Classification Experiments of DNA Sequences by Using a Deep Neural Network and Chaos Game Representation

Riccardo Rizzo, Antonino Fiannaca, Massimo La Rosa, Alfonso Urso

Abstract: *Analysis and classification of sequences is one of the key research areas in bioinformatics. The basic tool for sequence analysis is alignment, but there are also other techniques that can be used. Frequency Chaos Game Representation is a technique that builds an image characteristic of the sequence The paper describes the first experiment in the use of a deep neural network for classification of DNA sequences represented as images by using the Frequency Chaos Game Representation.*

Key words: *Deep Neural Networks, DNA sequences Classification, Frequency Chaos Game Sequences Representation.*

INTRODUCTION

The analysis of DNA sequences in bioinformatics is one of the key research area with applications in database search, sequence annotation, gene function prediction and sequence classification. Sequence analysis often needs the comparison of a sequence with some known properties with an unknown one. This comparison is usually obtained using an alignment tool [20].

Sequence representation using features can speed up sequence comparison and similarity search that usually require sequence alignment. The frequency count of nucleotide couples (dinucleotides, e.g. AA, AC, AG, … TT) was already used as a feature for discriminating between class of organisms [14]. This idea can be further expanded considering trinucleotides that are nucleotide triples in the sequence [12], and then generalized to *k*-mers representation [4, 6], where the sequence is represented using a vector of *k*-mer frequency counts. The *k*-mer representation, also called "spectral representation", in recent years has been adopted for sequence classification in [16], where the authors used as classifiers two algorithms such as *k* nearest neighbour (*k*NN) [8] and Support Vector Machine (SVM) [22]. Spectral representation has been also used for sequence classification considering neural gas algorithm [11] and probabilistic topic models [21].

Another sequence representation is the Chaos Game Representation (CGR) [1] where the sequence is used to generate an image that is similar to a fractal image. Starting from the CGR representation it is possible to obtain a matrix that contains the frequency of the *k*-mers extracted from DNA sequences [9]. This representation is called FCGR (Frequency Chaos Game Representation) and is organized as a data matrix. This representation was used for a study on sequence distances functions [24] by constructing some philogenetics trees, that were compared with the one obtained using CLUSTAL-W algorithm. The whole point of the FCGR representation is to obtain a sequence representation that maintains the patterns contained in the sequence and transforms these patterns in image features. The conclusions in [24] are that the image representation techniques are interesting because they seems to produce much more information than the vector ones but there is not yet an effective way to exploit the information in the image. It is still difficult to define an optimal value of *k* or an effective distance function.

CompSysTech'16, 23-24 June 2016, Palermo, Italy
Copyright © 2016 ACM. ISBN 978-1-4503-4182-0/16/06...$15.00
http://dx.doi.org/10.1145/2983468.2983489

In this paper, we are going to investigate what is the reliability of FCGR representation for the classification of genomic sequences. In particular, we chose as classifier a neural network belonging to the deep learning architecture family, that is the convolutional neural network (CNN) [18] [19]. Deep learning networks, in fact, have proved to be very effective for image classification [3].

MATERIAL AND METHODS

In this section, we introduce the proposed approach for classification of DNA sequences. First of all, a dataset of 16S gene sequences belonging to the 3 three most populous phyla of bacteria is presented. Then, we detail the sequence representation based on FCGR and finally we report the classification technique based on deep learning architecture.

Dataset

The proposed approach is tested on a dataset of 3000 16S ribosomal RNA sequences, downloaded from the RDP Ribosomal Database Project II [7], release 10.27. In order to test our classifier with high quality sequences, we only taken into account all the sequences with a length of about 1200-1400 nucleotides, from both uncultured and isolates sources, that have gone through a quality checking by RDP system. At this point, for each one of the three most populous phyla of bacteria, i.e. Actinobacteria, Firmicutes and Proteobacteria, we randomly selected 1000 sequences.

Table 1 reports the dataset taxonomic categories, or taxa, from phylum to genus. It clearly shows that, even if the dataset is balanced at phylum level, it becomes more and more unbalanced for the other taxonomic categories, reaching 393 different groups of 16S sequences at genus level.

	Number of taxonomic categories				
	Phylum	Class	Order	Family	Genus
Actinobacteria	1	1	3	12	79
Firmicutes	1	2	3	19	110
Proteobacteria	1	2	13	34	204

Table 1 - 16S bacteria dataset composition.

Sequence Representation

The FCGR representation technique provides a matrix that contains the frequency of the *k*-mers extracted from DNA sequences; usually, obtained matrices are normalized as

$$\overline{A}^k = \frac{4}{\sum_{i,j} a_{i,j}^k} A^k$$

where a^k_{ij} is the element of the matrix A^k.

The FCGR representation matrix can be used to build a gray scale image that can be considered as a "sequence fingerprint", see the right part of Fig. 1. Using the FCGR representation the image dimensions are function of the dimension *k* of the *k*-mers. The number of representing words (the *k*-mers) will be 4^k and the dimension of the matrix (image) will be $\sqrt{4^k} \times \sqrt{4^k}$.

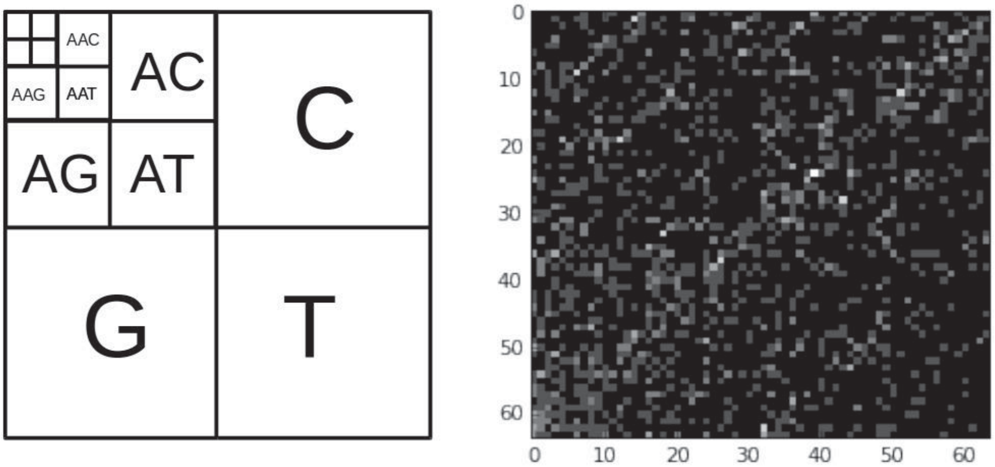

Figure 1 - The construction of the Frequency Chaos Game Representation on the left, on the right the image representing a complete DNA sequence with *k* = 6.

For example if *k* = 6, we have 4^6 = 4096 words and a $\sqrt{4096} \times \sqrt{4096}$, 64×64 matrix (image). Using the FCGR representation a sequence classification problem can be recast as an image classification task. An example of the FCGR output image is showed in Fig. 1; here the image is constituted by gray level pixels, were each of them represents the frequency of a specific *k*-mer. In this image some of the pixels are black because the corresponding *k*-mers are not present in the sequence.

Classifier Architecture

Image classification tasks require the use of highly varying non linear functions that can be obtained from very complex structures as deep neural networks [16]. These neural networks learn from the data very complex functions mapping the input to the output. These complex functions can be implemented by multi-layer neural networks, constituted by many stacked processing layers without intra-layer connections. The label of "deep learning" is used to identify a set of techniques that uses multiple non-linear transformations in order to model high-level abstractions of the input data. The networks based on the convolutional architecture reached very good results in image classification [15], generic visual recognition [10], and other application fields.

In the past the experimental evidence suggested that training deep networks are more difficult than shallow ones, deep networks get stuck in local minima if trained using gradient-based techniques. Convolutional Neural Networks, introduced by Le Cun [19], can be used for image classification tasks and can be constituted by many processing layers. These networks have a reduced set of weights in the convolutional layers, and use a method (the so-called max-pooling) for the reduction of the dimension of the signals from layer to layer. Using one of the available software framework, as Theano [2], implementation of Deep Convolutionary Networks (DCN) can be easily done. In this paper this framework was used in order to develop a DCN according to the architecture shown in Fig. 2, that is a simplification of the original architecture reported by Le Cun [19].

The network is reported in Fig. 2 and is made of three layers: the layer l_0 and the layer l_1 are constituted by a convolutional sub-layer and a max-pooling sub-layer; the third layer l_2 is constituted by a fully connected multi-layer network. This network is constituted by an hidden layer made by 500 units and an output layer with as many neurons as the output categories of the corresponding taxa.

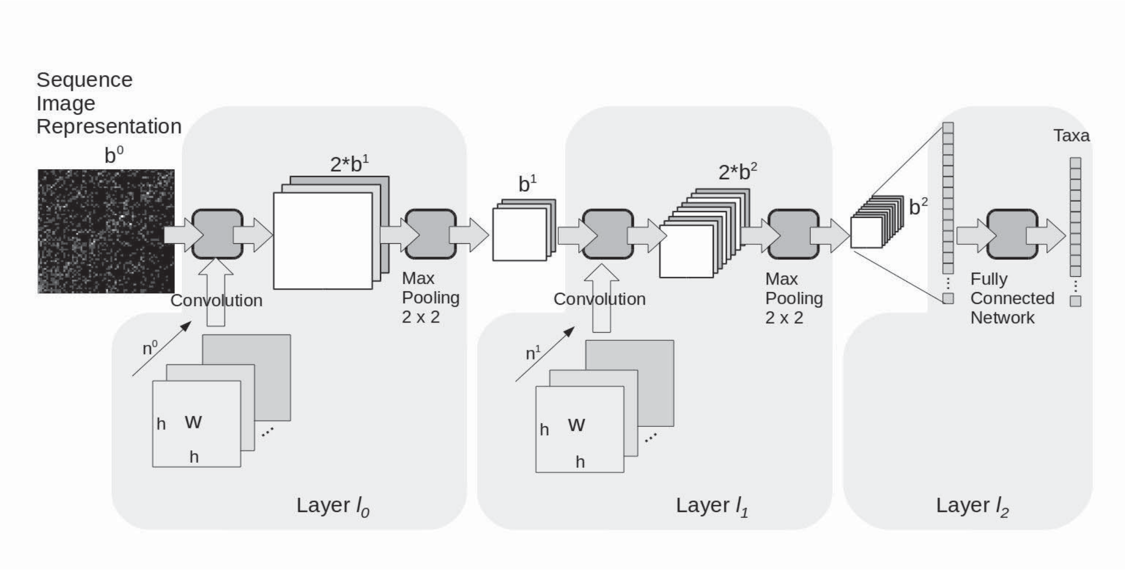

Figure 2 - The Architecture of the DCN network used.

The max-pooling stage is used for dimensionality reduction: for each not overlapping region of 2×2 elements the maximum value is extracted and transmitted to the output.

The two convolutional layers are shaped for square input images, and use n^i square kernels of dimension h and generate a set of n^i output images, for each input image, each of them of dimension $2 \times b^{i+1}$; the new dimension is calculated by using the following equation.

$$b^{(\mathrm{i}+1)} = \frac{b^i - \mathrm{h}+1}{2}$$

For example, assuming that n^0 is 10 , k = 5 so that $b^0 = \sqrt{4^5} = 32$ and h = 5, the l_0 layer produces 10 output images of dimension (32-5+1)/2=14. The layer l_1 will produce $14{*}n^1$ new images.

The output of layer l_1 is reshaped in order to obtain a one dimensional vector that is used as input to the fully connected network that constitutes the l_2 layer. The values of the parameters h, n^i (i = 0, 1) and the number of hidden units in the layer l_2 constitutes the parameters set of the DCN network.

EXPERIMENTAL RESULTS

In order to test the performances of the DCN network as classifier for ECGR images we first trained a DCN network for each taxa, as illustrated in Fig. 3, and then compared results with those obtained using the SVM algorithm. We used the SVM in our comparison because it is often used in sequence classification works [16, 17, 23]. We adopted the SVM implementation in the LibSVM [5] library provided by the WEKA platform [13].

As regards the DCN network, we used the following parameters h=5 for both convolutional layers, 10 convolutional filters in the l_0 layer and 20 in the l_1 layer (n^0=10, n^1=20), 500 units in the hidden layer of the multilayer perceptron in layer l_2. These values are chosen with analogy to the one used in the LeCun paper [19].

All the results are obtained using the ten-fold technique: in each fold, the DCN network is trained using the 90% of the input patterns and tested using the remaining 10%. We perform two different comparison tests, with both full-length and 500 bp fragments DNA sequences, because often the whole sequence is not available, and the representation must be obtained from a shorter sequence fragment.

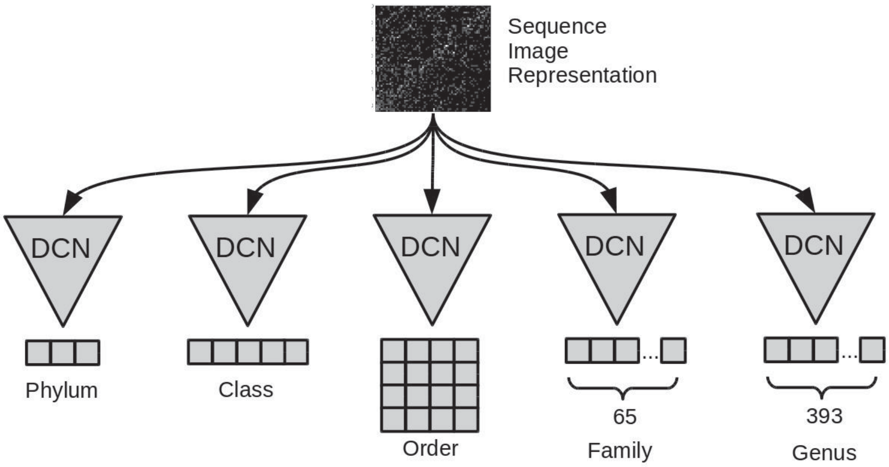

Figure 3 - The architecture of the classifier.

Results are reported in Table 1 for the full-length DNA sequence; the classification is obtained for the same sequence with representation images, calculated using *k* = 5, 6, and 7. In other works that use the feature representation better results are obtained with a longer representation word (i.e. *k* = 7). The results in Table 2 are not aligned with the results of other works that uses the feature representation. In these works the classification results improve if the length of the representing words (i.e. *k* value) increase. In Table 1 the results are only slightly better, but this probably does not compensate the increased computational cost, considering that the representing images are 4 times larger along all the stages of the DCN network.

***k* Parameter**	**Method**	**Phylum**	**Class**	**Order**	**Family**	**Genus**
5	Proposed	0.999	0.998	0.985	0.969	0.888
	SVM	0.997	0.995	0.987	0.975	0.901
6	Proposed	**0.999**	**0.998**	**0.989**	**0.982**	0.907
	SVM	0.997	0.994	0.977	0.946	0.715
7	Proposed	0.994	0.996	0.987	0.980	**0.908**
	SVM	0.997	0.995	0.930	0.822	0.541

Table 2 - Accuracy scores for full-length sequences. Proposed method Vs. SVM.

***k* Parameter**	**Method**	**Phylum**	**Class**	**Order**	**Family**	**Genus**
5	Proposed	0.962	0.947	0.701	0.471	0.315
	SVM	0.562	0.569	0.594	0.059	0.050
6	Proposed	**0.995**	**0.992**	**0.896**	**0.81**	0.554
	SVM	0.992	0.991	0.770	0.433	0.312
7	Proposed	0.992	0.974	0.870	0.744	**0.556**
	SVM	0.842	0.825	0.694	0.440	0.320

Table 3 - Accuracy scores for 500bp-length sequences. Proposed method Vs. SVM.

Table 3 reports the classification results obtained with a fragment of the sequence. The results reported in the tables show the comparison with an SVM classifier: the proposed method outperforms the SVM in both cases, full-length sequences and 500 bp fragments. In the last case the representing images lacks of many details and the classification results are worst; at the same time the improvements with the increasing of *k* values are much evident.

CONCLUSIONS AND FUTURE WORK

The paper proposed a new method for DNA sequence classification by using the FCGR images and deep learning convolutional network. The obtained results are very good, especially for the full length sequences that are recognized with a very high accuracy. The results obtained with the 500bp sequences should be improved and the method needs more investigation because the results drop abruptly if the classification is made using sequence fragments. The DCN implements the discussed highly non linear functions that can recognize crucial details in an image with "random" gray scale pixels and assign the correct label. Currently we are investigating different architectures for the DCN network in order to improve the results obtained with the sequence fragments.

REFERENCES

[1] Almeida, J.S. et al. 2001. Analysis of genomic sequences by Chaos Game Representation. *Bioinformatics*. 17, 5 (2001), 429–437.

[2] Bastien, F. et al. 2012. Theano: new features and speed improvements. *NIPS 2012 deep learning workshop* (2012).

[3] Bengio, Y. 2009. Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*. 2, 1 (2009), 1–127.

[4] Blaisdell, B.E. 1986. A measure of the similarity of sets of sequences not requiring sequence alignment. *Proceedings of the National Academy of Sciences*. 83, 14 (1986), 5155–5159.

[5] Chang, C.-C. and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*. 2, 3 (Apr. 2011), 1–27.

[6] Chor, B. et al. 2009. Genomic DNA k-mer spectra: models and modalities. *Genome biology*. 10, 10 (Jan. 2009), R108.

[7] Cole, J.R. et al. 2009. The Ribosomal Database Project: improved alignments and new tools for rRNA analysis. *Nucleic acids research*. 37, Database issue (Jan. 2009), D141–5.

[8] Cover, T. and Hart, P. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*. 13, 1 (1967).

[9] Deschavanne, P.J. et al. 1999. Genomic signature: characterization and classification of species assessed by chaos game representation of sequences. *Molecular biology and evolution*. 16, 10 (1999), 1391–1399.

[10] Donahue, J. et al. 2014. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *Proceedings of The 31st International Conference on Machine Learning* (2014), 647–655.

[11] Fiannaca, A. et al. 2015. A k-mer-based barcode DNA classification methodology based on spectral representation and a neural gas network. *Artificial Intelligence in Medicine*. 64, 3 (Jul. 2015), 173–184.

[12] Goldman, N. 1993. Nucleotide, dinucleotide and trinucleotide frequencies explain patterns observed in chaos game representations of DNA sequences. *Nucleic Acids Research*. 21, 10 (1993), 2487–2491.

[13] Hall, M. et al. 2009. The WEKA data mining software. *ACM SIGKDD Explorations Newsletter*. 11, 1 (Nov. 2009), 10–18.

[14] Karlin, S. and Burge, C. 1995. Dinucleotide relative abundance extremes: a

genomic signature. *Trends in genetics*. 11, 7 (1995), 283–290.

[15] Krizhevsky, A. et al. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25*. P. Bartlett et al., eds. 1106–1114.

[16] Kuksa, P. and Pavlovic, V. 2009. Efficient alignment-free DNA barcode analytics. *BMC Bioinformatics*. 10, Suppl.14 (Jan. 2009), S9.

[17] Kuksa, P. and Pavlovic, V. 2007. Fast Kernel Methods for SVM Sequence Classifiers. *Algorithms in Bioinformatics* (Berlin, Heidelberg, 2007), 228–239.

[18] LeCun, Y. et al. 2015. Deep learning. *Nature*. 521, 7553 (2015), 436–444.

[19] Lecun, Y. et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 86, 11 (1998), 2278–2324.

[20] Nei, M. and Kumar, M.D. 2000. *Molecular Evolution and Phylogenetics*. Oxford University Press.

[21] La Rosa, M. et al. 2015. Probabilistic topic modeling for the analysis and classification of genomic sequences. *BMC Bioinformatics*. 16, Suppl 6 (2015), S2.

[22] Scholkopf, B. and Smola, A.J. 2002. *Learning with kernels*. MIT Press.

[23] Seo, T.-K. 2010. Classification of nucleotide sequences using support vector machines. *Journal of molecular evolution*. 71, 4 (Oct. 2010), 250–67.

[24] Wang, Y. et al. 2005. The spectrum of genomic signatures: from dinucleotides to chaos game representation. *Gene*. 346, (2005), 173–185.

ABOUT THE AUTHORS

Riccardo Rizzo, Research Scientist, ICAR-CNR, National Research Council of Italy, Via Ugo La Malfa 153, Palermo, Italy. E-mail: ricrizzo@pa.icar.cnr.it

Dr. Antonino Fiannaca, Research Scientist, ICAR-CNR, National Research Council of Italy, Via Ugo La Malfa 153, Palermo, Italy. E-mail: fiannaca@pa.icar.cnr.it

Dr. Massimo La Rosa, Research Scientist, ICAR-CNR, National Research Council of Italy, Via Ugo La Malfa 153, Palermo, Italy. E-mail: larosa@pa.icar.cnr.it

Dr. Alfonso Urso, Research Scientist, ICAR-CNR, National Research Council of Italy, Via Ugo La Malfa 153, Palermo, Italy. E-mail: urso@pa.icar.cnr.it