EXCLI Journal 2009;8:195-202 ISSN 1611-2156
Received: June 28, 2009, accepted: August 05, 2009, published: August 05, 2009

Original article:

IDENTIFYING DNA SPLICE SITES USING PATTERNS STATISTICAL PROPERTIES AND FUZZY NEURAL NETWORKS

Essam Al-Daoud

Computer Science Department, Faculty of Science and Information Technology, Zarka Private University, Zarka, Jordan, Telephone: 962-796680005, e-mail: essamdz@zpu.edu.jo

ABSTRACT

This study introduces a new approach to recognize the boundaries between the parts of the DNA sequence retained after splicing and the parts of the DNA that are spliced out. The basic idea is to derive a new dataset from the original data to enhance the accuracy of the well-known classification algorithms. The most accurate results are obtained by using a derived dataset that consists from the highest correlated features and the interesting statistical properties of the DNA sequences. On the other hand, using adaptive network based fuzzy inference system (ANFIS) with the derived dataset outperforms well-known classification algorithms. The classification rate that is achieved by using the new approach is 95.23 %, while the classification rates 92.12 %, 86.75 %, 83.13 % and 84.51 % are obtained by Levenberg-Marquardt**,** generalized regression neural networks, radial basis functions and learning vector quantization, respectively. Moreover, this approach can be used to represent the DNA splice sites problem in form if-then rules and hence provides an understanding about the properties of this problem.

Keywords: DNA splice sites, fuzzy inference system, preprocessing, if-then rules

INTRODUCTION

The identification of the location and distribution of the gene sequences is an important source of knowledge. Several areas can benefit from studies of these structures, such as medicine, pharmacology and agriculture. They can provide, for example, insights for the development of new drugs and treatments for diseases like cancer (Lorena & de Carvalho, 2004). The estimated number of the human genes is 30,000, most of it encode proteins, but some are transcribed into non-coding RNA molecules that function in protein biosynthesis and gene regulation (Makal et al., 2008). In eukaryotic organisms, there are three major stages of producing a functional molecule of RNA or protein called gene expression. The first stage is transcription that describes the process of synthesizing a copy of the coding strand of the double-stranded DNA starting at the promoter site, thereby substituting Thymine (T) by the base Uracil (U) (Hammer et al., 2005). The second stage is the splicing, which is a result of the fact that eukaryote genes are composed of alternate segments of exons and introns. Exons are regions that code for the final protein. Introns intermediate exons and do not code for proteins. Thus, introns have to be removed from the mRNA molecule, which is accomplished in the splicing stage (Lorena & de Carvalho, 2004; Nantasenamat et al., 2005). The third stage is the translation, where the mRNA molecule is translated to the final protein (see Figure 1, from: Segovia-Juarez et al., 2007).

EXCLI Journal 2009;8:195-202 ISSN 1611-2156
Received: June 28, 2009, accepted: August 05, 2009, published: August 05, 2009

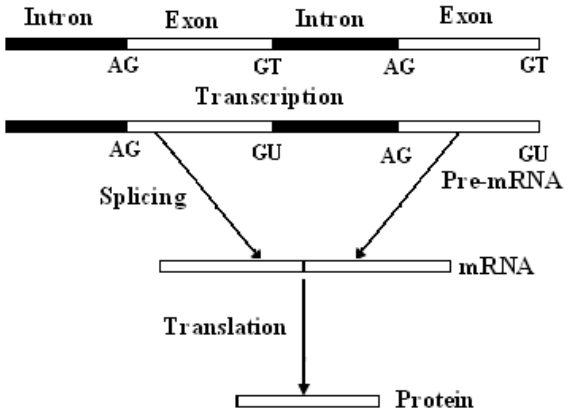

Figure 1: Major stages of producing a functional molecule of RNA or protein

The splice junction problem is to discover exon–intron boundaries (EI or donor sites) and intron–exon boundaries (IE or acceptor sites) from large DNA sequences. Various machine learning tools have been proposed to detect splice sites automatically based on DNA sequences. Pertea et al. (2001) used a Markov model with a decision tree. Sonnenburg et al. (2002) introduced specially designed support vector kernels. One of the kernels is to detect translation initiation sites in vertebrates and another used an extension to the well-known Fisher-kernel. Fogel and co-workers (2003) applied for the first time an evolutionary neural network to address the splice junction problem. Hammer et al. (2005) presented a prototype based pattern recognition tool trained for automatic donor and acceptor recognition. The developed classification model allowed fast identification of splice sites. Their method showed competitive results and the achieved model is much sparser. Segovia-Juarez et al. (2007) introduced the hypernetwork architecture as a novel method for finding DNA splice sites. The hypernetwork architecture is a biologically inspired information processing system composed of networks of molecules forming cells. Its learning is based on molecular evolution. Makal and contributors (2008) used Multi-layer Perceptron (MLP), Radial Basis Function (RBF) and Generalized Regression Neural Networks (GRNN) to analyze and detect the splice junctions of gene sequences.

MATERIALS AND METHODS

Data collection and preprocessing

The dataset in this study is downloaded from the center of machine learning and intelligent systems at the University of California. It contains 3190 instances (patterns) distributed into three classes (762 EI, 768 IE and 1655 Neither). The length of each pattern is 60 nucleotides. However, 15 patterns are deleted because it contains ambiguous letters (D, N, S or R). Five preprocessing strategies are applied in this study: Encoding the nucleotides, extracting the statistical properties, ignoring the low correlated features, normalizing the patterns, and reducing the redundant features:

- Encoding the nucleotides: Three different ways to encoding the data are suggested and used: firstly encoding the symbolic nucleotides (A, C, G, T) by integer numbers (1, 2, 3, 4). Secondly encoding the nucleotides pairs (AA, AC, AG, AT, CA…TT) by 16 integer numbers (1, 2 , 3,…, 16), in this case the DNA sequence of length 60 nucleotides is represented by 30 integer numbers. Thirdly encoding the nucleotides triples (AAA, AAC, AAG, AAT, CAA, CAC, …., TTG, TTT) by 64 integer numbers (1, 2, 3,… , 64), in this case the DNA sequence of length 60 nucleotides is represented by 20 integer numbers. On the other hand, the target classes (EI, IE and Neither) are encoded by the numbers 1, 2 and 3.
- Extracting the statistical properties: some interesting statistical properties are extracted and added as new features to the dataset. Table 1 and Table 2 summarize some statistical properties of the pairs and triples nucleotides in EI and IE patterns. For example, the percentage of the triple CAG in IE patterns at position 28 is 78 % which is approximately three times of these nucleotides in EI. This result is calculated as following:

EXCLI Journal 2009;8:195-202 ISSN 1611-2156
Received: June 28, 2009, accepted: August 05, 2009, published: August 05, 2009

$$\frac{\#\,\text{CAG at position 28 in IE patterns}}{\#\ \text{IE patterns}} * 100$$

Table 1: Some statistical properties of the DNA pairs and triples of Intron–Exon patterns

Nucleotides	Positions	EI		IE		Neither	
		Number	%	Number	%	Number	%
CAG	28	173	22	603	78	38	02
TAG	28	11	1	137	17	11	01
TT	15,17,19,21,23,25	238	31	826	107	656	39
CT	19,25	122	16	303	39	256	15
AG	29	385	50	761	99	127	07
CC	23, 25, 27	178	23	493	64	372	22

Table 2: Some statistical properties of the DNA pairs of Exon–Intron patterns

Nucleotides	Positions	EI		IE		Neither	
		Number	%	Number	%	Number	%
GT	31,35	1066	139	139	18	184	11
GG	59, 35, 41, 47	494	64	226	29	450	27
GA	33 ,13, 17	455	59	98	12	331	20
AA	33	252	33	42	5	100	6

Thus we can use Table 1 to construct a new vector (feature) such that each element in this vector is the number of the pairs and the triples that exist in the corresponding pattern at the specified position. This vector will be used to recognize the IE patterns. In the same way, Table 2 can be used to construct another vector to recognize the EI patterns.

- Ignoring the low correlated features: The correlation between each feature and the target classes is calculated, the features that have correlations less than a threshold are ignored, and the other features are processed. For example, the correlation of the third vector is 0.005, which considers very low, while the correlation of the vector 33 is 0.4811, which is the highest.
- Normalizing the patterns: it is often useful to normalize the inputs and the targets so that they always fall within a specified range as following:

$$xn = 2*(x-minp)/(maxp-minp) - 1$$

The Matlab function *premnmx* can be used to normalize the inputs and the targets so that they fall in the range [-1,1]:

```
[pn, minp, maxp, tn, mint, maxt] =
premnmx(p, t)
```

- Reducing the redundant features: In some situations, the dimension of the input vector is large, but the components of the vectors are highly correlated (redundant). It is useful in this situation to reduce the dimension of the input vectors. An effective procedure for performing this operation is principal component analysis PCA. It eliminates those components that contribute the least to the variation in the data set:

```
[ptrans, transMat] = prepca(p, 0.03).
```

This means that prepca eliminates those principal components that contribute less than 3 % to the total variation in the data set.

EXCLI Journal 2009;8:195-202 ISSN 1611-2156
Received: June 28, 2009, accepted: August 05, 2009, published: August 05, 2009

Adaptive Network Based Fuzzy Inference System (ANFIS)

ANFIS is widely tested in various applications including control, system identification, medical diagnosis systems, time series prediction, and noise cancellation (Kurian et al., 2005), ANFIS has been written in many programming languages including Matlab fuzzy logic toolbox.

Figure 2 illustrates the architecture of ANFIS. For simplicity, we assume that ANFIS has two inputs x and y and one output z. Suppose that the rule base contains two fuzzy if-then rules of Takagi and Sugeno's type:

Rule 1: *If x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$*
Rule 2: *If x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$*

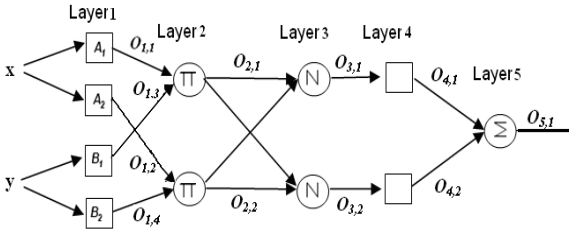

Figure 2: Adaptive Neuro-Fuzzy Inference Systems (ANFIS)

The ANFIS output is calculated by using the following steps, where $O_{j,i}$ represents the output of the i^{th} node in the layer j:

$$1\text{-}\quad O_{1,i} = \mu_{A_i}(x) \quad i=1,2$$

$$2\text{-}\quad O_{1,i} = \mu_{B_{i-2}}(x) \quad i=3,4$$

$$3\text{-}\quad O_{2,i} = O_{1,i} \times O_{1,i+2} \quad i=1,2$$

$$4\text{-}\quad O_{3,i} = \frac{O_{2,i}}{O_{2,1} + O_{2,2}} \quad i=1,2$$

$$5\text{-}\quad O_{4,i} = O_{3,i} f_i = O_{3,i}(p_i x + q_i x + r_i) \quad i=1,2$$

$$6\text{-}\quad O_{5,1} = \sum O_{4,i}$$

The membership function for *A* (or *B*) can be any parameterized membership function such as:

$$\mu_A = \frac{1}{1+\left(\frac{x-c_i}{a_i}\right)^2} \quad \text{or}$$

$$\mu_A = \exp\left(-\left(\frac{x-c_i}{a_i}\right)^2\right)$$

Training the network consists of finding suitable parameters for layer 1 and 4. Gradient decent are typically used for non linear parameters of layer 1 while batch or recursive least squares are used for linear parameters of layer 4 or even combination of both.

RESULTS AND DISCUSSION

In this section, ANFIS is applied on splice-junction gene sequences (DNA) dataset. Three datasets are derived from the main dataset: Firstly, Dataset1 contains 20 nucleotides at the middle of the sequence (most of the previous studies used 20 to 40 nucleotides at the middle). Secondly, Dataset2 contains the highest correlated features. The best results are obtained when the correlation's threshold is 0.06, at this threshold the recommended features are 28, 29, 30, 31, 32, 33, 34, 35, 36, 47, and 60. Thirdly, Dataset3 contains the highest correlated features and the two new features that are extracted from the statistical properties of the main dataset. The best results are obtained when the correlation's threshold is 0.07. Thus the recommended features in this dataset are 28, 29, 30, 31, 32, 33, 34, 35, 36, 47, 61 and 62.

A k-folding scheme with k=10 is applied. The training procedure for each dataset is repeated 10 times, each time with 90 % of the patterns as training and 10 % for testing. All the reported results are obtained by averaging the outcomes of the 10 separate tests. Table 3 shows the number of the rules generated by ANFIS, training performance, the classification rate for each class (EI, IE and Neither) and the overall classification rate for each dataset by using ANFIS. It can be noticed that the best classification rate among the datasets can be

EXCLI Journal 2009;8:195-202 ISSN 1611-2156
Received: June 28, 2009, accepted: August 05, 2009, published: August 05, 2009

obtained by using Dataset3 at spread=1.5 which is 95.23 %. Whereas the highest classification rate using Dataset1 is 88.28 % at spread=1.3, and the highest classification rate using Dataset2 is 93.31 % at spread=1.5. In general, the overall classification rate using Dataset3 is the best at all spread values as seen in Figure 3.

Table 4 compares the overall classification rate by using five neural networks methods, namely, backpropagation with Levenberg-Marquardt (LM), generalized regression neural networks (GRNN), radial basis functions (RBF), learning vector quantization (LVQ) and adaptive neuro-fuzzy inference system (ANFIS). The highest classification rate is obtained by applying ANFIS to Dataset3 (95.23 %), the next highest is by applying LM to dataset3 (92.12 %). Where the learning rate lr=0.1 and 2 hidden layers are used, the first hidden layer consists from 10 neurons and the second hidden layer consists from 9 neurons.

Table 3: The classification rate of the adaptive Neuro-Fuzzy Inference System for each dataset and numerous spreads

Data	Spread	Rules	Perf.	EI	IE	Neither	Overall
Dataset1	1	31	0.1502	0.8205	0.8381	0.9004	0.8619
	1.3	10	0.2773	0.8205	0.8937	0.9116	**0.8828**
	1.5	6	0.3025	0.7692	0.8664	0.9024	0.8606
	1.7	4	0.3315	0.8205	0.8947	0.9024	0.8805
	2	3	0.3379	0.8718	0.8684	0.9146	0.8931
Dataset2	1	18	0.1673	0.8718	0.7348	0.9136	0.8596
	1.3	9	0.1979	1.0000	0.8401	0.9268	0.9235
	1.5	6	0.2305	0.9744	0.8947	0.9350	**0.9331**
	1.7	3	0.2975	0.8462	0.9474	0.9268	0.9119
	2	2	0.3890	0.8718	0.7895	0.8293	0.8302
Dataset3	1	16	0.1790	0.9231	0.8321	0.9492	0.9122
	1.3	8	0.1890	0.9467	0.9211	0.9666	0.9460
	1.5	6	0.1950	0.9704	0.9211	0.9676	**0.9523**
	1.7	4	0.2240	0.8934	0.9434	0.9716	0.9437
	2	3	0.2808	0.9231	0.9472	0.9390	0.9370

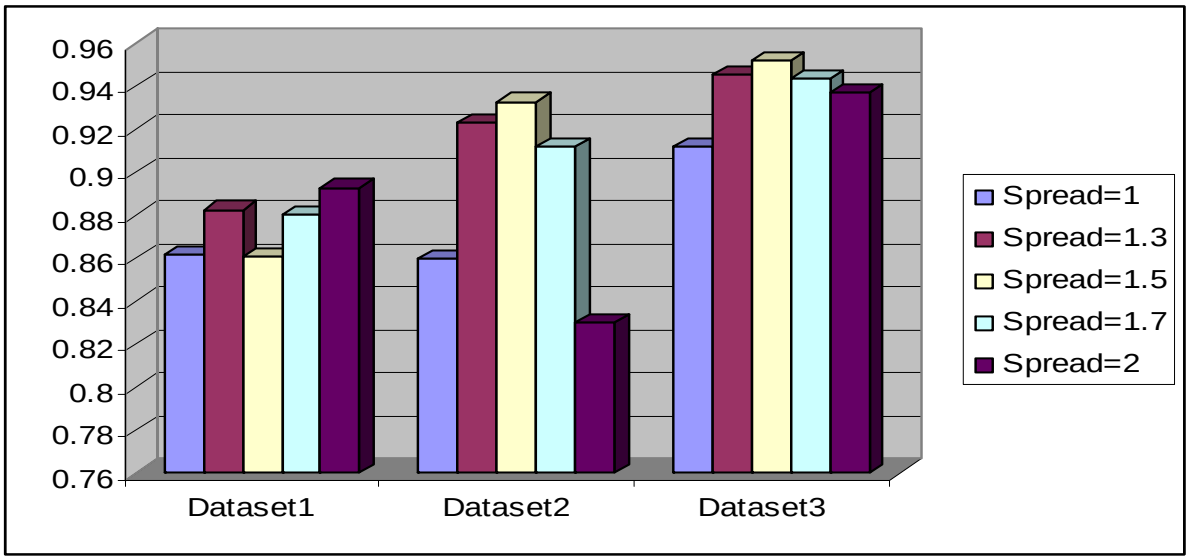

Figure 3: The overall classification rate for each dataset by using Adaptive Neuro-Fuzzy Inference System and numerous spreads

EXCLI Journal 2009;8:195-202 ISSN 1611-2156
Received: June 28, 2009, accepted: August 05, 2009, published: August 05, 2009

The next is by applying GRNN to Dataset3 (86.75 %) at spread σ = 0.1. The classification rate of applying LVQ to Dataset3 is (84.51 %), where the number of the clusters k=500 and lr=0.09. The worst testing accuracy is obtained by using RBF. Figure 4 illustrates the classification rate for each method that is applied to each dataset. It can be observed that when the Dataset3 (which contains the two new features that are extracted from the statistical properties of the main dataset) is used, then the classification rate is the highest for all neural networks methods, the next highest is obtained by using Dataset2, and the worst is by using Dataset1.

Table 5 summarizes the previous works from Makal et al. (2008). The highest classification rate in Table 5 was obtained by using Pebls (Parallel Exemplar-Based Learning Systems, a nearest-neighbor learning system designed for applications where the instances have symbolic feature values) which was only 84.24 %. The next was by using backpropagation where the classification rate was 83.51 %. Table 6 summarizes the previous works from Hammer et al. (2005). The highest classification rate in this table was 96.3 % which is obtained by using SVM_{LIK}. The next was by using SVM_{FK} where the classification rate was 94.7 %. The both results are very close to the results in this study.

Table 4: The overall classification rate by using five models and the optimal parameters

Method	Parameters	Dataset1	Dataset2	Dataset3
LM	lr=0.1 2 hidden layers Epoch 28	0.8771	0.8930	0.9212
GRNN	σ = 0.1	0.7042	0.7861	0.8675
RBF	σ = 0.3	0.5972	0.6284	0.8313
LVQ	k =500 lr =0.09 Epoch 5	0.4257	0.6412	0.8451
ANFIS	σ =1.5 Epoch 150	0.8606	0.9331	**0.9523**

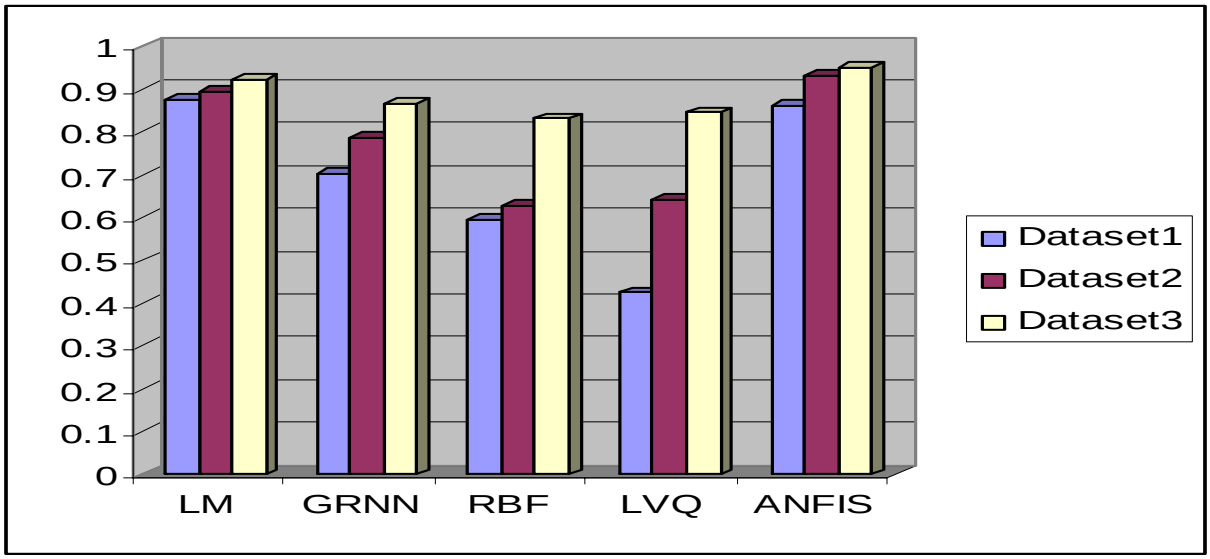

Figure 4: The classification rate for each dataset by using five neural networks models

EXCLI Journal 2009;8:195-202 ISSN 1611-2156
Received: June 28, 2009, accepted: August 05, 2009, published: August 05, 2009

Table 5: Previous works (Makal et al., 2008)

Method	KBANN	Backprob	Pebls	Perceptron	ID3	COBWEB	NN
Rate	83.37	83.51	84.27	67.27	75.43	75.5	79.26

Table 6: Previous works (Hammer et al., 2005)

Method	SVM_{LIK}	SVM_{TOP}	SVM_{FK}	HMM	DISC	BP	C4.5
Rate	96.3	94.6	94.7	94	94.1	91.2	92.4

The previous results indicate to that ANFIS with Dataset3 outperforms well-known machine learning models. Moreover, ANFIS can represent the DNA splice sites problem in form if-then rules and hence provides an understanding about the properties of this problem. It allows to incorporate a priori knowledge into the classification process and provide us with an explanation capability, which makes it possible for the user to check on the internal logic of the system. Figure 5 shows the generated membership functions and the corresponding rules.

CONCLUSION

Although the comparison in this study is restricted to few methods, but it is clear using the Dataset3 (which consists from the highest correlated features and the extracted statistical properties) with the other classification algorithms tend to increase the classification rate. On the other hand, using ANFIS with Dataset3 outperforms well-known classification algorithms. ANFIS can be used to represent the DNA splice sites problem in form if-then rules and hence provides an understanding about the properties of this problem and allows to incorporate a priori knowledge into the classification process.

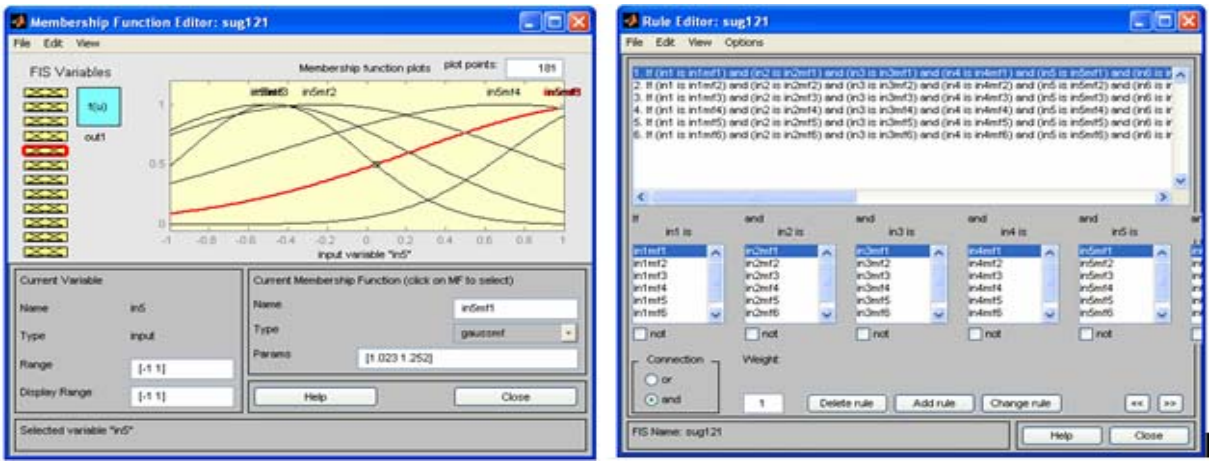

Figure 5: The membership functions of the Adaptive Neuro-Fuzzy Inference System and the corresponding rules

REFERENCES

Fogel GB, Challapilla K, Fogel DB. Identification of coding regions in DNA sequences using evolved neural networks. Evolutionary computation in bioinformatics. Amsterdam: Morgan Kaufmann Publ., 2003, pp. 195–218.

Hammer B, Strickert M, Villmann T. Prototype based recognition of splice sites. Bioinformatics Using Computational Intelligence Paradigms 2005;176:25-55.

Kurian CP, Kuriachan S, Bhat J, Aithal R S. An adaptive neuro fuzzy model for the prediction and control of light in integrated lighting schemes. Light Res Technol 2005; 37:343-52.

Lorena CA, de Carvalho AC. Evaluation of noise reduction techniques in the splice junction recognition problem. Genet Mol Biol 2004;27:665-72.

Makal S, Ozyilmaz L, Palavaroglu S. Neural network based determination of splice **j**unctions by ROC analysis. Proc World Acad Sci Eng Technol 2008;33:630-2**.**

Nantasenamat C , Naenna T, Isarankura C , Prachayasittikul V. Recognition of DNA splice junction via machine learning approaches. EXCLI J 2005;4:114-29.

Pertea M, Lin X, Salzberg SL. GeneSplicer: a new computational method for splice site prediction. Nucl Acids Res 2001;29: 1185–90.

Segovia-Juarez JL, Colombano S, Kirschner D. Identifying DNA splice sites using hypernetworks with artificial molecular evolution. Biosystems 2007;87:117-24.

Sonnenburg S, Rätsch G, Jagota A, Müller KR. New methods for splice site recognition. In: Proceedings of the International Conference on Artificial Neural Networks. ICANN 2002. Berlin, New York: Springer-Verlag, 2002, pp. 329-36.