Unsupervised and supervised analysis of protein sequences

Computational Science (M. Weigt)

Master's Degree in Physics of Complex Systems

Bicocchi Pietro, Cavallero Simone

January 30, 2023

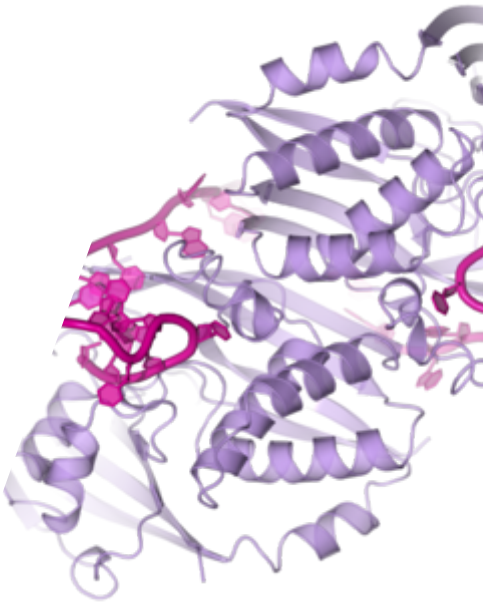


Table of Contents

Proteins

Proteins are large, highly complex and naturally occurring molecules can be found in all living organisms. A long chain of amino acids joined together by peptide bonds form proteins.

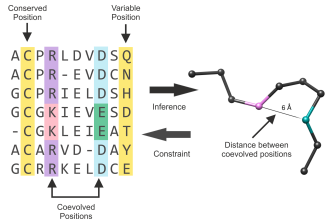


Figure: MSA

- We look for information in Multiple-sequence-Allignements (homologous sequences)
- It's difficult to get from a chain how the protein fold and function (limited data about secondary, tertiary etc. structure)

A lot of sequences of a protein are collected in a protein database. (ex: UniProt) stored in MSA.

MSA

Allignment of sequences that stores multicategorical variables (each element can assume 21 values)

Two datasets

- Natural sequences (1130)
- Artificial sequences (1003)

Data structure

- Data dimensionality: 96x20 dimensional data point
- Categories: **functional** or non-functional sequences

Figure: MSA

- **One-hot encoding** leads to use a 20-dimensional representation for each amminoacid, while the gap is mapped to the zero-vector.
- It blows up the feature vectors from L = 96 categorical variables to 20L = 1920 binary variables, but the numerical treatment is easier.

We construct a dictionary in order to rewrite the two matrices by changing the description of each amminoacid:

VARIABLE	CONTENT
seq_onehot_0	1130x1920 matrix of '0' and '1' for **natural** sequences
seq_onehot_1	1003x1920 matrix of '0' and '1' for **natural** sequences

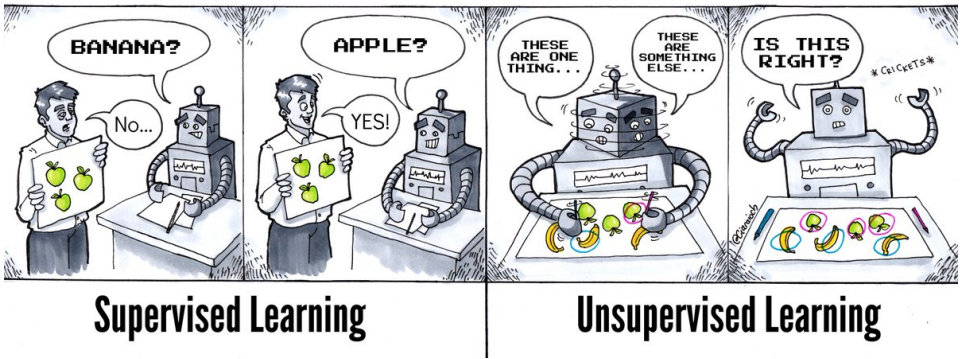


Figure: Supervised vs Unsupervised

Table of Contents

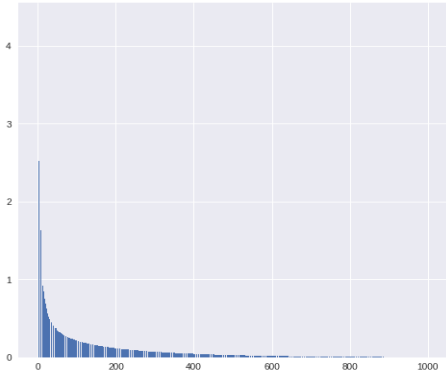
2 Dimensional reduction and data visualization

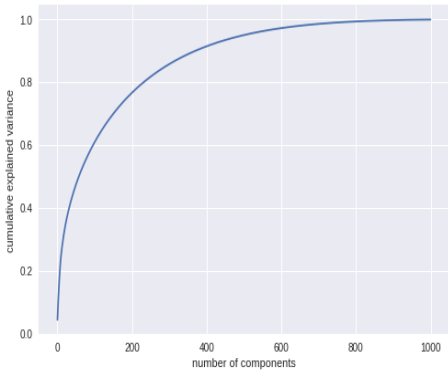
- **Curse of dimensionality:** large number of dimensions implies difficult in process the data and extract relevant information.
- **Dimensionality reduction:** reduce the complexity of the dataset via feature extraction

PCA

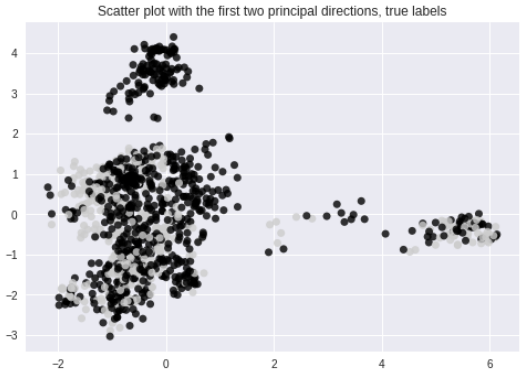
Principal Component Analysis aims to find the directions of maximum variance in high-dimensional data and projects it onto a new subspace with equal or fewer dimensions than the original one.

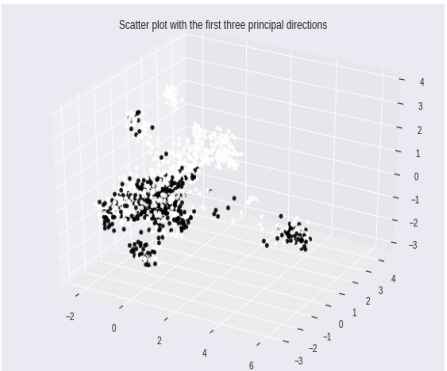
- **Goal:** to transform our dataset into one with a reduced dimensionality, without loosing too much information

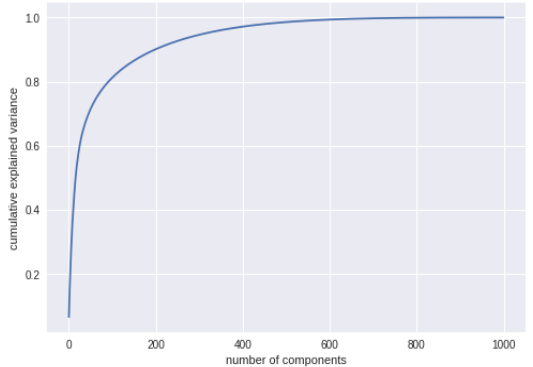




We just need around **370 features** in the new diagonalized and reduced sub-space instead of 1920 to keep almost all (90 percent) the information


Scatter plot with the first two principal directions, true labels
4
3
2
1
0
−1
−2
−3
−2
0
2
4
6


Scatter plot with the first three principal directions



In this case, one has to take around **200 dimensions** to keep 0.9 of the variance.

PCA - Artificial sequences

2 Dimensional reduction and data visualization

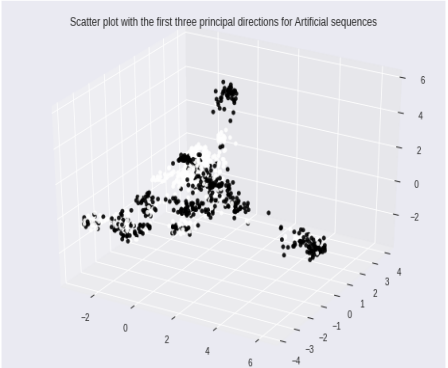


Table of Contents

3 Clustering

The aim of this task is now to perform **clustering** on the datasets. We want to see if the method is able to discriminate:

- Within the natural sequences, the functional from the non-functional ones.
- Within a concatenated dataset, the natural from the artificial ones.

Method: we are going to use K-Means clustering algorithm.

A **Cluster** is a collection of data points aggregated together because of certain similarities.

K-Means

The algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

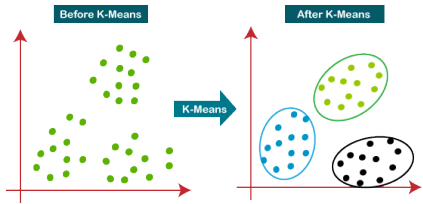
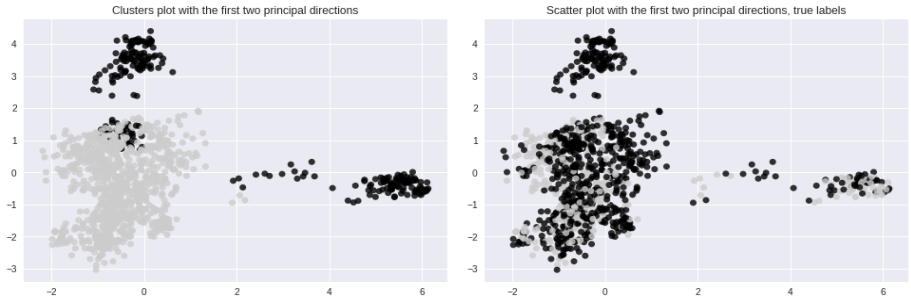
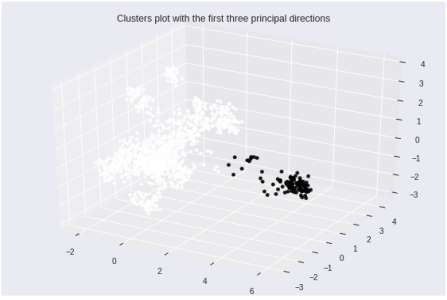
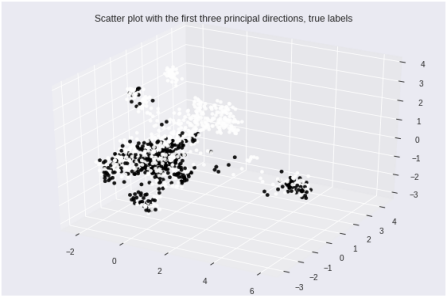


Figure: K-Means

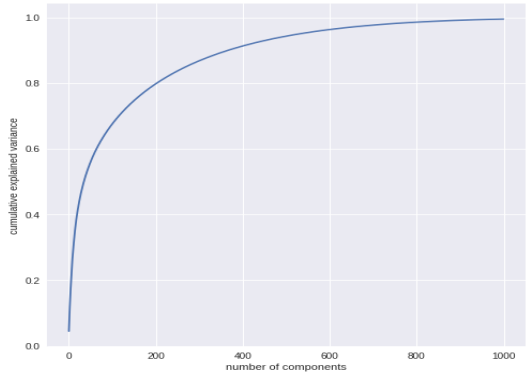

Clusters plot with the first two principal directions
Scatter plot with the first two principal directions, true labels
4
3
2
1
0
−1
−2
−3
−2
0
2
4
6





As we can see, the data don't seem in general to respect and follow the labels with two clusters,**functional and non functional sequences are not separated into different clusters**.

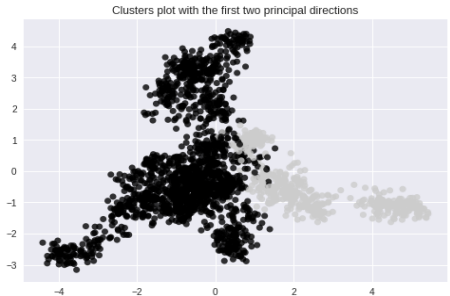
Doing PCA on the concatened set:

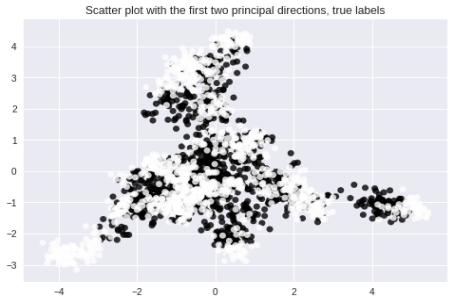


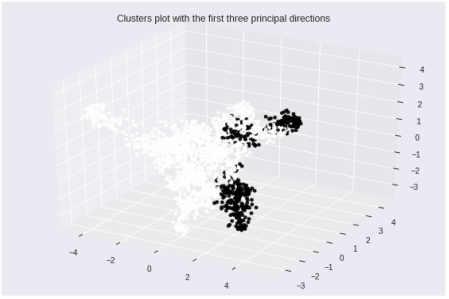
Hence, we need to keep around **360 dimensions** for this total set.

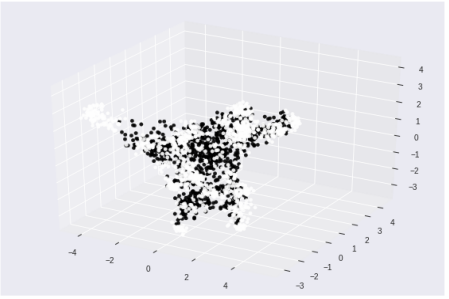
K-means on concatenated dataset

3 Clustering









Again, as we can see, the clustering algorithm was not able to catch the different features between **natural and artificial** thus, the two are not well discriminated.

Table of Contents

4 Predicting protein functionality

Idea: we focus now on a Supervised Learning task, for which we want to use a **classifier** in order to select as precisely as possible the correct label for our datapoints.
We are going to test several models:

- **Logistic Regression**
- Ensambling models
- K-NN

Splitting Data

- Training Set: 80 percent of the data
- Test Set: 20 percent of the data

Goals

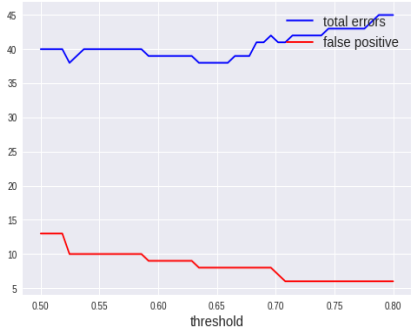
- Maximize Test score
- Avoid overfitting
- Keep the model as simple as possible

The score of the model results in:

- 99% on training set
- 73% on test set

With the following confusion matrix

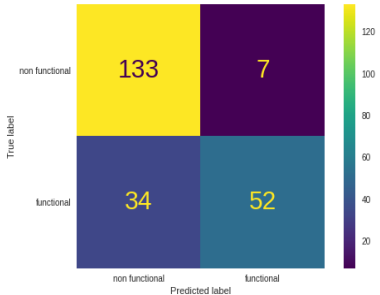
One can optimize the number of false positive and the total error changing the **treshold probability** for which the soft classifier gives a binary result.



The perfect trade-off seems to be around a treshold of **0.70**

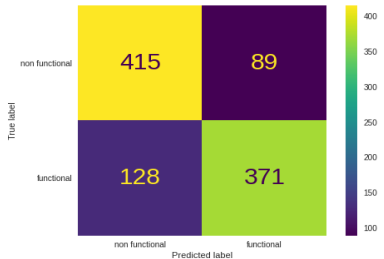
Setting this new treshold, we can see that the model will predict less false positive, increasing the average score of the model:

- Average score on test set 78.3%
- Std of the score 2.3%



One can now try to use the entire set of **natural** sequences as the **training** one, while the **artificial** sequences will be used as **test** set for the logistic model:
The score of the model results in:

- 99% on training set
- 78% on test set



One can ask how the performance depend on the **variability of the random state**, introduced necessarily everytime logistic regression is done, due to the splitting of the dataset. Making 100 iterations one gets:

- Average score 78.71%
- Std 2.44%

Higher than what we got without averaging over randomness.

How good the model perform under **k-cross validation**?
We will split the dataset into k subsets to see many possible train-validation combinations.

Using a **2-cross validation** one obtains:

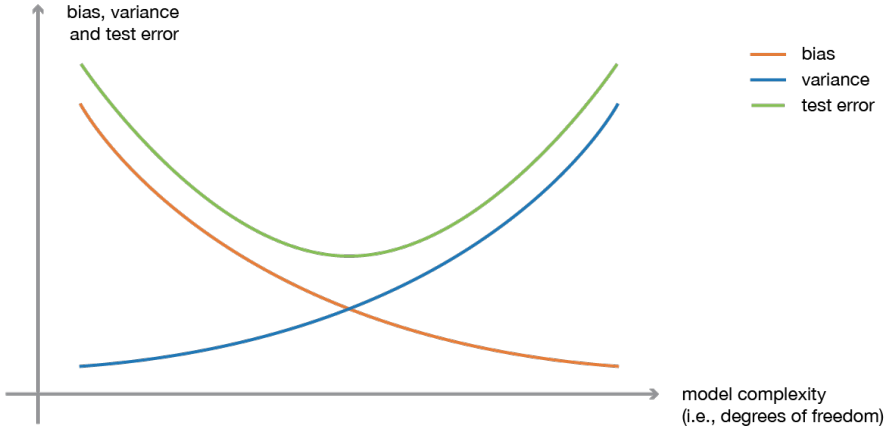
- score 0.756
- score 0.763

Whereas with a **10-cross validation** one obtains an average of 0.578, way lower than what expected!

Ensembling

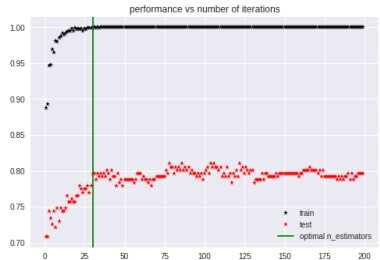
Ensembling strategies are ways to combine a certain number of weak classifier into a strong classifier.

- **What is a weak classifier?** A classifier whose prediction are weakly correlated to the real label (high-variance)
- **What is a strong classifier?** A good classifier (good bias-variance tradeoff)


bias, variance
and test error
bias
variance
test error
model complexity
(i.e., degrees of freedom)
underfitting
(high bias)
optimal
balance
overfitting
(high variance)

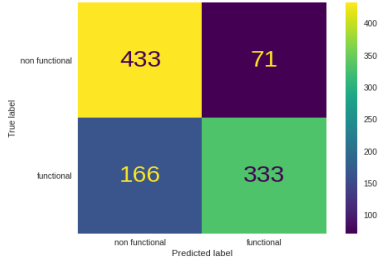
A **Random Forest** classifier is an ensemble method that gives as a result the majority vote between different decision trees

- We can measure the optimal number of trees to get the highest train and test score:



Giving the highest score of 79.6% when the number of estimators gets equal to **30**.

One can train the Random Forest classifier on a training set of the natural sequences, to then test it on the **total artificial dataset**. The confusion matrix looks like:



Evaluating the score now on the **test artificial set** this will be of **78.5**%

- Using **AdaBoost** as a strong classifier instead, the score has been evaluated to be of **77%**.
- Using **K-nn** with 20 neighbours and after PCA, the score has been evaluated to be of **73%**.

Table of Contents

5 Generating artificial sequences

Generative energy-based models are a class of models that are able to generate new data samples similar to a given training set. They work by learning an energy function that assigns a scalar value, called energy, to each possible configuration of the model's parameters and the input data.

Why?

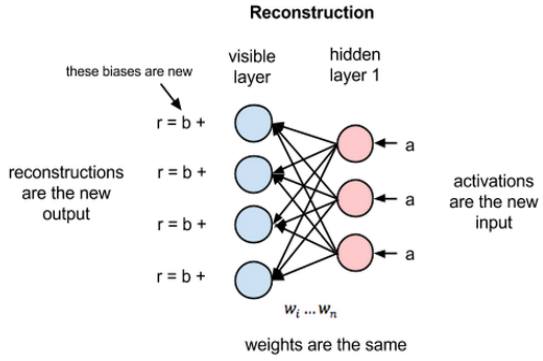
- By leveraging large datasets of known proteins, generative models can learn the patterns and rules governing protein folding
- to generate novel sequences that have never been seen before.

RBM

A Restricted Boltzmann Machine (RBM) is a popular choice for creating a generative model for protein sequences. RBMs are a type of generative stochastic artificial neural network that can learn to model the probability distribution of the input data.

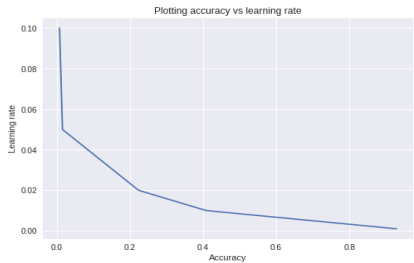
The RBM consists of two layers:

- **visible layer**, which represents the input sequences
- **hidden layer**, which captures the underlying patterns in the data.



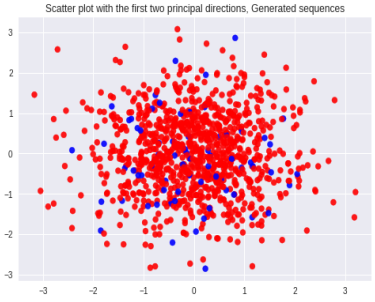
The two layers' variables are conditionally dependent on each other; in each iteration of **Gibbs sampling**, a sample from the hidden layer is generated given the current sample of the visible layer, and then a sample from the visible layer is generated given the updated sample of the hidden layer.

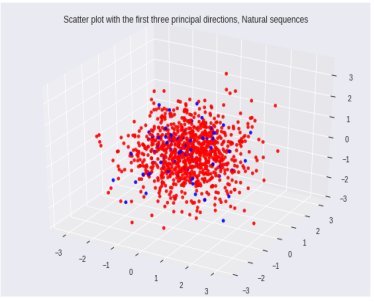
The velocity of convergence through gradient descent is led by the **learning rate**. One can play both with this and the **accuracy** (amount of proteins correctly predicted)



As we can see, an high learning rate leads to a small accuracy and viceversa, so we are going to use 21 hidden layers (as the number of important features should reflect the description of an aminoacid) and a learning rate of 0.001 to have a good accuracy.

The average prediction of the RBM on new proteins training the net with the natural sequences is 92.8%, which is an amazing result thanks to the parameters we chose. The prediction looks:





An amazing result... Let's discuss it

The model we chose is simple... only an hidden layer.
Q: Why this result then?

- Overfitting? Model has memorized the training data, and not generalizing well to new examples.
- Non variability? Maybe it is really good to produce just samples really close to a certain class without generalizing.

We could implement:

- Deep Belief Networks
- VAE
- GAN[1]

We could try to visualize data differently: (so far PCA... clustering)

- t-SNE (t-distributed Stochastic Neighbor Embedding)
- Kernel PCA

[1] ProteinGAN: A generative adversarial network that generates functional protein sequences

clearbox.ai

Me: I need more data

My generative model: Say no more

- “A fast learning algorithm for deep belief nets” G. Hinton, S. Osindero, Y.-W. Teh, 2006
- “Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient” T. Tieleman, 2008
- ”Inverse statistical physics of protein sequences: a key issues review” Simona Cocco, Christoph Feinauer, Matteo Figliuzzi, Rémi Monasson and Martin Weigt

Unsupervised and supervised analysis of protein sequences

Thank you for listening!

Any questions?