Computational Science Project: Unsupervised and supervised analysis of protein sequences

Computational Science – Machine Learning for Physicists

Physics of Complex Systems – 2022/2023

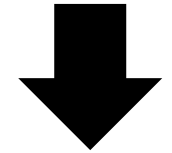
Davide Rossetti and Persia Kamali

30/01/2023

Task 1: One-hot encoding of protein sequence data

- Analysis of the file structure
- 20-dimensional representation for each amminoacid
- 96 features → 96x20 = 2920 features (binary variables)

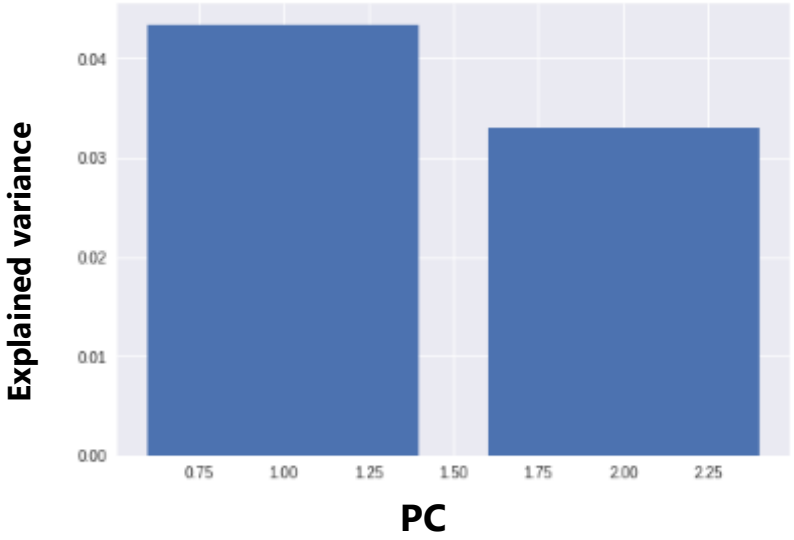
```
>sequence_1 functional_true
-----SLEELRKEIESIDREIVELIARRTYVAKTIAQIKRERGLPTTDESQEQRVMERAGSNAKQFD-VD
ANLVKAIFKLLIELNKEEQRENR---

>sequence_2 functional_false
---TERLNELRDQIDQVDKELLKLLAKRLSLVAEVGEVKSRHGLPIYAPEREASMLASRRTEAEKMG-IP
PDLIEDILRRIMRESYANENDHGFKT

>sequence_3 functional_true
--TDNPLLALREKISALDLKLLDLLAERRELALEVAQTKLKSHRPIRDKERERDLLNSLIAEGK-KRGLD
GHYITRLFQMIIEDSVLTQQALLQKH
```

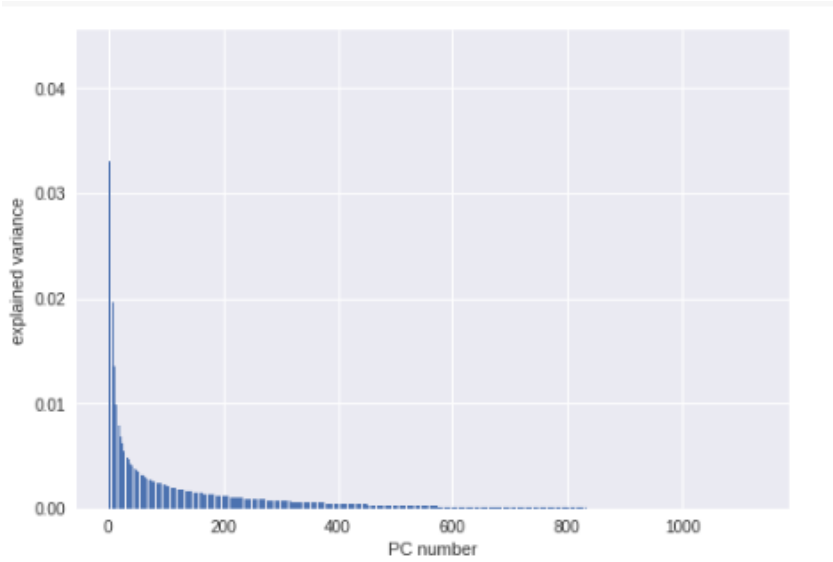


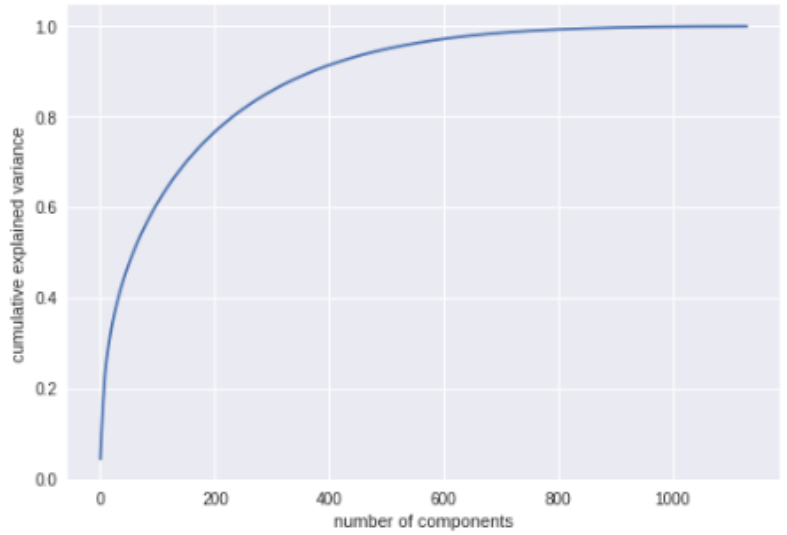
```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

Task 2: Dimensional reduction and visualization of sequence space

- PCA on the dataset of **natural sequences**:
  - 2D projection loses a lot of information
  - We would need at least 370 components to retain 90% of the variance.
  - With only two components we retained about 7.6% of the variance.



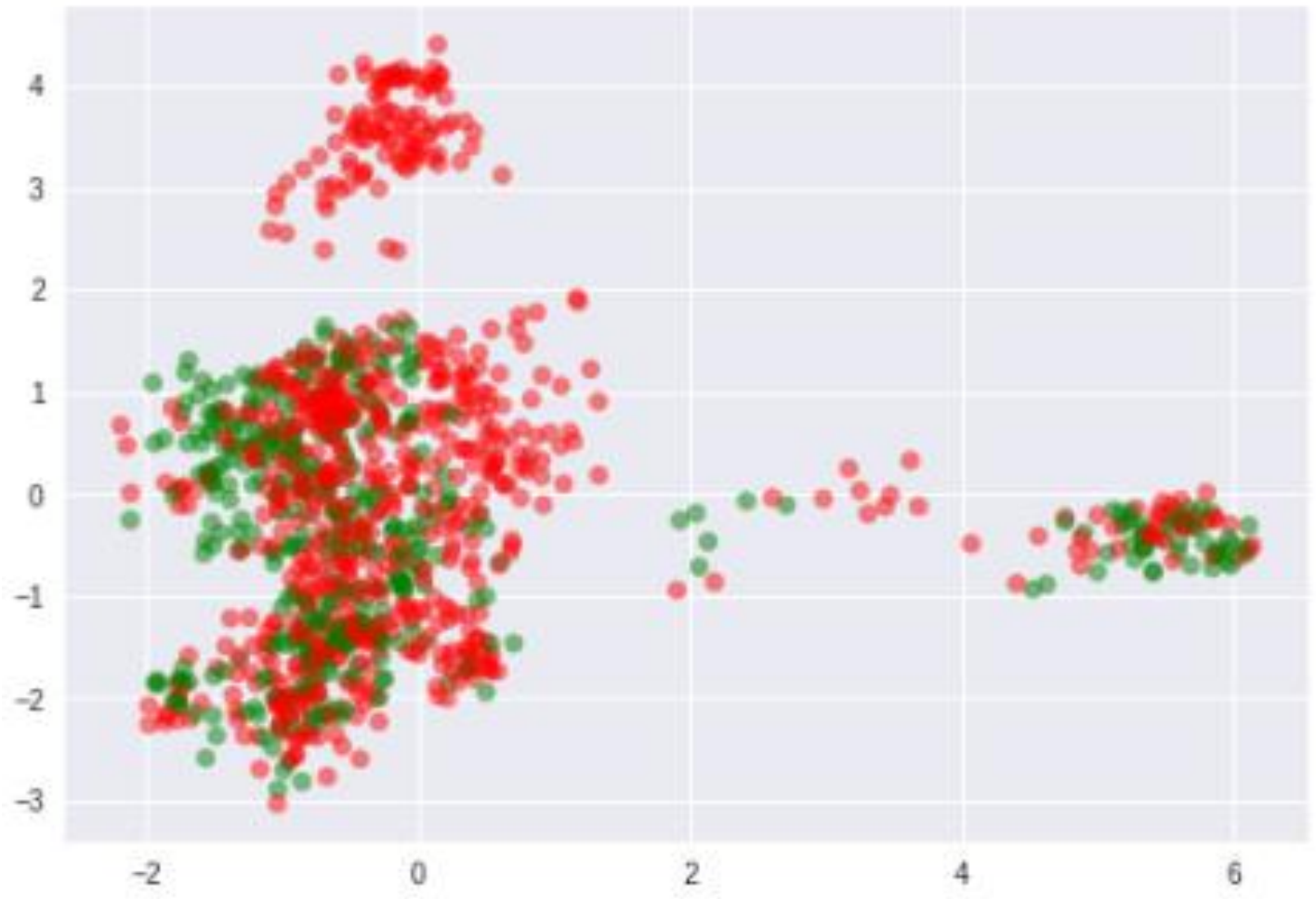




Task 2: Dimensional reduction and visualization of sequence space

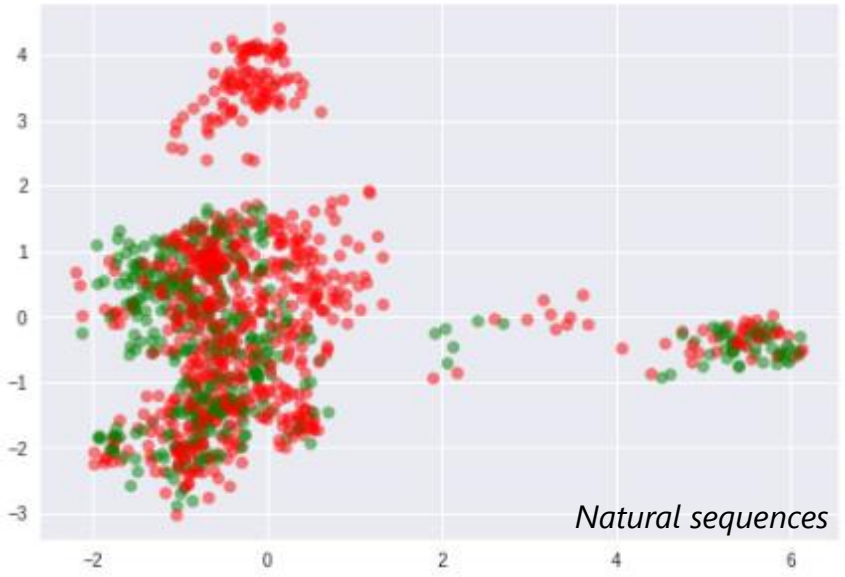
- Plot of *functional* and *non-functional* **natural sequences** using the first two PCs.
- Functional → Green
- Non-functional → Red

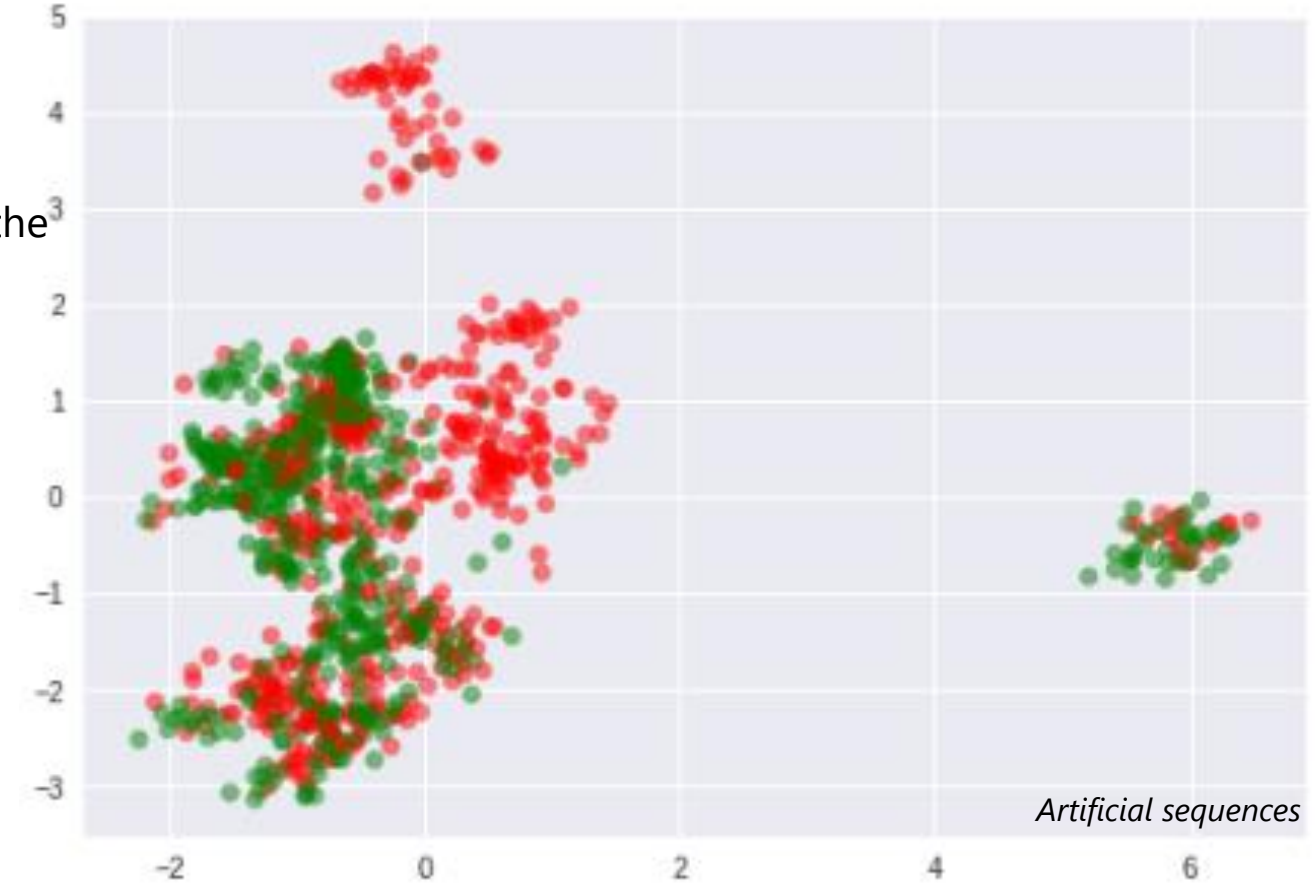
Q: Are functional and non-functional sequences well separated in 2D-PC space?



Task 2: Dimensional reduction and visualization of sequence space

- Projection of **artificial sequences** onto natural sequences PCs.
- We observe that:
  - They occupy same space as natural sequences
  - Distribution of functional and non-functional *almost* the same
  - Missing artificial samples in some regions



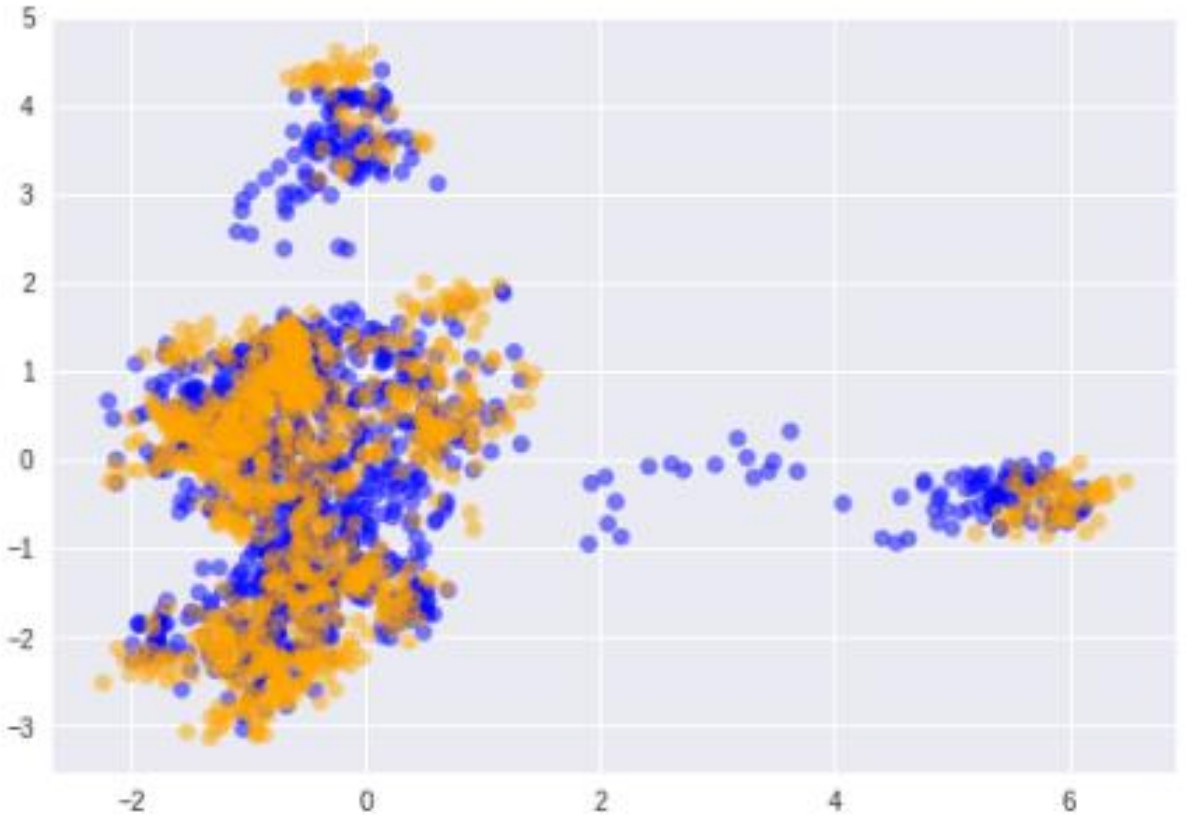


Task 2: Dimensional reduction and visualization of sequence space

- Projection of natural and artificial sequences onto PCs of natural sequences.

- Natural → Blue
- Artificial → Orange

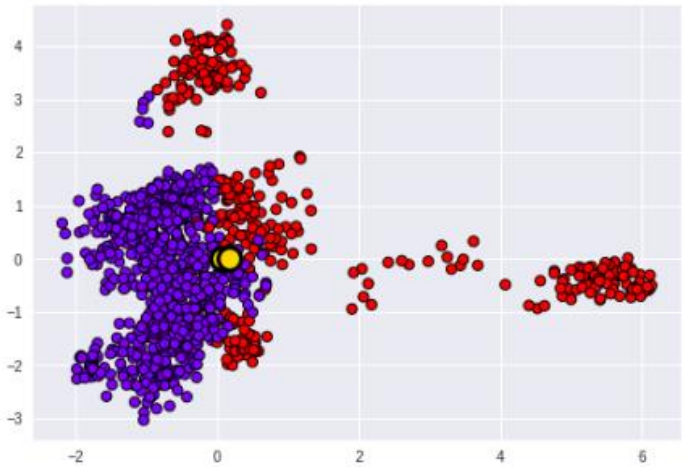
Q: Do they occupy a similar region in 2D-PC space?



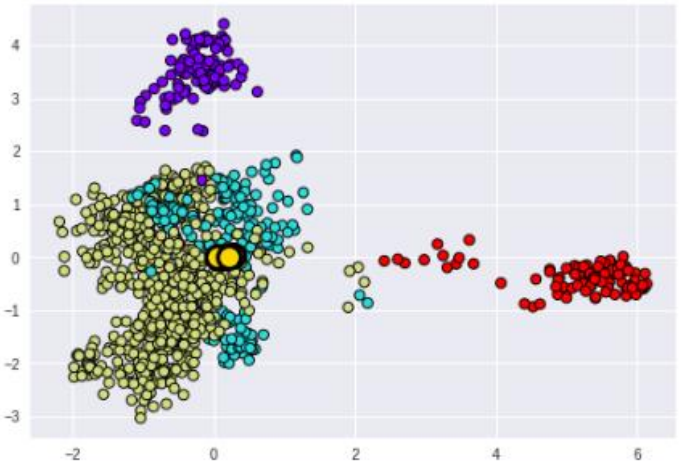
Task 3: Clustering sequence data

Natural sequences: functional vs non-functional

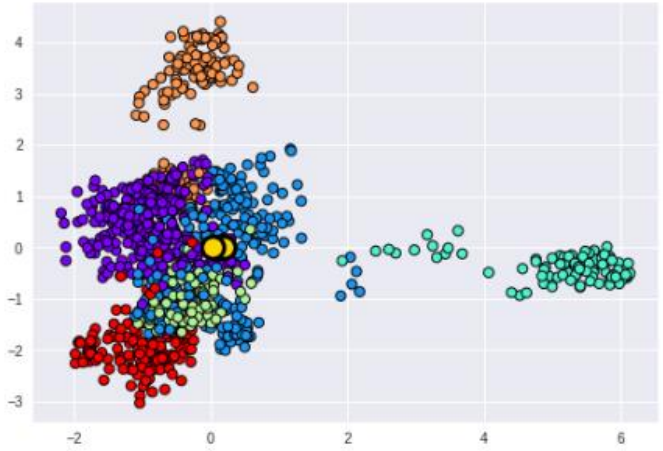
- Clustering on the **natural sequences** with K-means algorithm.



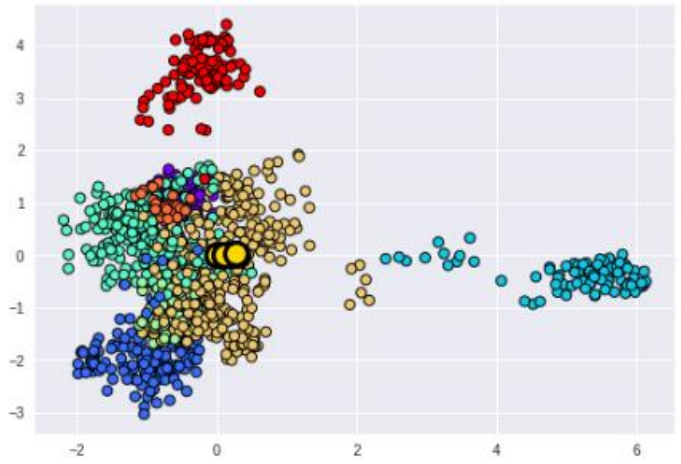
2 clusters



4 clusters



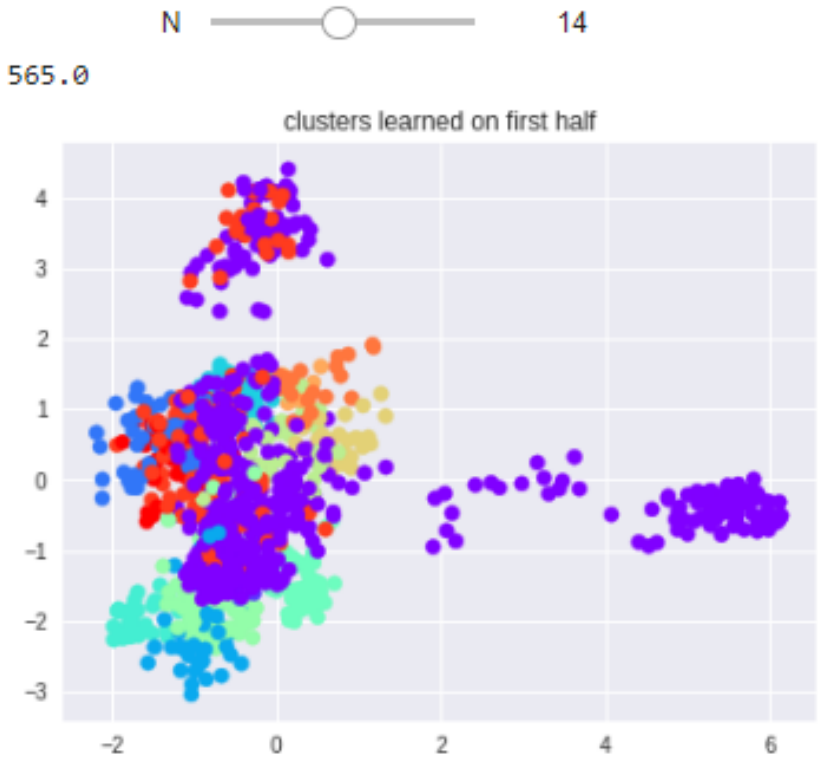
6 clusters



8 clusters

Task 3: Clustering sequence data

Natural sequences: functional vs non-functional

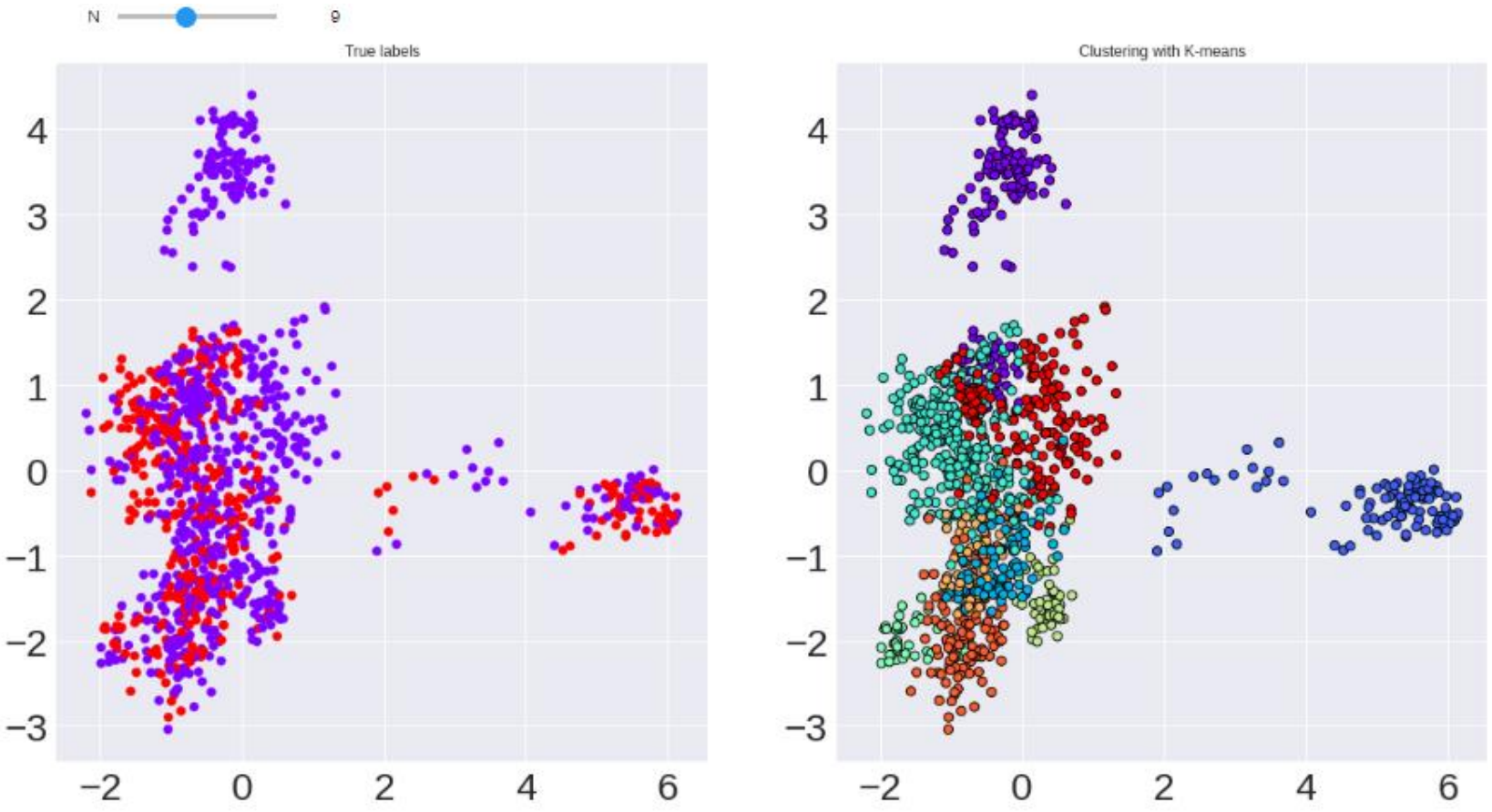


- Application of different numbers of cluster on two different partitions of the **natural sequences** dataset

- No number of clusters for which the two algorithms can agree upon.

Task 3: Clustering sequence data

Natural sequences: functional vs non-functional



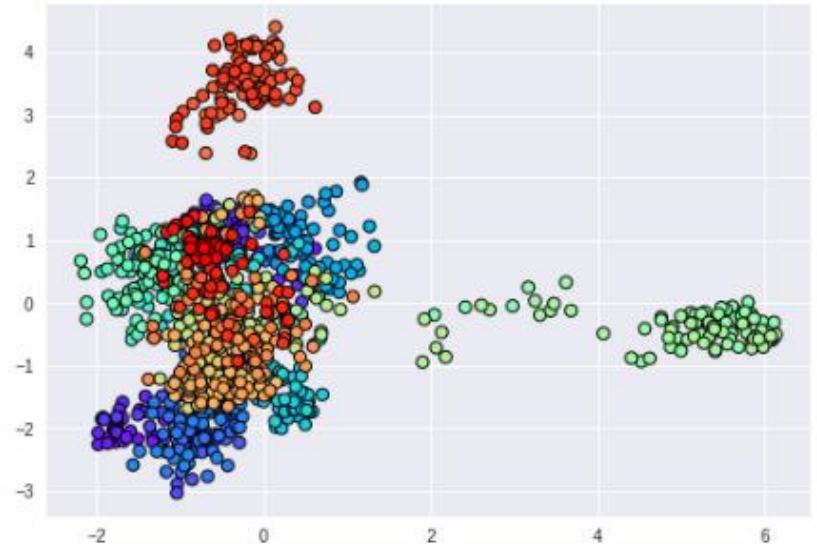
- Purple dots → non-functional sequences
- Red dots → the functional sequences

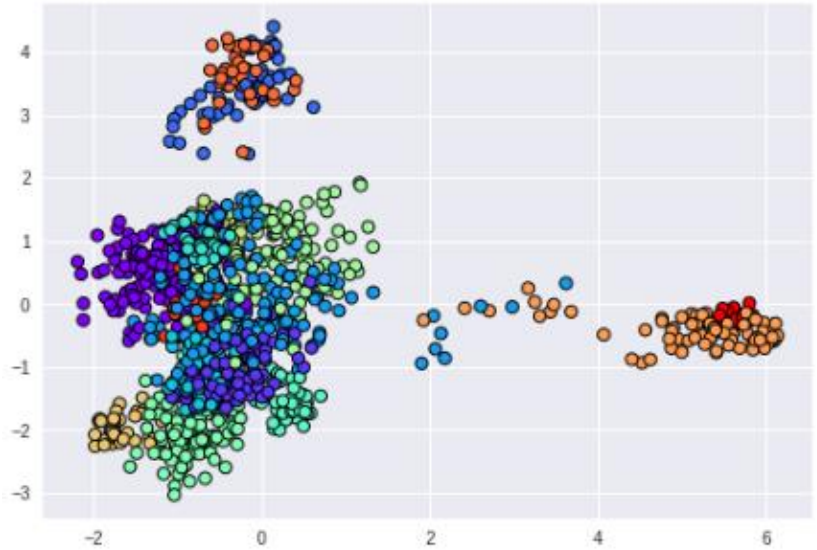
Result: clustering is not very useful for distinguishing functional and non-functional sequences.

Q: Are functional and nonfunctional sequences separated into distinct clusters?

Task 3: Clustering sequence data

Natural sequences: functional vs non-functional





- **DBSCAN**: not good results in clustering task (different values of epsilon and min-sample tried)

- **Agglomerative Clustering** result: no improvement with respect to K-means

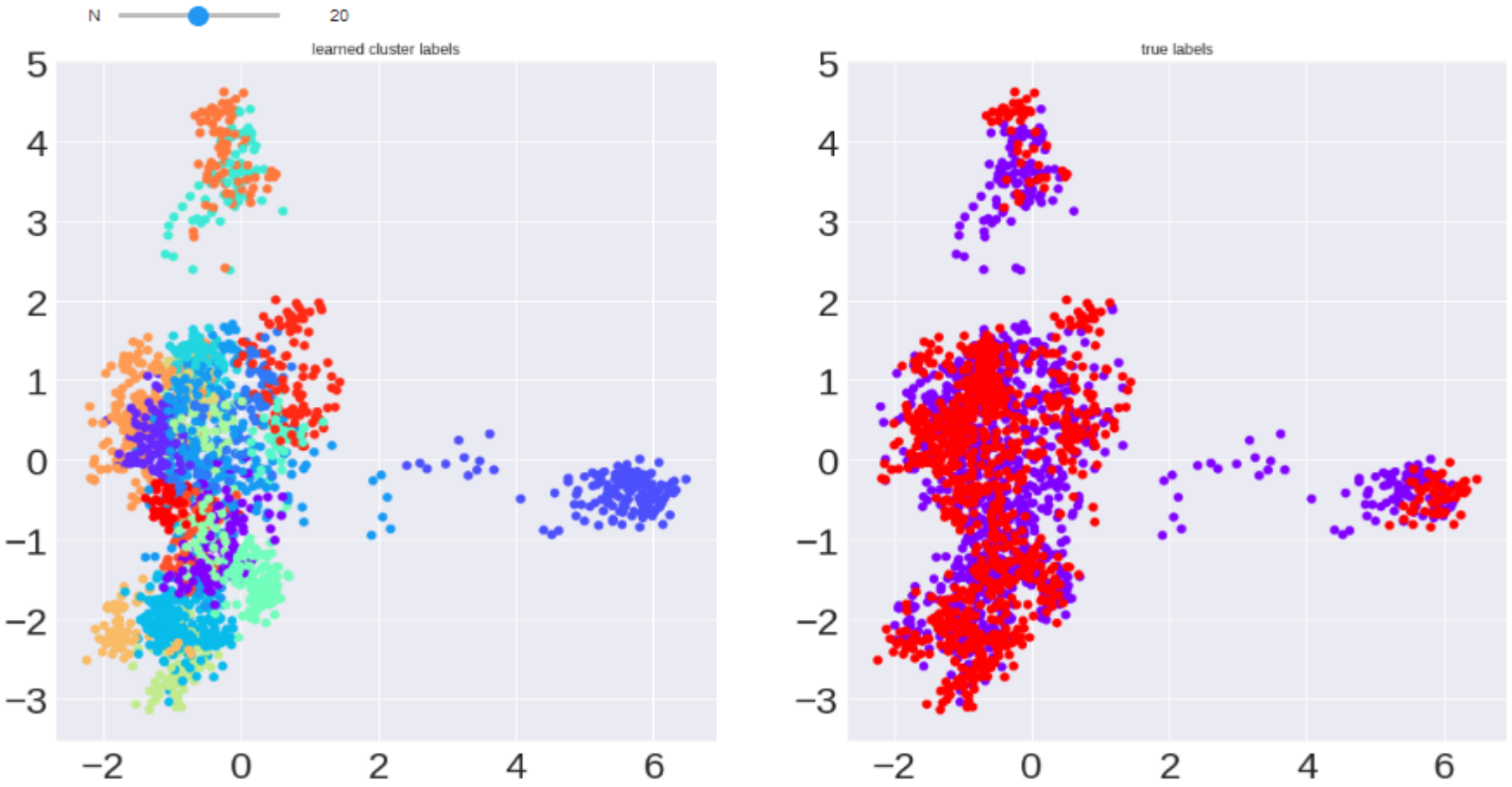
Task 3: Clustering sequence data

Natural sequences: functional vs non-functional

- **Agglomerative clustering**
- Comparison between True labels and Learned cluster labels with agglomerative clustering algorithm.

Task 3: Clustering sequence data

Natural vs artificial sequences



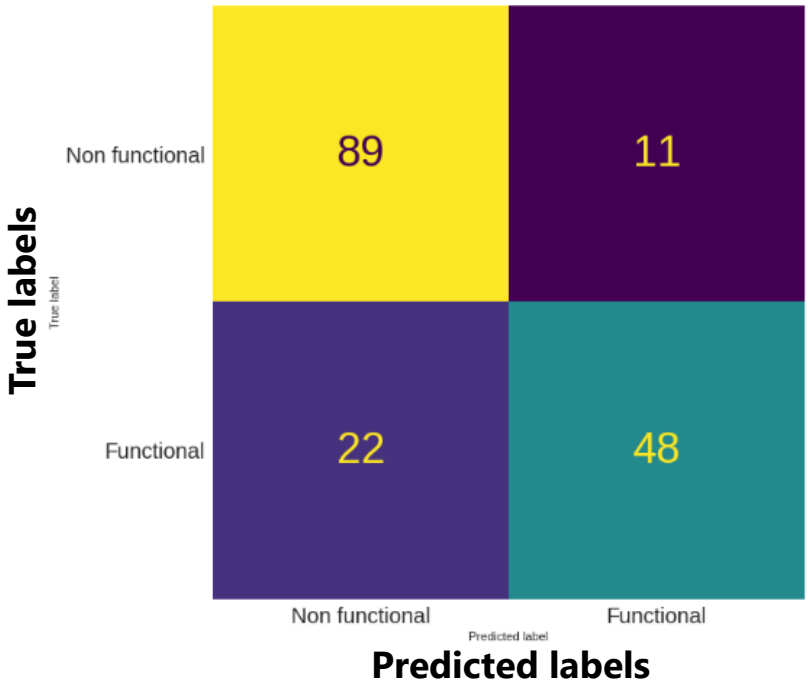
- Comparison between true labels and learned cluster labels with Kmeans Algorithm (natural and artificial sequences)
- Purple dots → natural sequences
- Red dots → artificial sequences

Q: Are the two datasets separated by this procedure, or are clusters mixed in natural and artificial sequences?

Task 4: Predicting protein functionality

Natural sequences

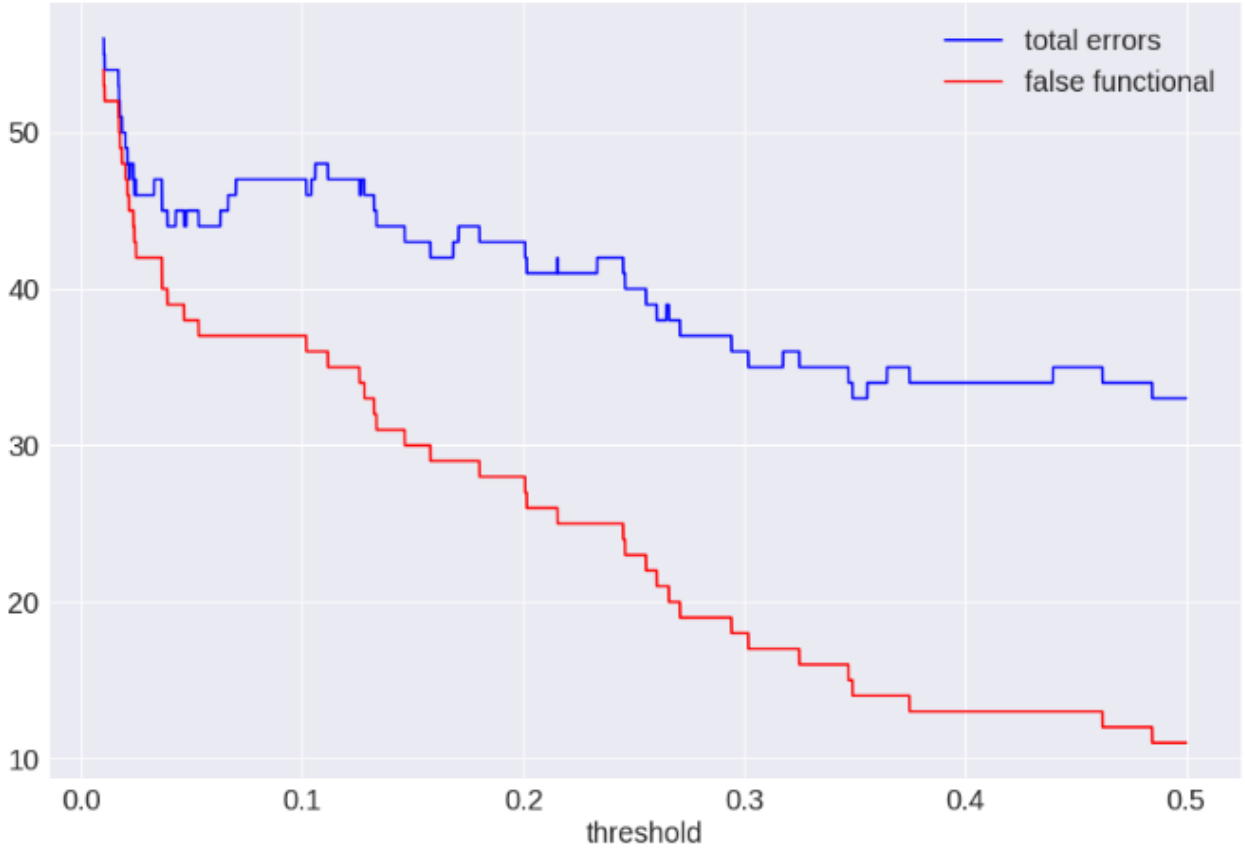
- Division of the dataset into Training, Validation and Test Dataset
- Logistic Regression as classifier
- Fitting of the model on Training Dataset → prediction on Validation Dataset for adjusting the threshold



- Confusion Matrix of the validation set
- 81% of accuracy

Task 4: Predicting protein functionality

Natural sequences



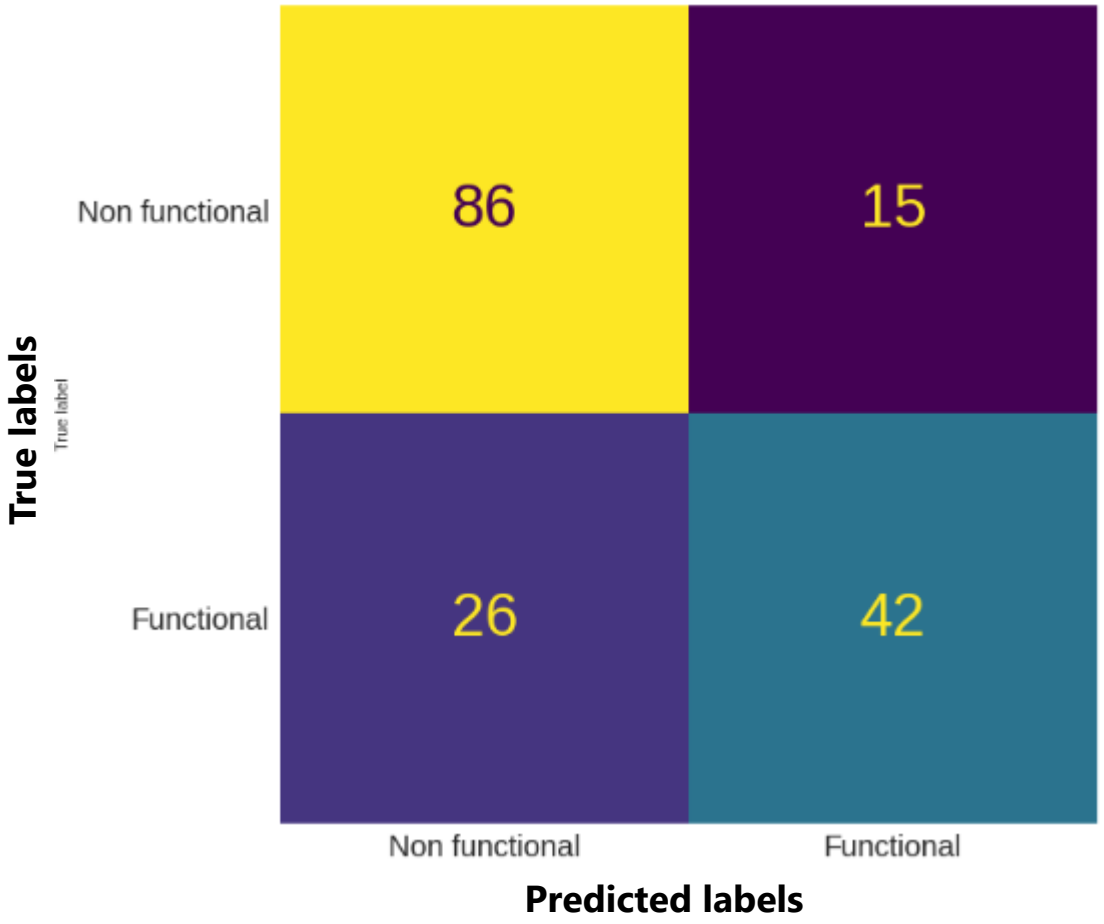
- Evaluation of **total errors** and **false functionals** as a function of the threshold (hyperparameter).
- Here the threshold is such that:

  P(x is functional) > thr → x is functional
  P(x is functional) < thr → x is not functional

- By increasing the threshold, both the total errors and the total number of false functionals decrease → keep the threshold equal to **0.5**.

Task 4: Predicting protein functionality

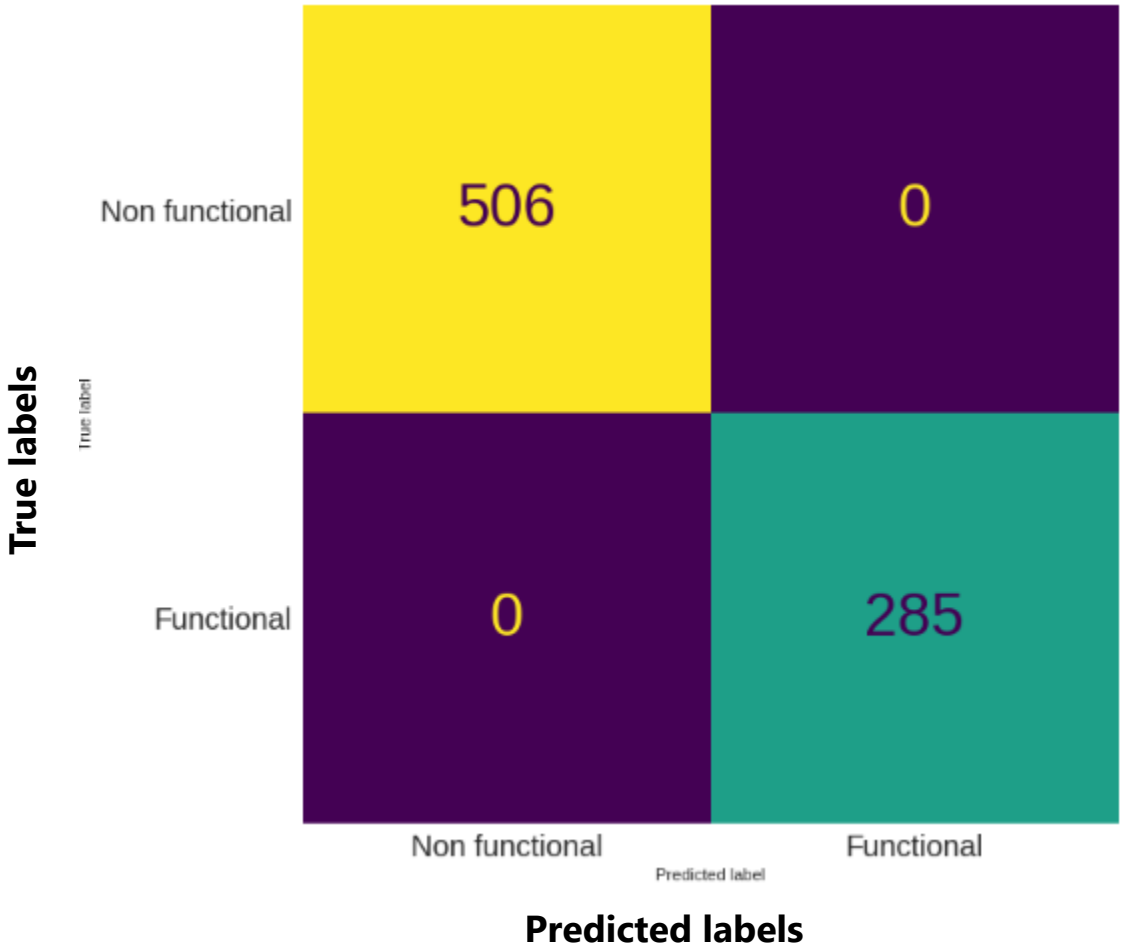
Natural sequences



- Confusion matrix by mantaining a threshold at 0.5 on **test dataset**
- 76% accuracy

Task 4: Predicting protein functionality

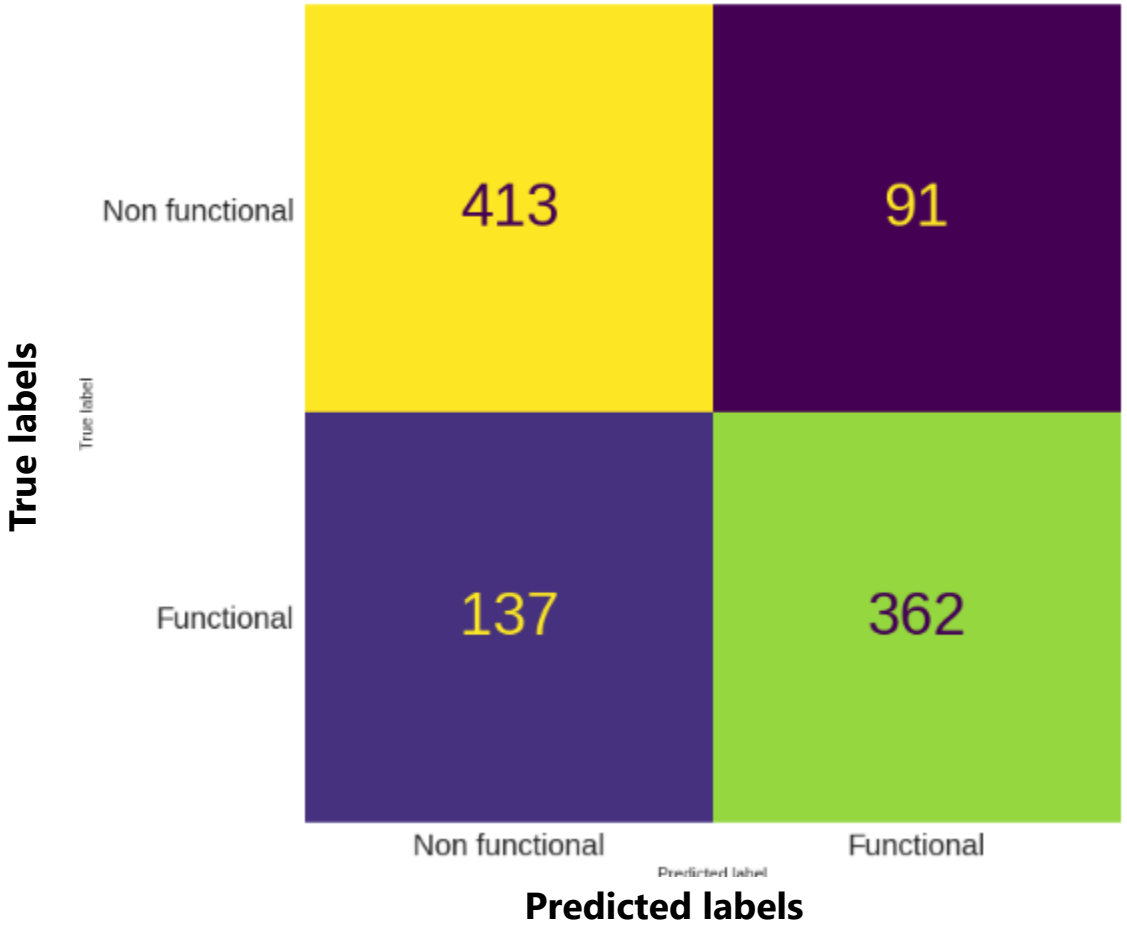
Natural sequences



- Threshold: 0.5
- Confusion matrix performed on the **training set**
- 100% accuracy

Task 4: Predicting protein functionality

Artificial sequences



- Threshold: 0.5
- Confusion matrix of the **artificial sequences**
- 77% accuracy
- Note that the training was done on the natural sequences

Task 5: Generating artificial sequences

- Pseudolikelihood Maximization Algorithm
- Estimation of the probability of the sequence as the factorization of each site probability given the amino acids on the other sites .

$$P(\bar{s}) = \prod_{i=1}^{N} P(s_i \mid \underline{s}_{-i}, h_i, \{J_{ij}\})$$

- Find:

$$\{J_{ij}^*, h_i^*\} = \underset{\{J,h\}}{\operatorname{argmax}}\, PL(J,h \mid D) \text{ where:}$$

$$\cdot PL(J,h \mid D) = \sum_{i=1}^{N} PL_i(h_i \{J_{ij} \mid i \neq j\} \mid D)$$

$$\cdot PL_i(h_i, \{J_{ij} \mid j \neq i\} \mid D) = \frac{1}{M} \sum_{m=1}^{M} \log P(s_i^m \mid s_{-i}^m, h_i, \{J_{ij} \mid j \neq i\})$$

Task 5: Generating artificial sequences

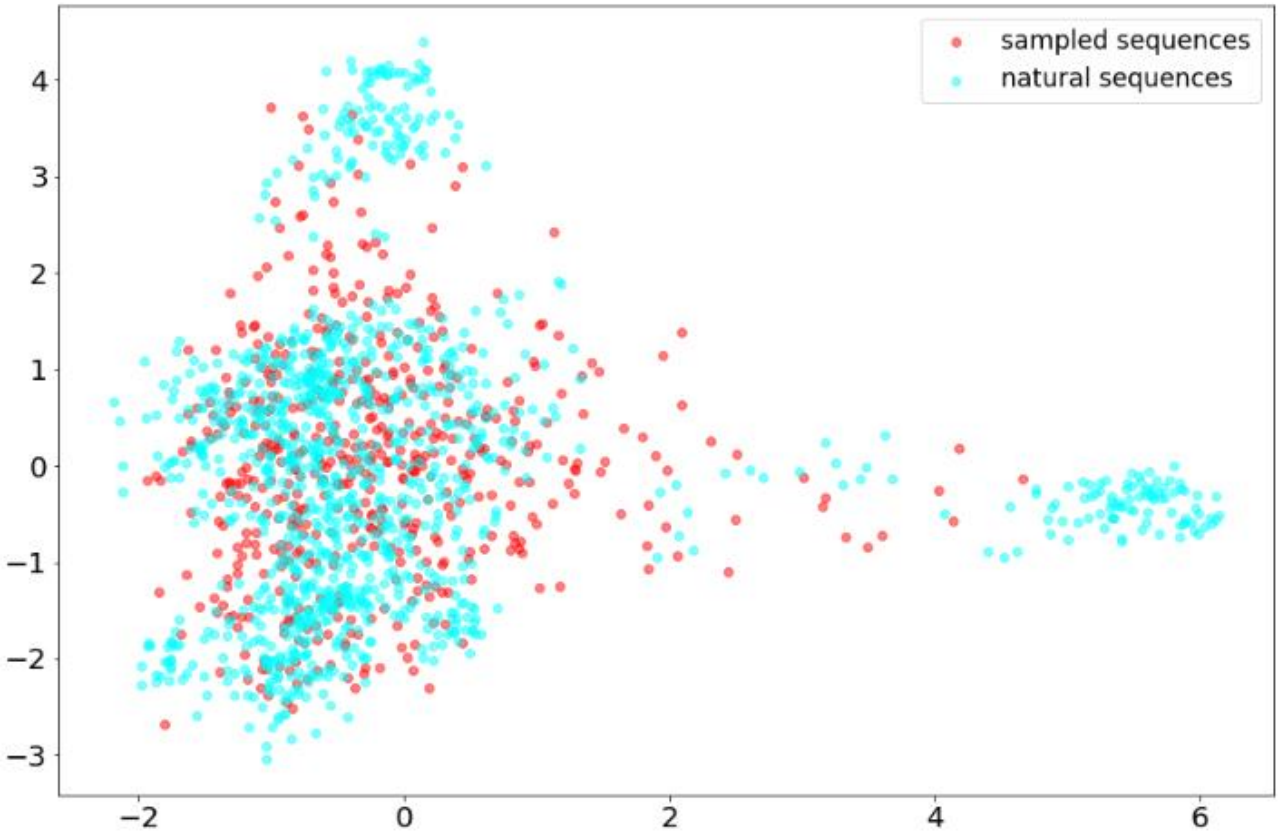
- To find the single probability for each site we performed a multi-class logistic regression (Softmax)
- Hard to sample from real probability distribution → Sample from an approximation of the probability, given by the conditional probabilities of each site → Gibbs Sampling
- Gibbs Sampling generates a correlated Markov Chain. In order for MC to reach stationarity → wait 1000 steps before sampling → next, wait 1000 steps between sampling in order to have independent samples
- Gibbs Sampling Algorithm:

1. Random initial condition taken from a uniform probability distribution
2. To get the next sample at step i+1 we sample each site at step i+1 (in sequential order) from the probability distribution:

$$P(s_j^{(i+1)} \mid s_1^{(i+1)}, \ldots, s_{j-1}^{(i+1)}, s_{j+1}^{(i)}, \ldots, s_N^{(i)})$$

By repeating these steps we obtained 540 different and independent sequences

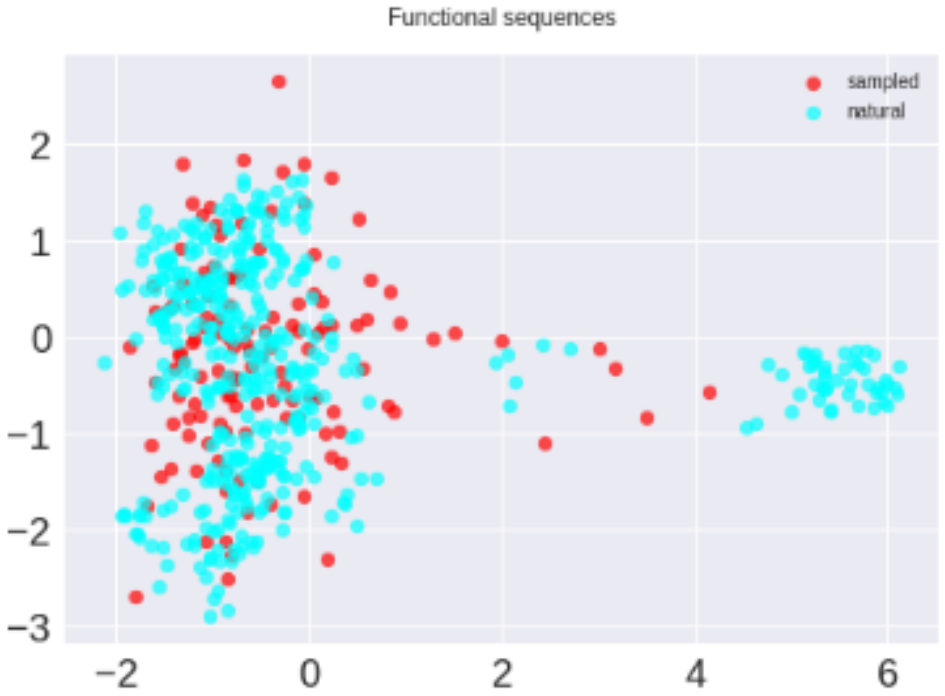
Task 5: Generating artificial sequences

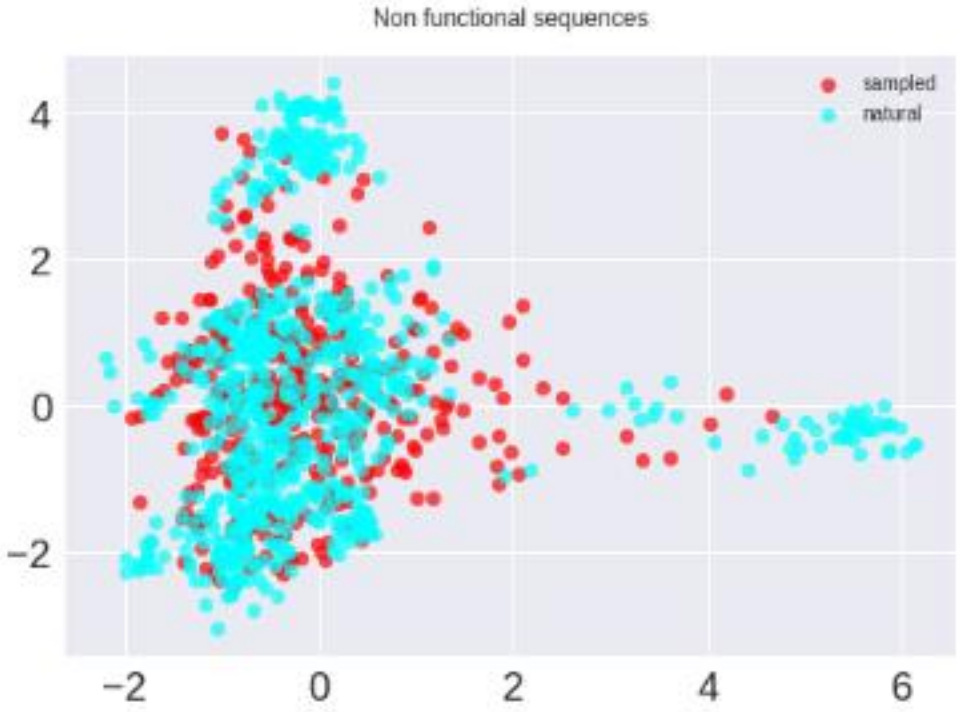


- Sampled and natural sequences in 2D-PCs of natural sequences

Q: Do they occupy similar regions?

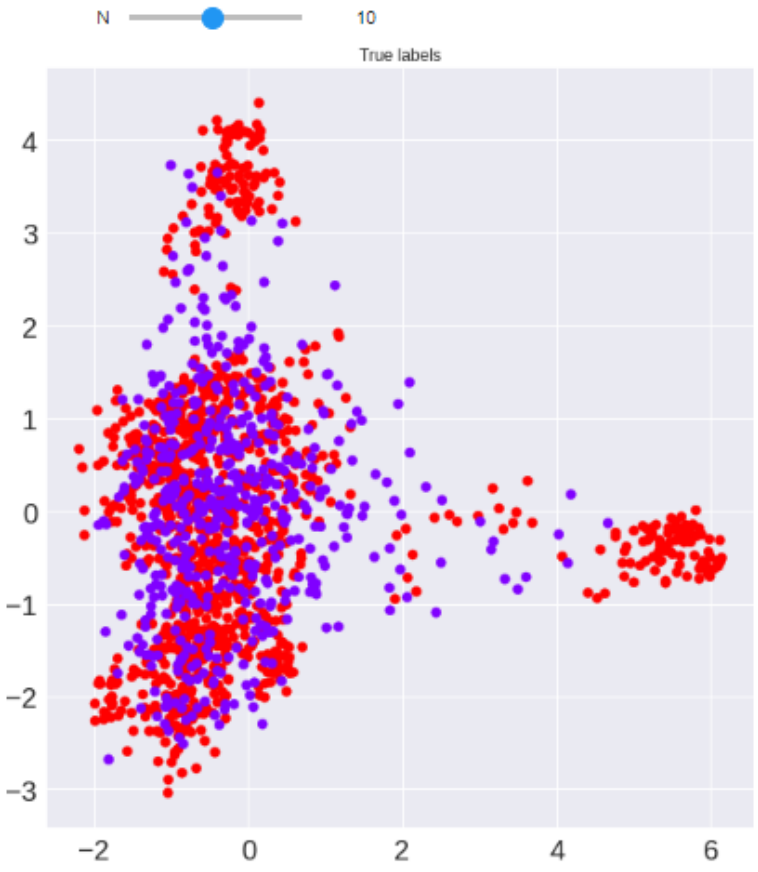
Task 5: Generating artificial sequences

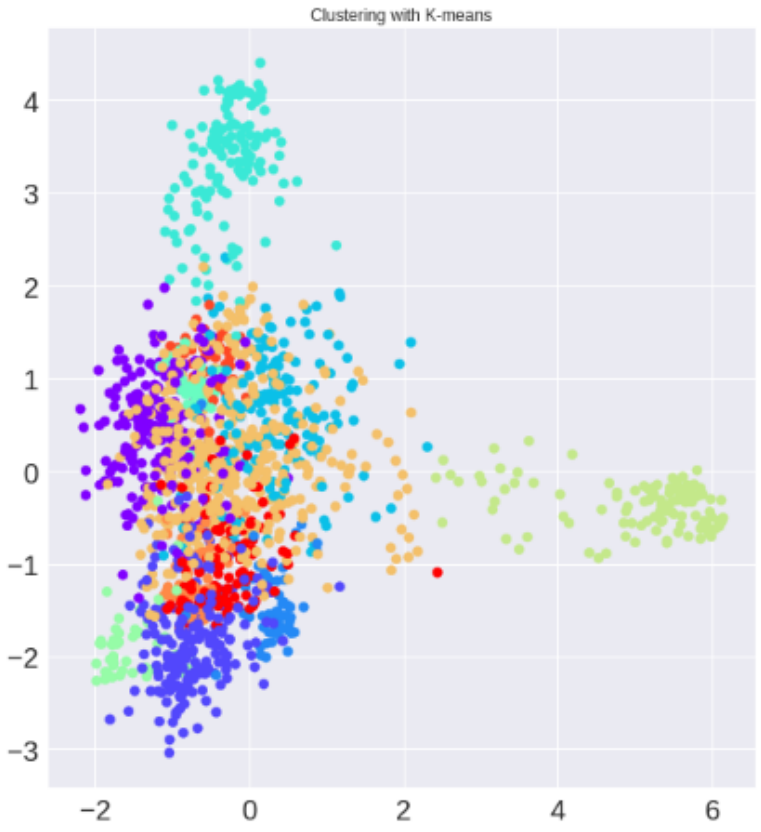




- With a logistic regression, trained on the natural sequences, we predicted the functionality of the sampled sequences. As a result, we obtained 131 functional sequences and 409 non functional sequences.

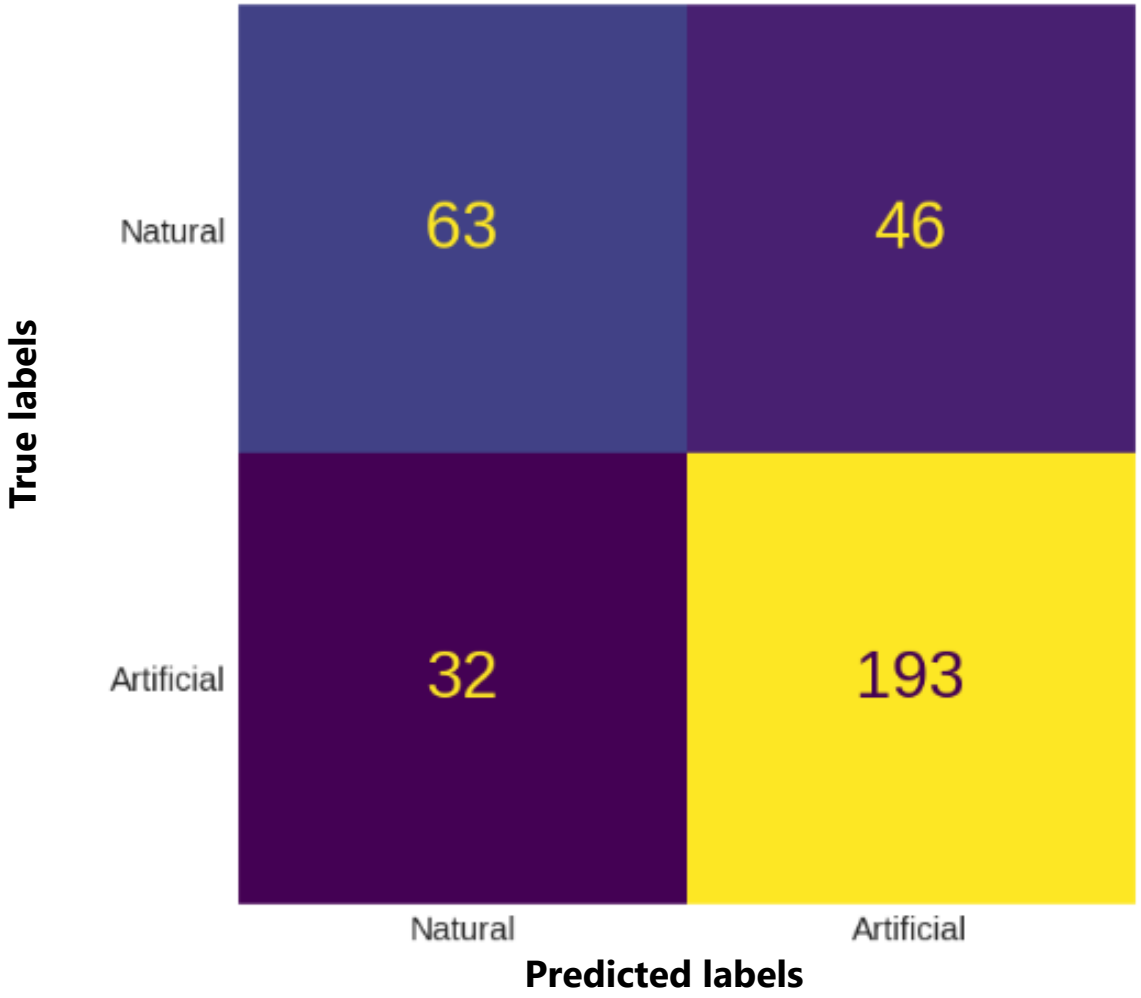
Task 5: Generating artificial sequences





- K-means algorithm applied on both natural and sampled sequences.
- The clustering does not divide the two types of sequences.
- Note that red dots are the natural sequences, while the purple dots are the sampled sequences.

Task 5: Generating artificial sequences



Q: Is our generative model a good model?

- To answer this, we performed a logistic regression, trained on a data set that contained the natural and sampled sequences.
- The data: 1130 natural sequences and 540 artificial sampled sequences.
- Results:
  - Confusion matrix on the test dataset
  - 77 % accuracy
- More analysis needed.