, the data for analysis was collected in R^2 values for both test and train sets. The analysis of the data showed largely overfitting on the train data and did not make accurate predictions on the test data. In order to overcome this, more phenotype variables have to be incorporated and assigned multiple coefficients for the fitness of the linear regression model. The variables will have to be regulated and observed to end up with the most structured level of effect on the phenotypes' prediction.



(a)

(b) label 2

Figure 1: 2 Figures side by side

6 Future Work

- Looking at more fold combinations

This research's results look at only one combination of the train and test folds. However, there were 9 more combinations generated. A next step would be looking at the other combinations and comparing the results.

- Moving beyond linear models

The analysis for this research paper looks at only linear relations between the variables as the predictive machine learning model is trained as a linear regression model. This limits the explorative nature of research as other models might fit the data better.

- Including more parameters and hyperparameters

The analysis overlooks many parameters and hyperparameters such as learning rate of the model, pooling size and the dropout probability in the neural network.

7 Acknowledgements

I would like to express my gratitude to my mentor Benjamin Doran for guiding me through this research process. I would also like to extend my thanks to the Institute For Computing In Research as well as Rhonda Crespo and Mark Galassi.

8 References

[1]M. Weigt, H. Szurmant, in Brenner's Encyclopedia of Genetics (Second Edition), 2013

"NCI Dictionary of Genetics Terms." Comprehensive Cancer Information - NCI, www.cancer.gov/publications/dictionaries/genetics-dictionary/def/phenotypeComputer

Memory Upgrade. Retrieved from www.computermemoryupgrade.net/. Accessed 29 July 2024.

“FPbase Help.” Glossary of Terms, help.fpbase.org/glossary#em-max. Accessed 29 July 2024.

[2]“What Are Proteins and What Do They Do?: Medlineplus Genetics.” MedlinePlus, U.S. National Library of Medicine, medlineplus.gov/genetics/understanding/howgeneswork/prote Accessed 29 July 2024.

“14.2: Biochemistry of Gene Behavior.” Biology LibreTexts, Libretexts, 20 June 2023, bio.libretexts.org/Bookshelves/Genetics/Online_Open_Genetics_(Nickle_and_Barrette-Ng)/14:_Appendices/14.02:_Biochemistry_of_GeneBehavior.

Bai, Liam. “How to Represent a Protein Sequence.” Liams Blog RSS, 29 Sept. 2023, liambai.com/protein-representation/.

[3]Ganji, Lekhana. “One Hot Encoding in Machine Learning.” GeeksforGeeks, 21 Mar. 2024, www.geeksforgeeks.org/ml-one-hot-encoding/.

[4]“What Is an ML Model?” HPE, Hewlett Packard Enterprise Development LP , www.hpe.com/us/en/what-is/ml-models.html. Accessed 29 July 2024.

[5]Sharma, Abhishek. “Cross Validation in Machine Learning.” GeeksforGeeks, 21 Dec. 2023, www.geeksforgeeks.org/cross-validation-machine-learning/.

[6]“What Is Lasso Regression?” IBM, 16 Jan. 2024, www.ibm.com/topics/lasso-regression.

[7] “Coefficient of Determination, R-Squared.” Numeracy, Maths and Statistics - Academic Skills Kit, www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/coefficient-of-determination-r-squared.html. Accessed 29 July 2024.

Taylor, Sebastian. “R-Squared.” Corporate Finance Institute, 22 Nov. 2023, corporatefinanceinstitute.com/resources/data-science/r-squared/.

“Residuals.” Numeracy, Maths and Statistics - Academic Skills Kit, www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/residuals.html. Accessed 29 July 2024.

Qian Jiang, Xin Jin, Shin-Jye Lee, Shaowen Yao, Protein secondary structure prediction: A survey of the state of the art, Journal of Molecular Graphics and Modelling, Volume 76, 2017, Pages 379-402, ISSN 1093-3263, https://doi.org/10.1016/j.jmgm.2017.07.015, (https://www.sciencedirect.com/science/article/pii/S109332

Frost, Jim. “Mean Squared Error (MSE).” Statistics By Jim, 28 May 2023, statisticsbyjim.com/regression/mean-squared-error-mse/.

Frost, Jim. “Root Mean Square Error (RMSE).” Statistics By Jim, 28 May 2023, statisticsbyjim.com/regression/root-mean-square-error-rmse/.

Nyuytiymbiy, Kizito. “Parameters and Hyperparameters in Machine Learning and Deep Learning.” Medium, Towards Data Science, 28 Mar. 2022, towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac.

[8]“What Is Overfitting?” IBM, 15 Oct. 2021, www.ibm.com/topics/overfitting.