Enriched DNA Strands Classification using CGR Images and Convolutional Neural Network

Sarah Safoury
Software Engineering Department
Faculty of Informatics and Computer Science
The British University in Egypt
sarah.safoury@bue.edu.eg

Walid Hussein
Computer Science Department
Faculty of Informatics and Computer Science
The British University in Egypt
walid.hussein@bue.edu.eg

ABSTRACT
Bioinformatics is the biological study which applies programming techniques for more understanding and analysis of living objects such as the study of genome structure. The genome structure could be represented in the form of an image. Chaos Game Representation (CGR) is the practice of converting the DNA sequence (i.e., genomes) into images, where each image is a graphical appearance for an individual DNA strand's signature. CGR is a method of converting a long one-dimensional DNA sequence into a graphical form. This method provides a visual image of a DNA sequence different from the traditional manual linear arrangement of nucleotides polymerase chain reaction. In the recent years, CGR was introduced to automatically classify genomes not only by archival references but also through its' unique signature. In this paper, a novel CGR classification approach is developed combining the advances of image processing and pattern recognition approaches. The approach starts by declaring the genome and using the CGR technique to map it to the graphical interface (i.e., 16x16 signature images). Then, an image processing procedure is prepared to handle complex geometric shapes, analyze structured and visualized genome sequences and fractal point the included nucleotides of these images. Finally, the convolutional neural network was designed and well-trained by those signatures to classify each genome tested.

CCS Concepts
Applied computing →Molecular sequence analysis

Keywords
Image Processing; Genomic Signature; Chaos Game Representation (CGR); Convolutional Neural Network (CNN), DNA Sequence

1. INTRODUCTION
Despite various number of researches that was done in the field of CGR recently, still no applications have yet been published. However, the idea is not impossible to embrace, thus here I come trying to develop a scientific tool that uses this scientific bioinformatics theory to make an application that reads a part or whole DNA strand and use fractals which is a mathematical calculated equation that transfers the genome into an image to be shown. This image would have a signature that represents similar signatures to the signatures resulting from the same family. This image would be a four to the power 4 giving 256 pixels composed of the 4 nucleotides A, C, G and T. Letter A stands for 'Adenine', C for Cytosine', G for 'Guanine' and T for 'Thymine'. In the science of Biology of DNA each DNA strand has its' complementary. This mean that each of those nucleotides has its' complementary, where A complements T and vice versa and C complements G and vice versa as shown below in Figure 1. After experiencing many DNA samples with their images, the entire data is collected, clustered and classified according to the collected features. Furthermore, a testing set is prepared without its' class name would be applied on it the image processing and state to which family does it belongs. In this project there would be eleven families or classes each having nineteen, twenty or twenty-one different genomes. Those generated images of each family would be then optimized by using an artificial neural network called 'convolutional neural network (CNN)' that enhances the 'SoftMax' function giving optimized images and signatures resulting to a useful and light weighed dataset with high accuracy rate of classification and distinguished fingerprints.

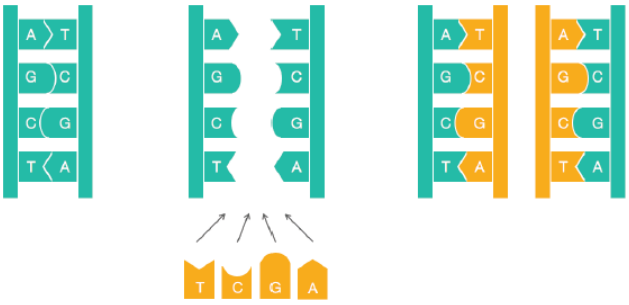

Figure 1. DNA nucleotides Adenine bonds with Thymine and Guanine bonds with Cytosine (The Complementary Rule)

Bioinformatics is the science of collecting and analyzing complex biological data such as genetic codes, it combines computer science, mathematics and biological data in its' greatest scale. Bioinformatics requires computer-programming methodology especially in the partition of 'genomics' which would be the case in this paper. Bioinformatics gives you the identification of genomes that gives the ability to understand the basis of diseases or the difference between families and others in a bigger scope. This genome could be expressed as an image and be used in extraction of important data from a big dataset. Bioinformatics tools help you to find hidden messages within DNA genomes and the repetition of a strand and it's complementary and patterns the data to classify all related patterns in one family for any input to be categorized and classified easily.

ICBBS 2019, October 23–25, 2019, Beijing, China

ACM ISBN 978-1-4503-7251-0/19/10…$15.00
DOI:https://doi.org/10.1145/3369166.3369176

At early age typically by the 70s Paulien Hogeweg and Ben Hesper introduced bioinformatics but in a different scope that was meant to refer to the information processes in biotic systems that meant to categorize the bioinformatics in two branches biophysics or biochemistry. Sooner in the 1980 to 1991 bioinformatics became what we are studying nowadays including the foundation of it. In the addition to pattern recognition, data mining, machine learning algorithms, as well as various researches in sequence alignment, perdition of gene expression, gene finding and a lot more studies that must be kept evolving as it adds to the science of bioinformatics and widen the scope of it.

From the common activities in bioinformatics is the CGR [1]. Chaos game representation is the mathematical means of 'Fractal'. Fractal is done by iteratively pointing random points inside a polygon and calculation of the distance between each of the points to result for a fractal shape that holds a definite meaning. This fractal shape could be considered as an image or to be more precise the shape appearing would hold a signature of a genome or DNA strand [2], [3]. This signature is uniquely identified by the originality of the extraction of this exact genome that was taken from and classify each of the images by the same concept given their sequence of DNA as shown in Figure 2.

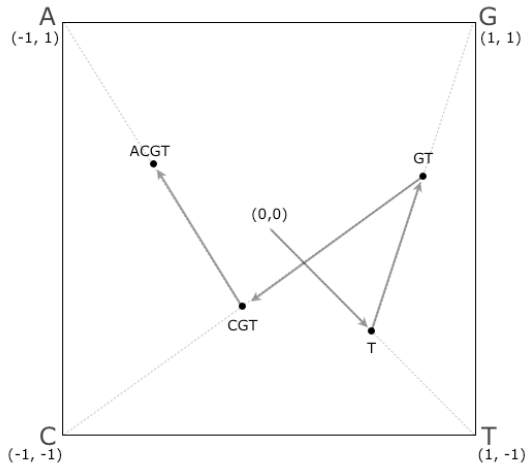

Figure 2. The result of the CGR on Four Points representing x-axis, y-axis, positions on x-axis and y-axis

Furthermore, DNA structure similarity is high (i.e., above 50% matching) between different species, for example, human share 60% of their protein (genome structure) with banana, while it is 98.8% with apes [4], [5]. Therefore, the genomic structure-based classification of families and species is considered a very challenging problem. Additionally, the collection of one proper dataset of a single species takes years as evolution science day-by-day changes and chemicals around the world affects the biological structure in a way that could cause the clean DNA structure a deformation.

In this paper, the advances in the convolutional neural network (CNN) are adopted to aid in resolving the above-mentioned challenges and to efficiently classify genomes of 11 families. This developed CNN is based on CGR technique that creates a unique fingerprint (i.e., CGR image) for each genome, with a level of similarity between the created images of the same family, and a noticeable variation from other families.

2. MATERIAL AND METHODS

2.1 Genome to Image Conversion Algorithm

The proposed approach is converting a DNA strand from a sequence form to an image form. Hence, this graphical representation is unique for each corresponding DNA strand that maps to an individual sample of a family. Therefore, all genomes from each family of the dataset are converted into 16x16 sized images, that are composed of A, T, C and G nucleotides. Each of the nucleotide would be marked on one of the four edges on a square shape starting from the origin (0, 0) and continue using Markov Model [6], [7] to underline a pattern to graph the genome as shown in Figure 3. Every repeated pixel would be given a color to distinguish its repetition. K-mer is the number of pixel division to number of internal squares, if k-mer = 4 this means that 4^4= 256 pixels image each pixel defining a specific oligomer which is a DNA sequence like: ACCT and given its color indicating the number of repetitions of this oligomer as shown in Figure 3. The color of the images would be grey scaled between 0 to 255. Given the resulting table which is called the FCGR (frequency matrices extracted from chaos game representation) as shown in Figure 4.

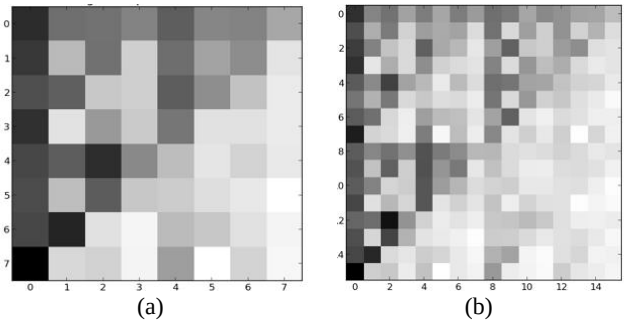

Figure 3. CGR representation of a DNA sequence (oligomers) in k-mer where (a) k=3, and (b) k=4

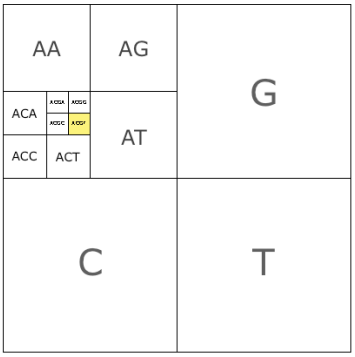

Figure 4. CGR Algorithm of how the quadrants are pointed in representing a genome structure

To schematically explain the CGR algorithm [8], a quadrant of a sample CGR image is marked in Figure 5 as a sub-sequent that shows what each pixel holds an Oligomer. Some pixels such as (CCC) that appears on the bottom – left corner has the darkest intensity. This is due to the occurrence of this oligomer repetitively in the analysis of this genome. The brightest intensity pixel (CTG) is due to its rare occurrence in the whole genome. In the same manner, all of the other oligomers are shown in different intensities pixels correspond to their level of occurrence in the genome.

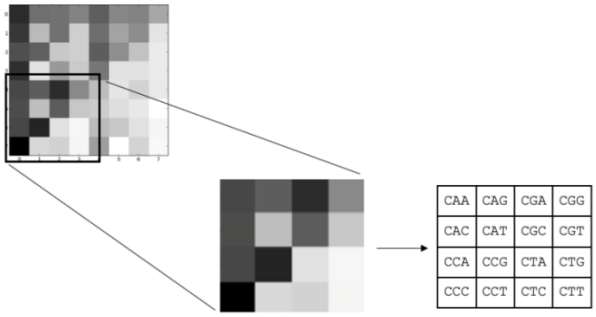

Figure 5. CGR image representation for a single quadrant (in a k-mer, where k=3) and the oligomer it holds. Each pixel carries a color which relevant to how frequent an oligomer occurred in the sequence

2.2 Formation of the CGR Image

The technique of forming a CGR image starts by plotting a square and diving it into four pixels, which is considered as a 1-mer division. Each pixel holds one of the four nucleotides that are kept consistent throughout the whole process. Let the top left square holds 'A', top right square holds 'G', bottom left holds 'C' and bottom right holds 'T'. Consequently, each of these pixels is divided into another inner four pixels that would be considered as a 2-mer division. Consistency is kept, as by dividing the next squares A, G, C and T would be still located at their exact previously stated position as was clarified in Figure 4.

2.3 Building the Classification Model

The produced dataset of 16x16 CGR images are divided into training and testing sets. The training dataset composes all the necessary genome characteristics (relevant features), to be utilized in building a robust classification model.

The classification model which will be designed in the proposed method is the Convolutional Neural Network (CNN). The process of developing a CNN starts by extracting feature maps from its input image (i.e., our CGR image), which defines a convolutional layer. Afterwards, the formed feature maps undergo a linear scaling step using a Rectified Linear Unit (ReLU) function. The ReLU function sets all negative feature maps values to zero while (i.e., zero-padding), all the other values are kept as they are. Consequently, the feature maps become ready for the pooling step, the purpose of which is providing our CNN with the facility of

"spatial invariance" which does the shrinking, commonly to the half of the feature map size to prepare it before being inserted into an Artificial Neural Network (ANN). Throughout this entire process, the networks parameters (weights and bias) are continually updated with the objective of minimizing the classification error for the training dataset. The CNN steps are schematically shown in Figure (6), and more details about the implementation and results of CNN in classification tasks can be found in [9]-[16].

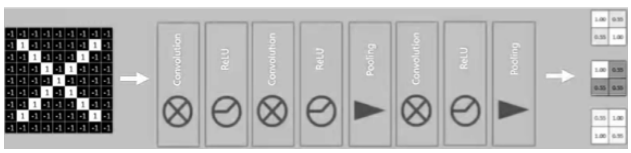

Figure 6. All stages of layers from Convolutional Layer, Rectification Layer and Pooling Layer of Convolutional Neural Network as Deep Learning of combined features to create a Model

3. EXPERIMENTAL RESULTS

3.1 Dataset Collection

The dataset used in this paper was collected by the NCBI ref-genome/ re-seq® [20] as a clean dataset. It included eleven families where each family contains [19-21] different genome samples. Those eleven Families are: Amphibians, Ascomycetes, Basidiomycetes, Bird, Fish, Flatworms, Insect, Land Plants, Mammals, Reptiles and Round Worms. Meanwhile, this dataset is randomly divided into 70% for training and 30% for testing.

3.2 CGR Images Classification: The Literature Results

The summarization of the literature results for CGR images classification obtained by Probabilistic Neural Network (PNN) are shown in Table 1 [17-19]. On the other hand, the average classification accuracy using the Artificial Neural Network (ANN) for each class are shown in Table 2.

However, CNNs' accuracy rate is chosen over other NNs as CNNs can automatically extract feature from the image. While other models with pixel vector choose a lot of spatial interaction between pixels however, important features are usually lost, a CNN effectively uses adjacent pixel information to effectively down sample the image first by convolution and then uses a prediction layer at the end.

This concept was first presented by Yann le cun in 1998 for digit classification where the scientist used a single convolution layer. It was later popularized by Alexnet in 2012 which used multiple convolution layers to achieve state of the art on ImageNet. Thus, making them an algorithm of choice for image classification challenges henceforth, more details can be found on [21], [22].

Table 1. The maximum accuracy in CGR images classifications using the Probabilistic Neural Network (PNN) in each of the nine families [17-19].

Class	Accuracy obtained (%)
Plant	73.3
Fungi	65.38
Cnidaria	94.12
Porifera	80.0
Acoelomata	85.71
Pseudocoelomata	60.0
Protostomia	86.7
Vertebrata	96.8
Average	**92.3**

Table 2. Average Accuracy of CGR images classification obtained by ANN (Artificial Neural Network) [17]-[19].

No. of hidden layers	No. of neurons in each layer	Average accuracy (%)	Training time (min: sec)
1	20	44.78	0:06
1	40	39.76	0:23
1	60	39.48	1:04

1	80	32.43	1:19
2	[20 10]	64.18	0:13
2	[25 15]	73.13	0:23
2	[30 15]	70.01	0:27
2	[50 25]	55.9	1:15
2	[50 30]	44.09	2:13
3	[20 15 10]	69.2	0:20
3	[25 20 15]	69.5	0:40
3	[30 15 5]	70.15	0:44
3	[30 20 10]	66.3	0:58
3	[30 25 20]	55.8	1:30

3.3 CGR Images Classification: The Proposed Method Results

As specified above, the data set was divided into 70% training set and 30% testing set. The training set ran into a CNN of 2000 epochs that shows the train loss, validation loss, the accuracy and the duration as shown in Figure 7.

Epochs were chosen to be 2000 because it achieved the minimum classification error, and to avoid the over fitting issue.

The training loss was monitored throughout all epochs to make sure it is converging.

The distribution of the training loss with the number of epochs is shown in the first 2x2 graphs of Figure 7.
Also, the validation loss variation with respect to the training loss in Figure 7 represents that the training loss results are greater than the validation loss in all specified epochs proving that overfitting and underfitting of the network was avoided.

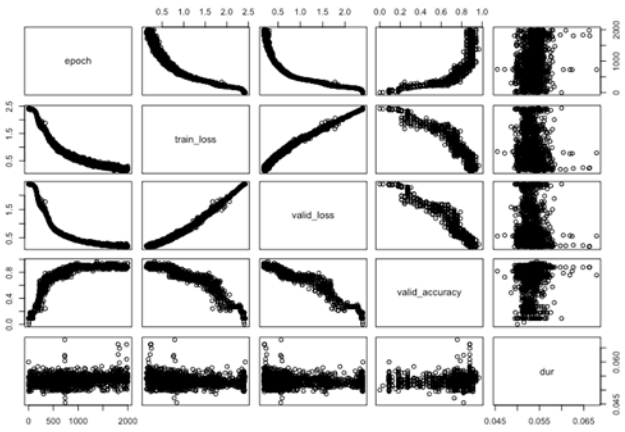

Figure 7. Pairwise parametric studies applied on the validation data for the relative variation of the critical analysis characteristics: Epoch, Training loss, Validation loss, Validation accuracy, and Duration

The results of the classification of the eleven families are shown in Table 3 throughout the calculated the precision, recall, f1-score and support.

The precision shows the percentage of prediction class with relation to the actual class, pointing out to the precision of Basidiomycetes family of only 57% and this happened as a result of diversity of its categories whether white rot or brown rot wooden decay fungi which happen to affect the data accuracy rate.

The Recall shows the true precision over the true precision added to the false negatives which eventually indicates all positive samples.

f1-score is the mean of precision and recall where it reaches its best at 1 and worst at 0.

The last column shows the support where the number of occurrences of all 10 families to be 6 as each family had 20 genomes and only 30% of it is tested (20*30/100) except Land plants as the genomes were total of 19 and 30% of it was being tested which leads for the support of only 5.

The last row in the below table shows the average accuracy of 87% of the classification happened to give the exact classification and the predicted matched the actual as shown in the following table.

Table 3. Convolutional Neural Network Accuracy results of the 11 families (momentum rate= 0.9, learning rate= 0.01)

	Precision	Recall	F1-score	Support
INSECT	1	0.83	0.91	6
Roundworms	0.8	0.67	0.73	6
MAMELS	1	1	1	6
Reptiles	0.71	0.83	0.77	6
Flatworms	1	0.83	0.77	6
Basidiomycetes	0.57	0.57	0.62	6
Land Plants	0.83	1	0.91	5
BIRD	0.86	1	0.92	6
Amphibians	1	0.5	0.67	6
FISH	1	1	1	6
Ascomycetes	0.75	1	0.86	6
Average	0.87	0.85	0.85	6

The accuracy results show a 93% of a maximum accuracy occurring in all the epochs.

4. DISCUSSION

Regarding the dataset some problems were faced, that the size of the data set was high since it holds eleven families each has from 19 to 21 genomes, which means almost 660 pictures generated for each k-mer (i.e., 660 images for 3-mer CGR representation and another 660 images for 4-mer CGR representation).

Since each k-mer needs an image to be created of each genome, only 3-mer and 4-mer were developed in this project. However, this does not help in solving the problem, as the size of the picture itself is big enough to cause runtime error.

The model results and accuracy depend on the size of the input dataset. As a result, all images were chosen to follow a 16x16 grid images, as a result the images size would be lighter on display. The CNN is used to optimize the images by using a function inside this ANN called 'SoftMax' function. Since CNN has many layers of three types, the previously mentioned function is not beneficious in the last type of layer which is the "fully connected layer".

In CNN any 2D or 3D data like sounds, images, text …etc. satisfies the needs of such technique. Which means it only captures local "Spatial" patterns in data, in cases data cannot be made to look like an image Conventional network is less useful.

CGR and FCGR were used to overcome such a interruption by converting the sequence into a 2D image and using this neural network to result an optimality of images of all provided dataset from the 11 family genomes.

Searching for a dataset of high quality to use it as a training set to maintain an ultimate accuracy results for the testing set took a lot of time. The effort and time spent to find the dataset clean from being corrupted and not accurate was high.

The dataset chosen in this paper is the same dataset used by previous scientists who all experiments and theories depend upon it. As it appears that this dataset is free from genomes that could be harmed, or carrier of such diseases that would cause a deformation to the DNA.

Computational tool is developed to implement the proposed method and applied being able to connect the scientific tool with the implemented code of this project that results for the classification and optimization of output. The code was written in Python and the scientific tool was implemented using markup languages: HTML, CSS and JavaScript.

The scientific tool required to use Django to connect the code being implemented for classification of genomes.

As a result, the 30% of the dataset that was taken as the testing set would be listed in the samples of the scientific tool provided for users to be able to test on their own and get classification reports.

5. CONCLUSION

This paper holds scientific facts in biology about the DNA and the genomic structure held. The DNA structure is consistent in every living cell; however, the combination of the A, C, G and T have different permutation positions. This work provided classification of species by a method called Chaos Game Representation through a given samples of genomes. The genome is converted to a 2D image, which forms a signature of each genome, however the signatures of same species would be somehow correlated. Some genomic sequences could be repeated in a DNA strand that could be given a frequency to form what is called an FCGR. Those genomes frequencies could be viewed in the form of an CGR image, which is the art of converting DNA strands into images. These images are found to be fingerprints where every single genome has its unique signature.

A CNN was developed to classify the collected 16x16 CGR images to their associated families. Hence, the result provided by the classification of the convolutional network of the tested genomes had an average accuracy of 87%. This accuracy percentage is a superior percentage in comparison to the commonly used neural networks.

6. REFERENCES

[1] H. Joel Jeffrey, "Chaos game representation of gene structure", Nucleic Acids Research, Vol. 18, No. 8, pp: 2163-2170,1990.

[2] Nick Goldman, "Nucleotide, dinucleotide and trinucleotidefrequencies explain patterns observed in chaos game representations of DNA sequences", Nucleic Acids Research, 1993, Vol. 21, No. 10, pp: 2487-2491, 1993.

[3] Almeida J S, Joao A. Carrico, Antonio Maretzek, Peter A.Noble and Madilyn Fletcher, "Analysis of genomic sequences by Chaos Game Representation", BIOINFORMATICS Vol. 17,no. 5, Pages 429-437, 2001.

[4] Simpson, G. G. in Classification and Human Evolution (ed Washburn, S. L.) (Aldine, Chicago, 1963).

[5] Dobzhansky Th Ayala, F. J., Stebbins, G. L. & Valentine, J. W. Evolution (Freeman, San Francisco, 1977).

[6] Almagor (1983) A Markov analysis of DNA sequences, J. Theor. Biol. 104: 633—645

[7] Zanoguera and de Francesco, "Protein classification into domains of life using Markov Chain Models", CSB 2004, The Computational Systems Bioinformatics Conference, IEEE, pp. 517-519, 16-19 Aug. 2004.

[8] Almeida JS, Carrico JA, Maretzek A, Noble PA, Fletcher M. Analysis of genomic sequences by ChaosGameRepresentation. Bioinformatics. 2001;17(5):429–37.doi: 10.1093/bioinformatics/17.5.429.

[9] Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. NIPS'12 Proceedings of the 25th international conference on neural information processing systems, 1, 1097–1105.

[10] Kawano, Y., & Yanai, K. (2014). Food image recognition with deep convolutional features. ACM UbiComp workshop on cooking and eating activities.

[11] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv Preprint, arXiv, 1312–6229.

[12] Yigit, O. G., & Ozyildirim, B. M. (2017). Comparison of convolutional neural network models for food image classification. 2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Gdynia, 2017, pp. 349–353.

[13] Bell, S., & Bala, K. (2015). Learning visual similarity for product design with convolutional neural networks. ACM Transactions on Graphics, 34(4), 98–107.

[14] Chatfield, K., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. arXiv 1405.3531.

[15] Chellapilla, K., & Puri, S., & Simard, P. (2006). High performance convolutional neural networks for document processing. In Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition. Los Alamitos, CA: IEEE Computer Society.

[16] Ciresan, D. C., Meier, U., Masci, J., Maria Gambardella, L., & Schmidhuber, J. (2011) Flexible, high performance convolutional neural networks for image classification. In Proceedings of the International Joint Conference on Artificial Intelligence (vol. 1, pp. 1237–1242). Menlo Park, CA: AAAI Press.

[17] Karthika Vijayan, Vrinda V. Nair, Deepa P. Gopinath, 10th National Conference on Technological Trends (NCTT09) 6-7 Nov 2009, Classification of Organisms using Fractal-Chaos Game Representation of Genome Sequences and ANN.

[18] Ahmed, A., Yu, K., Xu, W., Gong, Y., & Xing, E. (2008). Training hierarchical feed-forward visual recognition models

using transfer learning from pseudo-tasks. In Proceedings of the European Conference on Computer Vision (pp. 69–82). Berlin: Springer.

[19] Sandberg R., Winberg G., Branden C. I., Kaske A., Ernberg I. and Coster, “Capturing Whole - Genome characteristics in short sequences using a naive Bayesian classifer”, Genome Res., Vol. 11, pp. 1404-09, May 2001.

[20] https://www.ncbi.nlm.nih.gov/refseq/

[21] I. Goodfellow, et al., Deep Learning. MIT Press, 2016.

[22] LeCun, Yann, Bottou, L ́eon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.