Solid State Nanopore Data Classification with Singular Value Decomposition

Chen Chen, MinGyu Kim

December 8, 2022

1 ABSTRACT

In this project, we utilize an unsupervised machine learning technique, singular value decomposition (SVD) to classify five classes biomolecular data from solid state nanopore (SSN). Our model can achieve 100% accuracy when the K value (i.e., the rank) is appropriately selected. Then, we investigate which features are important based on the analyses of eigenvector and residual norm error. The important features we found show distinct distributions, implying that they contribute to the decision-making behind the model.

2 NOMENCLATURE

U = eigenvector matrix
Σ = singular value matrix
V = complex unitary matrix
X = training data
u = eigenvector
w = weight vector
W = weight vector matrix
z = test data
Z = test dataset
L = loss function
$\hat{Y}$ = estimated regression model
Y = true model
K = number of eigenvectors/rank

3 INTRODUCTION

Our goal through this project is to classify five different bio-structures (DNA plasmid PUC19, dsDNA 1kb ladder, ssRNA ladder, dsRNA ladder, and E. coil ribosome) using the singular value decomposition (SVD) method and investigate which features play a role in the classification. Previous works have used machine learning model as the classifiers. For instance, *Xia et al., 2021* [7] converts the scatter plots (dwell time vs. current blockage ratio) into feature vectors by a deep neural network model, and then uses principal component analysis (PCA) to do the classification. They also try other machine learning models, e.g., support vector machine (SVM), random forest, k-nearest neighbors (kNN), etc. Although machine learning techniques have been applied in nanopore sensing as a classifier with a good performance (e.g., as summarized by *Arima et al., 2021* [1]), we are not sure about what drives the model make the decision as some models work as a black-box or there is no detailed analysis of it. Therefore, we will use the SVD as the classifier, as it is an unsupervised model and could tell us the story behind the decision-making.

4 METHODS

4.1 Data

For this project, we use five classes biomolecular data from solid state nanopore, which are DNA plasmid puc19 (2284 events), dsDNA 1kb ladder (919 events), ssRNA ladder (1874 events), dsRNA ladder (1398 events), and ecoli ribosome (17077 events). Initially, there are 22 features in total and the distribution of these featues are shown in Figure 7. We select 14 features after removing not informative features. The selected features are *pore_diam_nm*, *a1_pA*, *s1_pA*, *s3_pA*, *pot_sec*, *dwell_sec*, *t12_sec*, *mean_amp_pA*, *mean_amp_nS*, *max_amp_pA*, *max_amp_nS*, *med_amp_pA*, *std_amp_pA*, and *area_pA_sec*.

Additionally, in order to have the high quality events, we select the events with an SNR > 5. Finally, we have 2282 events for DNA plasmid puc19, 907 events for dsDNA 1kb ladder, 1183 events for ssRNA ladder, 1000 events for dsRNA ladder, and 17020 events for ecoli ribosome. The distributions of selected features (high quality events with an SNR > 5) are shown in Figure 1. The x axis represents the class, which is also color-coded as shown by the legend. We can see that different classes have different magnitudes and ranges of these selected features. Additionally, we do not take buffer events into account in this project.

After we have selected features and events, the data is split into training (80%) and testing (20%) set. We will apply SVD on training set for different classes separately. The testing data will be classified based on the residual norm with respect to different training data after SVD. The class of the training set, which has the minimum residual norm with the testing data, will be assigned as the prediction label. More details are introduced in the following sections and our procedure of classification is based on [3].

4.2 Singular Value Decomposition (SVD)

Each class has a matrix X made by the training data, whose rows are features and columns are events. Then by following equation 1, we conduct SVD for each class separately, where U is a unitary matrix or eigenvectors that represent features, Σ is singular values that are sorted in the order of their own magnitudes, V is a complex unitary matrix, m is the number of features, and n is the number of events.

$$X = U_{m\times m} * \Sigma_{m\times n} * V^T_{n\times n} \tag{1}$$

The following shows the python code of SVD, we are going to use NumPy [2] or SciPy [6] packages to conduct SVD.

```
# python code :
u, sigma, v = numpy.linalg.svd(data sample X)
```

4.3 Residual Norm Error

In order to do the classification, we calculate the residual norm error of the input vector with respect to different classes, and classify the input into the class with the smallest residual norm error. The residual norm error calculates the 'distance' between the data vector and the feature space of a specific class. The equation is shown below.

$$L = min_{w_i}||z - \sum_{i=1}^{k} w_i * u_i|| = min_W||Z - U_k W|| \tag{2}$$

where z is a test data or input data, w is a weight vector, and U is the unitary matrix of the training data X.

4.4 Least Square Regression Optimization

In this section, we introduce the derivation of equation 2 in details. To obtain the estimated weight, we optimize a linear regression model that replaces the estimated term $U_k W$. In a linear regression model, we can express that

$$\bar{Y} = \sum_{i}^{m} w_i * u_i = W * U_k$$

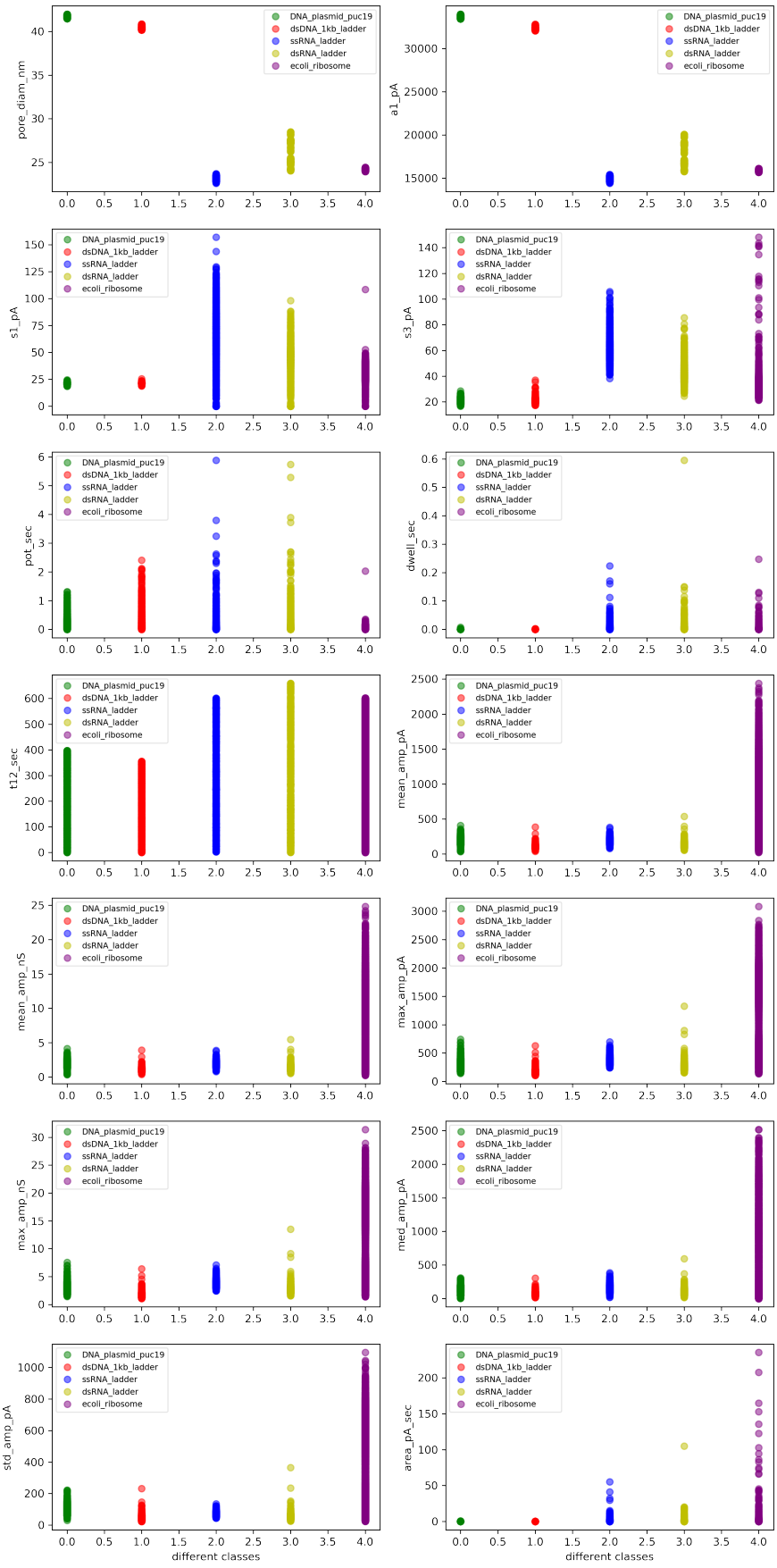

Figure 1: Distributions of selected features. The x axis shows different classes, which also have corresponding colors as shown in legend.

where w is weight, u is the unitary matrix, and $\bar{Y}$ is the estimated feature vector.
Least square error is

$$L = ||Y - \bar{Y}||^2 = ||Y - U_k W||^2 = (Y - U_k W)^T (Y - U_k W)$$

In order to get the minimum error, the gradient of the error function must be zero (local minima) [5].

$$\frac{\partial L(W,U)}{\partial W} = -2U_k^T Y + 2U_k^T U_k W \overset{SET}{=} 0 \tag{3}$$

$$\hat{W} = (U_k^T U_k)^{-1} U_k^T Y \tag{4}$$

Using Equation (2),

$$L = ||Y - U_k \hat{W}|| = ||Y - U_k * (U_k^T U_k)^{-1} U_k^T Y|| \tag{5}$$

$U_k^T U_k$ is an orthogonal matrix, so this term can be regarded as an identity matrix. Therefore, the error term L from Equation (2) can be reduced as

$$L = ||Y - U_k U_k^T Y|| \overset{replace Y with Z}{=} ||Z - U_k U_k^T Z|| = ||(I - U_k U_k^T) Z|| \tag{6}$$

5 Results

5.1 Classification results with different K

Figure 2 shows the classification result with respect to different K values. The x axis is K value, which means we choose the first K_{th} important eigenvectors to calculate the residual norm. The y axis shows the accuracy of the testing data. We can observe that accuracy varies with K value. Table 5.1 lists the value of accuracy with respect to different K values. When K is 12, the accuracy reaches 1.0, which implies that the SVD successfully classifies our data.

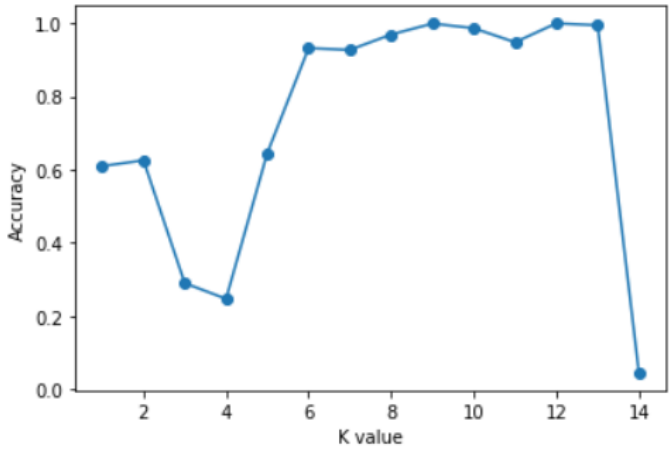

Figure 2: Classification result with different K values.

K value	Accuracy
1	0.6096
2	0.6254
3	0.2897
4	0.2469
5	0.6458
6	0.9321
7	0.9275
8	0.9694
9	0.9993
10	0.9866
11	0.9482
12	1.0
13	0.9946
14	0.0446

Table 1: Accuracy with respect to K value

5.2 Explainable SVD

Given our SVD model has a high accuracy, we would like to investigate which features play a role in the classification. In the following, we find the important features based on the analysis of eigenvectors and residual norm error.

5.2.1 Analysis based on the eigenvectors

Figure 3 shows the U matrix (eigenvectors) and singular values for the training data of DNA plasmid puc19. For the U matrix (left panel), each column represents an eigenvector. The first eigenvector (from left) is the most important one, and the importance decreases from left to right. This importance is related to the corresponding singular values which are shown in the right panel of figure 3. The first singular value corresponds to the first eigenvector and its magnitude is 10^6, which is much larger than other singular values (ranging from $\sim 10^{-4}$ to $\sim 10^4$). The U matrix and singular values for other classes of data are shown in figure 8 (Appendix).

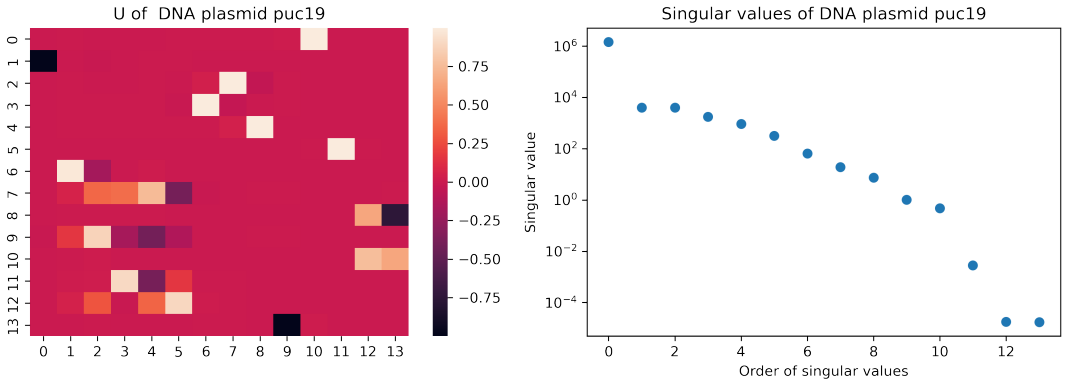

Figure 3: U matrix (left) and singular values (right) of DNA plasmid puc19 training set data.

In order to find the features that are important to the classification, we check the first three important eigenvectors for each class. For each eigenvector, we investigate which element has the largest absolute value and the corresponding feature could play a role in the classification. The results are shown in the table 5.2.1. For most classes except E. coli ribosome, the important features

are a1_pA, t12_sec, and max_amp_nS. For E. coli ribosome, these features are a1_pA, max_amp_nS, and med_amp_pA.

Biomolecular class	the number of component that has the largest absolute value for the first three important eigenvectors and the corresponding features
DNA plasmid puc19	2nd, 7th, 10th (a1_pA, t12_sec, max_amp_nS)
dsDNA 1kb ladder	2nd, 7th, 10th (a1_pA, t12_sec, max_amp_nS)
ssRNA ladder	2nd, 7th, 10th (a1_pA, t12_sec, max_amp_nS)
dsRNA ladder	2nd, 7th, 10th (a1_pA, t12_sec, max_amp_nS)
E. coli ribosome	2nd, 10th, 12th (a1_pA, max_amp_nS, med_amp_pA)

Table 2: Important features for each class based on the analysis of eigenvector.

Next, we plot these features in a three-dimension plot. In order to be consistent, we show features a1_pA, t12_sec, and max_amp_nS for all classes. The results are shown in figure 4, and different classes are color-coded. Due to the different magnitudes of features, we plot different classes in different plots in order to give a better visualization. It is clear that each class has its own distinct feature distribution without any obvious overlap, supporting that these features are important for classification. The main feature that makes the distribution distinct is a1_pA, which is selected from the first important eigenvector.

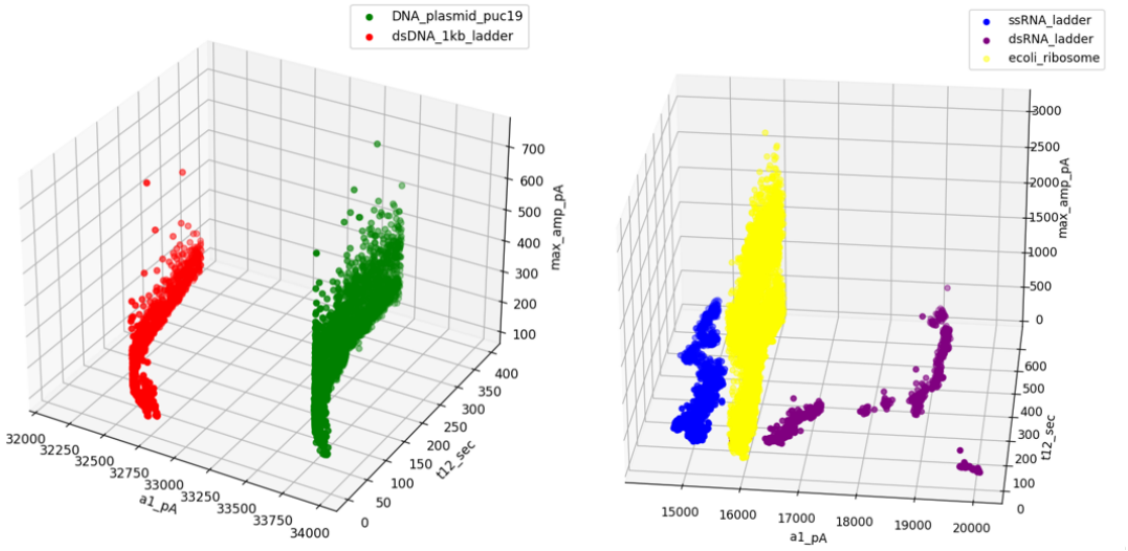

Figure 4: Distribution of three important features for the classification based on the analysis of eigenvector. Each axis represents a feature.

5.2.2 Analysis based on the residual norm error

Using Equation 6, the residual norm errors derived by all classes were calculated for every input data. The coefficients of the error functions that were used to estimate the error changed depending on the number of eigenvectors, K. When our model had the highest accuracy (k = 12), the corresponding residual norm error could provide information about the important features for the classification.

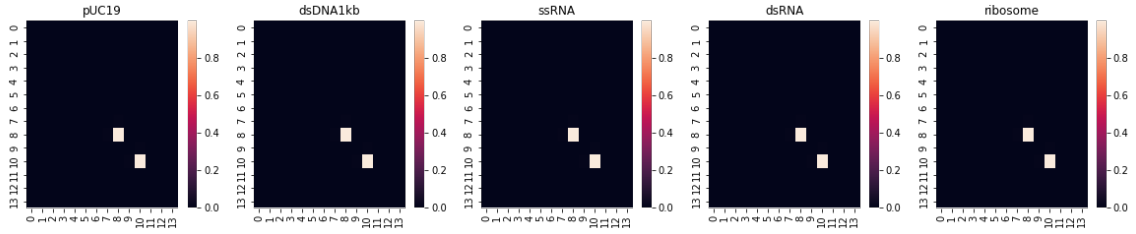

Figure 5: Residual norm error for each class

As shown in the figure 5, the error functions found the 2 main features that were regarded as critical coefficients in the regression model. The two features (white pixels) had relatively higher values than the other components of the vector, which mainly amplified the 9th and 11th feature values of the regression model. Those features were maximum conductance and mean conductance.

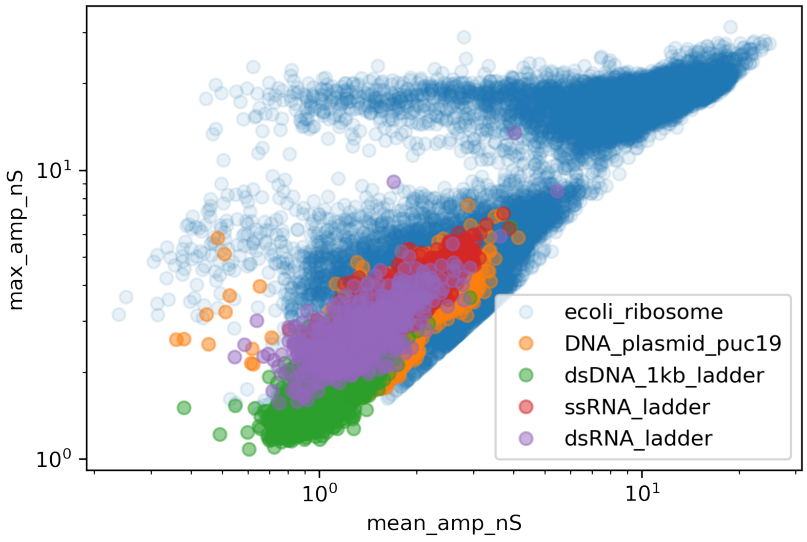

Figure 6: Distribution of important features, which are selected based on the residual norm.

The feature selection was made based on the result in Figure 5. The mean conductance and max conductance were expected to be the important features for the classification, and they were plotted in Figure 6. Despite the careful attention on the feature selection, the distributions were overlapped. As a further step, we might need to explore other features as to find critical variables that can present a more distinct clustering. In a higher dimensional space, additional features could contribute to discovering the distinct distribution (for the classification) in some ways.

6 DISCUSSION AND CONCLUSION

In this project, we successfully classify the biomolecule data using SVD method. The model is able to achieve 100% accuracy when K is appropriately selected. However, the accuracy could be due to the data leakage or not enough data. The data leakage means the data label information is leaking to the input of the model during the training [4]. In other words, the eigenvectors of the classes and error function optimized for each class might cause our model to perform perfectly on the given classes. When we do the detection via SSN, it is possible that we give the device a specific setting as we know which biomolecule is going to be detected. This specific setting could reflect on the feature distribution. Regarding the amount of data, it is possible that the data is not enough and these features coincidently show a distinct distribution without overlaps. Therefore, regarding these issues, the future steps are to check if there is any potential data leakage and to use a larger dataset,

Additionally, we find features, a1_pA, t12_sec, and max_amp_nS, play important roles in the classification based on the analysis of eigenvector. Investigating the residual norm error also revealed that max_amp_nS and mean_amp_nS could be important variables for the prediction. These findings help us to address how the model makes such decision. Moreover, it will be interesting to investigate the result of SVD with these features only and see if we could find more stories behind the decision-making. However, the decision-making behind the model could be more complicated than just considering a single feature from each eigenvector. We need to further investigate in more details about how features contribute to the classification.

Lastly, we didn't normalize our data before conducting SVD. Normalization could show us a different characteristic of eigenvectors and residual norm errors. Further analysis with normalization is needed to better understand the value of the feature for the SVD classifier.

DATA AND CODE

The extracted data and code can be accessed via the folder named *code* under T10 SML Solid State Nanopore Machine Learning on Teams (`https://gtvault.sharepoint.com/:f:/s/SLDFall20222-T10/ElOJox6ht1NIsO7GBYPUfVwBgzUujEyFTV2lJ9WxuByNfQ?e=nX9jKj`).

APPENDIX

Figure 7 shows the distribution of all features before feature and event selections. The x-axis represents different classes, which are also color-coded. The y-axis shows the values of the feature.

Figure 8 illustrates the U matrices and singular values of different classes. The first row is the class of dsDNA 1kb ladder. The second row is the class of ssRNA ladder. The third row is the class of dsRNA ladder. The last row is the class of E. coli ribosome.

Figure 7: Distribution of all features before feature and event selections.

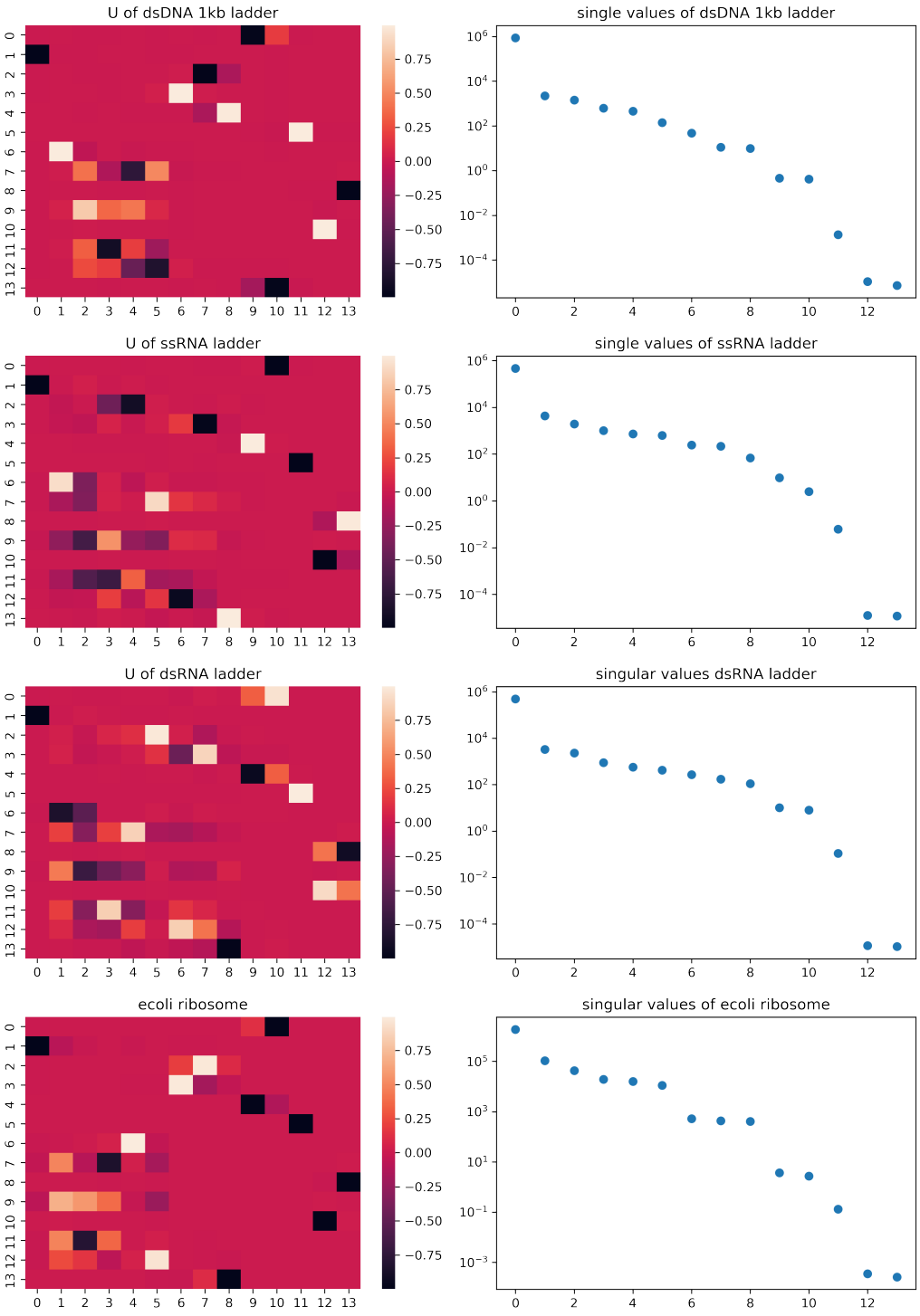

Figure 8: U matrices and singular values of different classes. The left column shows the U matrix and the right one represents the singular values. The color bar represents the magnitude of component in eigenvectors.

References

[1] Akihide Arima et al. "Solid-State Nanopore Platform Integrated with Machine Learning for Digital Diagnosis of Virus Infection". In: *Analytical Chemistry* 93.1 (2021), pp. 215–227. ISSN: 0003-2700. DOI: 10.1021/acs.analchem.0c04353.

[2] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[3] Nikos Kafritsas. *How to Use Singular Value Decomposition (SVD) for Image Classification in Python*. 2021. URL: https://towardsdatascience.com/how-to-use-singular-value-decomposition-svd-for-image-classification-in-python-20b1b2ac4990 (visited on 10/26/2021).

[4] Shon Mendelson. *Be Careful from Data Leakage – Potential Pitfalls in your Machine Learning Model*. 2022. URL: https://jfrog.com/community/data-science/be-careful-from-data-leakage-2/#:~:text=Data%5C%20leakage%5C%20refers%5C%20to%5C%20a,would%5C%20like%5C%20to%5C%20predict%5C%20on. (visited on 01/11/2022).

[5] Richard M. Murray. "Optimization-Based Control". In: 2b (2009), p. 18. URL: http://www.cds.caltech.edu/~murray/books/AM08/pdf/obc09-obc09_03Mar09.pdf.

[6] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[7] Ke Xia et al. "Synthetic heparan sulfate standards and machine learning facilitate the development of solid-state nanopore analysis". In: *Proceedings of the National Academy of Sciences* 118.11 (2021), e2022806118. DOI: 10.1073/pnas.2022806118. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.2022806118. URL: https://www.pnas.org/doi/abs/10.1073/pnas.2022806118.