Kernel Methods Challenge : DNA sequence classification

Shahine Bouabid Alex Delalande Houssam Zenati
Team Name: Kernelito
CentraleSupélec - ENS Paris Saclay
`firstname.lastname@student.ecp.fr`

Abstract

The goal of this challenge is to engineer kernels and classification algorithms in order to predict if a DNA sequence is prone to accept a specific transcription factor. We implement general sting kernels and biological sequences kernels along with classifiers relying on a kernel formulation, and evaluate the different kernels and classifiers against each other on the provided dataset. Results show that a simple and sound choice of kernel and classifier with well chosen hyperparameters allow to reach competitive results on the Kaggle challenge.

1. Introduction

A DNA sequence consists of a word built out of the alphabet $\{A, T, C, G\}$, each character depicting a possible nucleotide. The goal here was to build a classifier able to predict whether a given sequence would be binding to a specific transcription factor.

We were provided with 3 training sets (denoted Dataset 0, 1 and 2) consisting each of 2000 DNA sequences, along with 3 testing sets consisting each of 1000 DNA sequences. All DNA sequences were of length 101, hence inducing 4^{101} possible words. The inherent string type of such data led us to naturally consider string kernels such as Spectrum Kernels [4][2] or Weighted Degree Kernels [6] in order to efficiently tackle sequences characterization against such numerous possible combinations of nucleotides. Another approach was to experiment kernels designed for biological sequences processing purposes with Local Alignement Kernels [7].

In the following, we will first present the explored kernels and classifiers while pointing out insights on the data, before briefly motivating the model we decided to stick with and discuss the obtained results. **All the code and precomputed kernels are available on our Dropbox.**

2. Explored kernels

2.1. String Kernels

2.1.1 Spectrum and Mismatch Kernels

Given the small size of the alphabet, we decided to implement a preindexed version of these algorithms[4][2], hence ensuring computational efficiency. While using the spectrum kernel for our early experiment, we noted that mers length between 7 and 9 were providing the best results but started to get too picky when confronted with the task of matching two sequences. We hence switched to the Mismatch Kernel in order to allow non-exact matches and keep a reasonably distributed range of pairwise scores, thus embedding more granular similarity information as depicted in Figure 1.

Mismatch Kernel can be tuned with the mer length considered n and the maximal number of allowed mismatch k. It came out that $(n = 8, k = 1)$ did score best for dataset 0 and 2 while $(n = 9, k = 1)$ outperformed it on dataset 1. It reveals dataset 1 can actually be better characterized by mers of length 9.

2.1.2 Weighted Degree Kernels

Spectrum-like kernels did not weight in any positional information, which is essential in genomics. We hence decided to try the weighted degree kernel. However, sequences matching with exact positional alignment constraint seemed too restrictive to extract a relevant characterization of such small datasets. We hence decided to go with a relaxed version of the algorithm allowing shift [6] but also increasing computational cost to $\mathcal{O}(ns)$ where n is the maximal mer size considered and s maximal shift allowed. Due to this computational greediness, we did not perform parameter tuning and chose $n = 20$ and $s = 5$ as recommended in the literature.

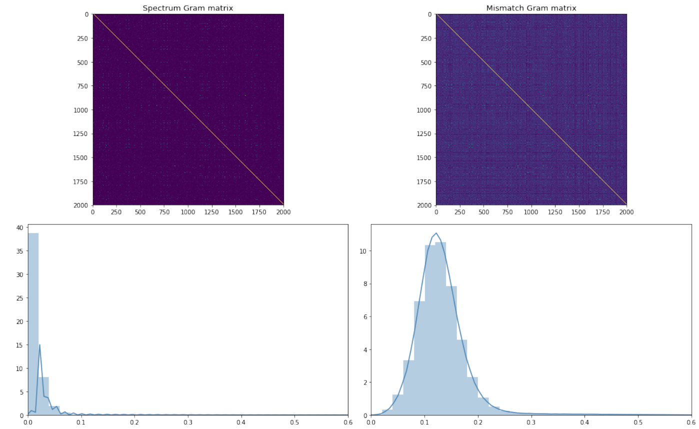

Figure 1. Normalized Gram matrix of Spectrum Kernel with $n = 7$ (top left) and Mismatch Kernel with $n = 7$ and $k = 1$ (top right) on dataset 0 and matrix values distributions (bottom)

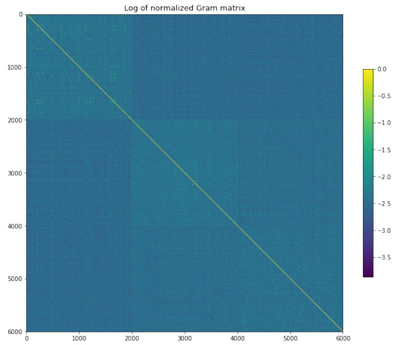

Figure 2. Shift Weighted Degree Kernel $(n = 20, s = 5)$ on dataset concatenation. The 3 prominent square blocks on the diagonal show the specificity of each dataset w.r.t. to each other dataset.

First experiments pointed out a strong overfitting behavior, revealing a lack of generalization power which was imputed to the small size of the dataset for such picky kernel. We hence decided to merge the 3 datasets and compute the joint normalized gram matrix which yielded performances barely competing with Mismatch Kernel.

However, we notice in Figure 2 that dataset essentially match with themselves. Hence, any attempt to concatenate these datasets would actually not be a good idea.

2.2. Biological Sequences - Local Alignment Kernels

Local Alignment (LA) kernels [7] belong to the family of convolution kernels and are well-suited to the comparison of biological sequences: they detect sequence alignments, a widely used principle in bioinformatics. Exploring the use of these kernels

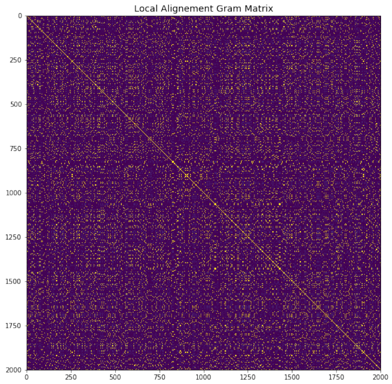

Figure 3. Local Alignment Kernel - Gram matrix of training set of dataset 0. Normalized pairwise terms are either quasi-binary, showing the highly selective behavior of this kernel.

was thus natural to tackle our task of protein classification. Due to the computational complexity of this kernel, we did not tune the hyperparameters of this kernel and chose the article [7] setting: linear gap function with $e = 11$, $d = 1$, penalty coefficient $\beta = \frac{1}{2}$ and BLOSUM62 substitution matrix. In practice, the presence of exponential terms in the dynamic programming formulation of the kernel led to numerical overflows that were handled through the use of the `float128` format, but this trick worsened the computational complexity of the kernel. An example of a LA Gram matrix is available in Figure 3. This figure illustrates the fact that this kernel is "highly selective": the similarity scores it produces are almost boolean. Thus the sequences tend to be greatly discriminated against each other, however this discrimination may not be related to the task of protein classification: overfitting the training set was very easy with a simple classifier (see below), but generalizing to a test set was almost impossible. Another issue encountered with this kernel is the dominance of the diagonal of the Gram matrix which can hamper the use of a SVM-like classifier: even if [7] proposed a spectral translation to alleviate this problem, we observed in practice that the smallest eigenvalue was negative and of great value: the translation it induces leads to retrieve a dominant diagonal.

2.3. Probabilistic Models - Fischer Kernels

Probabilistic modeling of biological sequences is historically older than kernel designs. We tried to explore HMM-based models [3] for protein sequences to estimate the probability of having a sequence to compute a Fischer Kernel. Concretely, we suppose a given sequence to be generated by a hidden state that we wish to estimate with an HMM and we use class posteriors (computed using forward-backward algorithm) to estimate Fischer vectors [3]. We one hot encode characters in the string sequences and suppose to have Gaussian emission probabilities.

2.4. Technical considerations

We parallelized Kernel computations for the Shift Weighted Degree and Local Alignment Kernel on 32 cores Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz.

3. Explored learning algorithms

3.1. Unsupervised learning

We implemented the Kernel-PCA algorithm mainly to allow simple data visulization for a chosen kernel. Example of visualization is available in Figure 4.

3.2. Supervised learning

We implemented the three following supervised learning algorithms: Kernel Logistic Regression, Kernel SVM and Kernel 2SVM. In addition, we implemented a "Multiple Kernel Learner" [5], a type of ensembling kernel method which learns the optimal convex combination of several kernels for a given kernel classifier and a given supervised learning task.

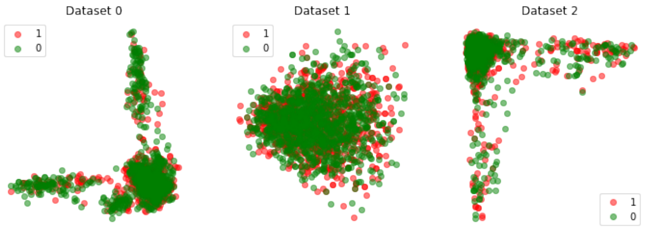

Figure 4. PCA in dimension 2 of the Mismatch kernel with $n = 8$. 2 dimensions may not be enough to visualize the discriminative power of this kernel.

Dataset	0	1	2
Kernel	$(n=8,k=1)$ $+(n=9,k=2)$	$(n=9,k=1)$	$(n=8,k=1)$ $+(n=9,k=2)$
λ	$1.6 \quad 10^{-3}$	$1.1 \quad 10^{-3}$	$8 \quad 10^{-4}$
Accuracy	62.5 ± 0.5	76.7 ± 0.7	65.7 ± 0.5

Table 1. Settings for the 3 datasets. Kernels are all Mismatch kernels of parameters n and k, $+$ indicates simple addition of the Gram Matrices. Accuracies were measured with 10 5-folds cross-validations

4. Model selected and results

Kernels, classifiers and hyperparameters were all selected through cross-validation based on accuracy measures. We found that despite their low-level of refinement, Mismatch Kernels were the best for the classification task at hand. This may illustrate the fact that the size of the data sets available is not big enough to use more complex kernels for statistical use. We also found in practice that the MKL algorithm to combine kernels always led to choosing only one kernel, meaning that this one kernel was outperforming the other kernels for the task at hand; but manually combining kernels led to better validation results: this may indicate that MKL can be prone to overfitting and kernel combination may be seen as a regularization. Finally, the Kernel-2SVM algorithm was observed to be the best learning algorithm for the chosen kernels. Details of choices of kernels, regularization parameter λ of the 2SVM algorithm and cross-validation results are available in table 1.

5. Future Work

We can further improve Probabilistic Model Kernels with Marginalized Kernels [1] (Fisher Kernels are special case of marginalized kernels). Marginalized Kernels also suppose that objects are generated from latent variable models (and use HMMs).

Also, given how relevantly the Local Alignement Kernel manages to match sequences, it's a pitty we were not able to leverage its potential. It would be of great interest to study an eventual relaxation of its formulation, allowing a wider spectrum of similarity measure, which would worsen its intrinsic potential but might help performances on such a small dataset.

Conclusion

Our model consisted of ensembled kernels and helped us achieve a ranking of 10 out of 76 on the public leaderboard, and finished 15th on the private leaderboard. Our model was fully selected with cross-validations and the small gap observed on our ranking between the public and private leadearboard is due to the intrinsic bias of the training set w.r.t. to private test set. From an educational point of view, we implemented various kernels and classifiers and learned a lot from this hands-on practical experience. During this experience, we also focused on designing a efficient and maintainable software.

References

[1] K. Asai, K. Tsuda, and T. Kin. Marginalized kernels for biological sequences. *Bioinformatics*, 18(suppl$_1$) : $S268 - -S275, 072002.$2

[2] A. Cohen, C. S. Leslie, E. Eskin, J. Weston, and W. S. Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476, 01 2004. 1

[3] T. S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*, pages 487–493, Cambridge, MA, USA, 1999. MIT Press. 2

[4] C. Leslie, E. Eskin, and W. Stafford Noble. The spectrum kernel: A string kernel for svm protein classification. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 7:564–75, 02 2002. 1

[5] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008. 2

[6] G. RÃtsch, S. Sonnenburg, and B. SchÃ¶lkopf. Rase: Recognition of alternatively spliced exons in c.elegans. *Bioinformatics (Oxford, England)*, 21 Suppl 1:i369–77, 07 2005. 1

[7] J. Vert, H. Saigo, and T. Akutsu. Local alignment kernels for biological sequences. *Kernel Methods in Computational Biology*, pages 131–154, 01 2004. 1, 2