Prediction of normalized signal strength on DNA-sequencing microarrays by n-grams within a neural network model

Charles Chilaka¹,⁵, Steven Carr²,³,*, Nabil Shalaby³,⁴, Wolfgang Banzhaf³,⁶

¹Program in Scientific Computing; ²Department of Biology; ³Department of Computer Science; ⁴Department of Mathematics and Statistics, Memorial University of Newfoundland and St. John’s, Newfoundland, Canada A1C 5S7; ⁵Department of Mathematics, FUT, Owerri, Nigeria; ⁶Present address: Department of Computer Science and Engineering, Michigan State University, East Lansing MI 48824; Steve Carr - Phone: +1 (709) 764 4776 office; E-mail: scarr@mun.ca; *Corresponding author

Received November 25, 2018; Accepted December 1, 2018; Published May 30, 2019

DOI: 10.6026/97320630015388

Abstract: We have shown previously that a feed-forward, back propagation neural network model based on composite n-grams can predict normalized signal strengths of a microarray based DNA sequencing experiment. The microarray comprises a 4xN set of 25-base single-stranded DNA molecule (“oligos”), specific for each of the four possible bases (A, C, G, or T) for Ade-nine, Cytosine, Guanine and Thymine respectively at each of N positions in the experimental DNA. Strength of binding between reference oligos and experimental DNA varies according to base complementarity and the strongest signal in any quartet should “call the base” at that position. Variation in base composition of and (or) order within oligos can affect accuracy and (or) confidence of base calls. To evaluate the effect of order, we present oligos as n-gram neural input vectors of degree 3 and measure their performance. Microarray signal intensity data were divided into training, validation and testing sets. Regression values obtained were >99.80% overall with very low mean square errors that transform to high best validation performance values. Pattern recognition results showed high percentage confusion matrix values along the diagonal and receiver operating characteristic curves were clustered in the upper left corner, both indices of good predictive performance. Higher order n-grams are expected to produce even better predictions.

Keywords: Neural networks, n-grams, Performance, Regression values, Confusion matrix, Receiver Operating Characteristic curves.

Background: DNA sequences are strings of hundreds to millions of four nitrogenous bases (Adenine, Cytosine, Guanine and Thymine) represented by the letters A,C,G, and T respectively. Representation of these strings as numerical values enables the application of powerful digital signal processing techniques. Desirable properties of a DNA numerical representations and some examples are given in [3, 15]. N-gram method was first introduced by C.E Shannon in 1948 [9]. Neural network learning methods provide a robust approach to approximation of real-valued, discrete-valued and vector-valued target functions, [12] such as numerical DNA data. The study of artificial neural networks has been inspired by the observation that biological learning systems are built of very complex webs of interconnected neurons, [10, 11, 12], which communicate through a large set of interconnections assigned variable strengths (weights) in which the learned information is stored, [13]. Each neuron computes a weighted sum of its y input signals. The activation function for neurons is the sigmoid function defined in [12] as σ(y) = 1/1+e⁻ʸ. where y is the weighted sum of the inputs. The output of the sigmoid function ranges from 0 to 1, increasing monotonically with its input and the weights of the interconnections between the different neurons are adjusted during the training process to achieve a desired input/output mapping. Artificial Neural Networks studies have been used for computational analysis of human DNA sequence [14], single base pair discrimination of terminal mismatches [16], biological phenomena through computational intelligence [17], human donor
and acceptor sites prediction [18], coding region recognition and gene identification [19], predicting transmembrane domains of proteins [20] and the prediction of nucleotide sequences using genomic signals [21], [22]. We previously [15] used mainly 1-gram, 2-gram and their composition to predict normalized signal strengths from a DNA-sequencing microarray. Here, numerical data from an Affymetrix [1] DNA re-sequencing experiment are normalized and partitioned into training, testing and validation set within a Matlab [2] neural network with 4, 16, 64 for mono-, di-, and tri-nucleotide strings respectively in the input layer. The influence of the length of oligo nucleotide in the nucleotide hybridization intensity experiments [23] is examined by replacement of mono-, di- and tri nucleotide strings with their respective n-gram equivalents. The n-gram ratios for mono-, di- and tri nucleotides are shown in Table 1, Table 2 and Table 3 respectively. The results with 1-gram and 2-gram and their composition have been discussed in [15]. We advance the results obtained previously by examining the influence of 3-grams on overall performance of our predictions based on the data evaluation functions. We examine the effect of the different number of neurons in the hidden layer on optimal prediction performance. The output node layer has 4 nodes reflecting our choice of sequence signals to predict. The schematics of DNA neural network architecture are shown in Figure 1. The DNA sequence data are first converted by a sequence encoding schema into neural network input vectors (ratios of n-gram). The neural network then predicts those normalized intensities according to the sequence information embedded in the neural interconnections after network training.

**Data evaluation functions:**
In [15], we explained the concept of performance and regression values. We also examined their results using 1-gram, 2-gram and their composition. We now check for consistency of the results with the inclusion of 3-gram using two other Matlab neural network data evaluation functions. Performance and regression values are also considered with this inclusion.

**Confusion Matrix:**
This is a 2-dimensional matrix with a row and column for each class for training, validation, testing and all datasets. Each matrix element shows the number of test examples for which the actual class is the row and the predicted class is the column. Good results correspond to large numbers down the main diagonal. The diagonal (green cells) in each table show the number of cases that were correctly classified. The off-diagonal (red cells) show the misclassified cases. Blue cells in the bottom right show the total percent of correctly classified cases (in green text) and the total percent of misclassified cases (in red text). Figure 2 shows a confusion matrix with 4 tables each displaying the network response for the training, validation, testing and all datasets.

---

**Figure 1:** A neural network system for signal intensity prediction. The DNA sequence data are first converted into n-gram profiles as input vectors. The neural network then predicts the normalized signal intensities after network training.

**Figure 2:** A typical confusion matrix showing various types of errors that occurred for the final trained network.
Figure 3: A typical ROC plot

**Receiver Operating Characteristic, ROC:**

ROC curve shown in Figure 3 is another form of visualization and analysis of the quality of our network. It is a plot of true positive rate (sensitivity) versus the false positive rate (1-specificity). The colored lines in each axis represent the ROC curves for each category of the problem. The ROC always goes through the origin and through (1, 1). A good test would show points in the upper-left corner.

**Methodology:**

We adopt the same methods as in [15]. The dataset is from the Cambridge Reference Sequence with ascension number NC-012920 and is made of 15,453 rows and 6 columns where 3 of the columns are the n-grams for n= 1, 2, 3 and the other 4 columns represent the normalized intensities for Adenine, Cytosine, Guanine and Thymine. We extract every 26th line of the dataset which reduces the dataset to 594 rows (lines) respectively.

3-gram are used independently to predict the normalized intensities for the four nucleotides ACGT and results obtained are compared with those obtained in [15]. We also use 1-3-gram, 2-3-gram and 1-2-3-gram to repeat the analysis and compare with earlier results.

The algorithmic steps for our data manipulation are as follows:

1. Compute n-gram profiles of the DNA data set using Python programming language.
2. Calculate the nucleotide, di nucleotide and tri nucleotide frequencies of these profiles.
3. Do substitution of the nucleotides, di nucleotides and tri nucleotide strings with their respective frequencies. Do the following on the intensity profiles:
4. Calculate the highest and lowest value along each row.
5. Do normalization along each row using $N(i) = (y_i - \text{min}) / (\text{max} - \text{min})$, where $y_i$ is the actual value of the attribute $i$, max and min are the maximum and minimum values along each row.
6. Repeat step 5 for every row of intensity profile.
7. Combine results obtained from step 1 to step 6.
8. Extract every 26th line (to avoid subsequence overlap and possibility of random match) from the data set after the operations above.
9. Use Matlab subroutines to get performance plots, regression values, confusion matrices and Receiver operating characteristic curves.

The flowchart for the steps is shown in Figure 4.

![Flowchart](image)

**Figure 4:** Algorithmic flowchart for computing n-gram profiles and doing normalization on the DNA sequence

**Table 1:** The four nucleotides and their percentages (ratios)

| Nucleotides | Percentage |
|-------------|------------|
| A           | 0.31       |
| C           | 0.31       |
| G           | 0.13       |
| T           | 0.25       |
Table 2: The sixteen di nucleotides and their percentages (ratios)

| Tri nucleotides | Percentage | Tri nucleotides | Percentage |
|------------------|------------|-----------------|------------|
| GTA             | 0.02       | GAC             | 0.03       |
| TGT             | 0.02       | TCA             | 0.03       |
| TCG             | 0.01       | TAT             | 0.01       |
| TAG              | 0.01       | TAA             | 0.02      |
| TAC              | 0.01       | TAA             | 0.02      |
| TAT             | 0.01       | TAC             | 0.03      |
| TTC              | 0.01       | TGC             | 0.02      |
| TGC             | 0.02       | TCC             | 0.01      |
| TCG             | 0.02       | TGG             | 0.02      |
| TGG             | 0.03       | TGT             | 0.05      |
| TTA             | 0.02       | TAC             | 0.02      |
| TCA             | 0.02       | TCC             | 0.02      |
| TCG             | 0.02       | TGC             | 0.02      |
| TGG             | 0.03       | TGT             | 0.05      |
| TTA             | 0.02       | TAC             | 0.03      |
| TCA             | 0.02       | TCC             | 0.02      |
| TCG             | 0.02       | TGC             | 0.02      |
| TGG             | 0.04       | TGT             | 0.06      |
| TTA             | 0.03       | TAC             | 0.04      |
| TCA             | 0.03       | TCC             | 0.03      |
| TCG             | 0.02       | TGC             | 0.03      |
| TGG             | 0.02       | TGT             | 0.07      |
| TTA             | 0.01       | TAC             | 0.02      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
| TTA             | 0.01       | TAC             | 0.01      |
| TCA             | 0.01       | TCC             | 0.01      |
| TCG             | 0.01       | TGC             | 0.01      |
| TGG             | 0.01       | TGT             | 0.01      |
performance values extracted from 1 2-gram and 1-2-3-gram with variable number of neurons in the hidden layer. Tables 1, 2 and 3 show the percentages (ratios) from Affymetrix [1] dataset of nucleotides, di nucleotides and tri nucleotides respectively. Using pattern recognition toolkit to investigate the behavior of our predictions in terms of confusion matrices (CM) and receiver operating characteristic (ROC) curves, the results with 1-3-gram are shown in Table 10. The results with 2-gram and 1-2-3-gram (not shown) are not as good as those obtained using 1-3-gram.

Discussion:
As noted in [15], the absolute set comprises 4×594 values, where the four values are the absolute signal strengths of the bases [ACGT] on each of 594 lines. Absolute signal strengths are normalized to values between 0.0 - 1.0, from which the Neural Network/n-gram process predicts values (≥ 0.0 - 1.0). The Prediction set correctly identifies the highest value (1.0) in the normalized set for all 594 lines, which is, of course, the highest value and therefore the correct base call in the absolute set. This is not necessarily a trivial result, as the predictive function must accommodate 1.0, from which the values in Tables 4, 5 and 6 were generally better than the results obtained in [15] where 1-2-gram composition of the n-grams were used. This is in part due to the increment in the n-grams from 2 (two) to 3 (three). A look at Table 7, Table 8 and Table 9 shows reduction in the validation error and increment in the regression value when we compare the respective n-gram compositions. The average best validation performance (Bvp) and regression values obtained in [15] for 1-2-gram was 0.002793 which translated to 99.72% accuracy with average regression value of 0.98803. These numbers decreased (increased) to 0.002499 and 0.99112 when we used the 1-3-gram composition. Again, a comparison of 1-2-gram and 2-3-gram showed a decrease in best validation performance to 0.002253 and increase in the regression value to 0.99180 for the 2-3-gram. In the case of 1-2-3 gram, the best validation performance value again decreased to 0.002056 or 26.4% when compared with the value obtained with 1-2-gram. The regression value also increased to 0.99191 from 0.98803 obtained with 1-2-3 gram. This is again due to the increment in the n-gram number. The use of pattern recognition toolkit to investigate the behaviour of the confusion matrices and receiver operating characteristic (ROC) curves showed general confusion matrices value of 99.8% using 40 neurons in the hidden as shown in Table 10 and the points in ROC curve lying in the upper left corner. These are good signs of near expected results.

Conclusion:
We can predict the signal intensities via their normalized values from Affymetrix data using artificial neural network based n-gram model. It seems the higher the n-gram value and appropriate composition, the better the predictive accuracy of the model. The usage of higher n-gram values and their different compositions are considered in this paper. Efforts could be made to increase the number of n-grams to see if better results can be obtained which we envisage to be true. An effort could also be made to get optimal number of neurons in the hidden layer that give maximal regression values and lower mean square error. An increase in regression value to say 0.999 is indicative of a much better prediction with its attendant low mean square errors which is a measure of performance. As we increase the n-grams, we can also check which composition of the n-grams give better results. Greater confusion matrix values along the diagonal and ROC curves points in the upper most left corner can also be achieved for better classification.

References:
[1] http://www.affymetrix.com/
[2] http://www.mathworks.com/
[3] Abo-Zahhad M et al. International Journal of Information Technology and Computer Science 2012 8:22 [DOI: 10.5815/IJITCS.2012.08.03]
[4] Chang HT, Idea Group Publishing 2006 4: 63 [DOI: 10.4018/978-1-59140-863-5]
[5] Akhtar M et al. In Proc. of IEEE Workshop on Genomic Signal Processing and Statistics (GENSIPS), 2007 1 [DOI:10:1109/ GENSIIPS.2007.4365821]
[6] Cristea PD, J. Cell Mol. Med. 2002 6: 279 [PMID: 12169214]
[7] Zhang WF & Yan H, Pattern Recognition 2011 45: 947 [https://doi.org/10.1016/j.patcog.2011.08.016]
[8] Nair AS & Sreenadhan SP, Bioinformation 2006 1: 197 [PMCID: 1891688]
[9] Shannon CE, A Bell System Technical Journal 1948 27: 379
[10] Witten IH et al. Data Mining: Practical Machine Learning Tools and Techniques, Third Edition, Elsevier, MA, USA 2011 [ISBN:978-0-12-374858-0]
[11] Mitra S et al. Introduction to Machine Learning and Bioinformatics, CRC Press, London, 2008 [ISBN:158488682X]
[12] Mitchell TM, Machine Learning, WCB McGraw-Hill, Boston, 1997 [ISBN:0070428077]
[13] Wu C et al. ISMB 93 Proceedings 1993 [PMID:7584367]
[14] Hatzigeorgiou A & Megraw M, An Application of Artificial Neural Networks in Medicine, Nonconvex Optimization and Its Applications 85: 2006 [ISBN:978-0-387-30408-3]
[15] Chilaka C et al. Bioinformation 13: 313 2017 [ISSN 0973-2063]
[16] Urakawa H et al. Applied and Environmental Biology, 68: 235 2002 [PMCID:PMC126557]
[17] Pandey J & Tripathi K, Global Journal of Computer Science and Technology, 14: 1 2014 [DOI:10:11648/j.cbb.20140202.11]
[18] Abo-Zahhad M et al. Int. Journal of Inf. Tech and Computer Science, 51 2014 [DOI:10.5815/IJITCS.2014.08.07]
[19] Hapudeviya MM, Sri Lanka Journal of Bio-Medical Informatics I:104 2010 [DOI:10:4038/SLJBMI.VLI2.1719]
[20] Aloy P et al. CABIOS 13: 3 1997 [PMID:9183525]
[21] Cristea P et al. 8th WSEAS Int. Conf. on Neural Networks, Sofia Bulgaria, 2008 [ISBN: 978-960-6766-56-5]
[22] Cristea P et al. 9th Symposium on Neural Network Applications in Electrical Engineering, NEUREL, 2008 [DOI:10.1109/NEUREL.2008.4685575]

[23] Abbaci A et al. Second International Conference on Quantum, Nano and Micro Technologies, p28 2008 IEEE [DOI 10.1109/ICQNM.2008.23]

License statement: This is an Open Access article which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited. This is distributed under the terms of the Creative Commons Attribution License

Journal