A Multi-valued Approach For Fast Genomic Signal Processing Using Updated Perceptrons

D. Venkat Reddy[1], *Ch. D. V. Paradesi Rao*[2], *and E. G. Rajan*[3]

Abstract

This paper presents a multi-valued approach for genomic signal processing using updated perceptron for high speed genetic processing. A multi-valued approach is suggested for the improvement of system performance by means of multi valued switching function for updated perceptions during searching. The switching function proposed in our previous paper is integrated with the perceptron updation using multi-valued signal for faster convergence.(comments :this type of writing (referencing) in the abstract may be avoided.) A performance evaluation of such system for codon classification is performed and the system performances were evaluated in comparison to a single valued switching system.

Keywords : Multi-valued logic, genomic signal processing, updated perceptron, codon classification, switching function.

I-Introduction

With the increase in the development for automated and self-decisive electronic system, the need for faster computation with precise decision will increase. Where estimation precision could be achieved by various mathematical approaches, effort is required to improve the estimation at a faster rate. The need for information processing at the fastest rate is increasing, the information to be processed is also increasing. A balancing towards such an issue is needed for future systems. Various practical applications find the bottleneck of heavy data processing resulting in a slower operation. One such application is observed to be the genomic signal processing (GSP) for genome querying.

Genome sequences contain information about protein structure. This sequence is a sequence of twenty amino acids in various combinations. Each of twenty amino acids is represented in the sequence by a codon (triplet) of three nucleic acids. Since there are exactly four different nucleic acids, there exist $4^3=64$ different combinations of them. However, some of these combinations code the same amino acids. So there are exactly 20 amino acids that form a genome. There is also the 21st amino acid, which marks the end of the genomic subsequence. But it does not participate in the genome. Thus, the genetic code can be considered as the mapping between the four-letter alphabet of the nucleic acids (DNA) and the 20-letter alphabet of the amino acids (proteins) [1]. On the other hand, this mapping can be considered as a discrete function of three variables:

$G_j = f(x_1, x_2, x_3)$, where , G_j = ?1, ..., 20 is the amino acid and $x_i \in \{AGCT\}$? i? = 1, 2, 3, (A, G, C, T are the four nucleic acids Adenine, Guanine, Cytosine and Thymine, respectively). Since this function can take exactly 20 different discrete values and depends on three variables that can take exactly 4 different discrete values, it would

[1]Associate Professor, Department of E.C.E, M.G.I.T, Gandipet, Hyderabad, A.P., India dasari_reddy@yahoo.com

[2&3]Professor, Department of ECE, Vagdevi Engineering College, Warangal, A.P., India raocdvp@yahoo.co.in and rajaneg@yahoo.co.in

be natural to consider it as a partially defined multiple-valued function of a 20-valued logic.

In this paper we present the integration of the suggested multi valued logic with genomic signal processing using perceptron updation and compare the search operation with the existing single valued logic for performance evaluation. The rest of this paper is presented in VI (comments : 6) sections where a genomic signal processing operation is presented in section 2. A brief outline to multi level logic and it's derivation for generation of multi values are presented in section 3. Section 4 outlines the suggested perceptron updation approach using multi-valued logic for high speed genomic signal processing. A simulation result is presented in section 5 with the conclusion made on section 6.

II. Genomic signal processing

The standard representation of genomic information by sequences of nucleotide or amino acids symbols has certain advantages in what concerns storage, search and retrieval of genomic information, but limits the style of handling and processing genomic information to pattern matching and statistical analysis. Thus the standard representation of genomic information by sequences of nucleotide or amino acids symbols has certain advantages in what concerns storage, search and retrieval of genomic information, but limits the style of handling and processing genomic information to pattern matching and statistical analysis. This procedural limitation determines unnecessary computing costs in the case of studies involving feature extraction at the scale of whole chromosomes, multi resolution analysis, comparative genomic analysis, or quantitative variability analysis.

Genomic information represents the form of sequences of which each element can be one out of a finite number of entities. Mathematically, sequences like DNA and proteins, have been represented by character strings, in which each character is a letter of an alphabet. In the case of DNA, the size of the alphabet 4 and consists of the letters A, T, C and G; in the case of proteins, the size of the corresponding alphabet is 20. The main reason that the meadow of signal processing does not yet have significant impact in the field is because it deals with numerical sequences rather than character strings. But, if we properly map a character string into one or more numerical sequences, then digital signal processing (DSP) provides a set of novel and useful tools for solving highly relevant problems. For instance, in the form of local texture, color spectrograms visually provide significant information about biomolecular sequences, which facilitates understanding of local nature, structure, and function. In addition, both the magnitude and the phase of properly defined Fourier transforms can be used to predict important features like the location and certain properties of protein coding regions in DNA [13].

The transformation of the genome sequence into signal representation leads to the advantage of faster signal processing and computing. The issue of mining and it's feasibly for faster searching could be effectively been observed for searching algorithms using advanced learning algorithms. As though the learning algorithms were developed for the speed up and accuracy improvement of genomic signal analysis the issue of faster computation wrt. Multi level operation remains a issue. A multi level logic is an evolving approach for faster computation and is applied on various approaches for faster analysis and processing.

III. Multivalued logic

As mentioned multi level logics are used for the computation of faster operation in digital signal processing. The operation of multi level logic could be attained by representing the operational reference signal into multiple sub levels and processing under one reference signal shown in fig 1,2. Traditional logic is only two valued for any proposition. **Multi-valued logics** are logics in which there are more than two truth-values. Multi valued system has several important advantages over existing binary system. Expanding the existing logic levels to ternary and penta levels, higher processing rates could be achieved in various applications like memory management, communication throughput and domain specific computation. The first and foremost advantage of a ternary representation over binary is economy of digits. To represent a number in binary system, one needs 58% more digits than that of ternary . For example, to represent a 15-digit decimal number, one requires 34 ternary digits instead of 54 binary digits. Ternary representation accepts sign convention also. The most significant advantage is that there is reduction in the interconnection required to implement a logic function. This in turn causes reduction in the chip area while designing devices .

In contrast to the design it is also important to assess the cost of manufacturing of these types of multiple valued logic circuits. Let R be the radix, d be the number of digits to express a range of N numbers is given by $N = R^d$. If the number and/or cost of the basic hardware components C is proportional to the "digit capacity" R X d, then we have

$$C=k(Rd) = k(R\frac{\log N}{\log R})$$

where k is some constant. Differentiating this cost equation with respect to the radix R and equating to zero gives that R should equal e(=2.718) for minimum cost. From this analysis it is evident that R=3 should be more economical than the binary radix R=2. If we consider that devices or circuits are available which provide two, three four or more stable digital signals without any increase in individual costs for the higher-valued radices, then in such ideal circumstances total cost C would be proportional to d. Hence,

$$C=kd = k\frac{\log N}{\log R}$$

Which shows gradual decrease of cost C with increasing R [12].

Table 1 shows representation of decimal numbers using ternary symbols. The decimal number D in terms of ternary symbol is given by

$$D = \{T_n3^n+T_{n-1}3^{n-1}+\ldots+T_13^1+T_03^0\}$$

where T = ternary digit -1, 0, +1

n= significance of the ternary digit,

T_0=least significant

T_n = most significant.

However, the -1, 0, +1 numbering system has a unique advantage that any number can be changed from a positive value to the corresponding negative value by merely changing all -1's to +1's and vice versa, leaving all zeros unchanged [12]

Table 1. Representation of decimal numbers using ternary symbols

Decimal Number D	Ternary notations using the number -1,0,+1	Decimal Number D	Ternary notations using the number −1,0,+1
0	0000	4	0011
1	0001	5	01-1-1
2	001 −1	6	01-10
3	0010		

IV-Updated perceptron - GSP

During querying process the signals are to be matched with various defined feature before making the final decision. This decision could be improved if the estimation is evaluated and updated using defined updation. To improve the recognition factor and to speed the estimation process the system is update the operation in a recursive manner. This updataion may take longer time to provide the final decision. To improve the estimation efficiency by reducing the time of convergence for decision-making. To achieve the objective a multi-valued approach for updation and decision is presented.

Multiple-valued threshold logic over the field of complex numbers is based on the following idea proposed in [8]. Let K = {0,1, ... k-1} be a set of values of k-valued logic. Let us build one-to-one correspondence between set k and a set of kth roots of unity

$E=\{\varepsilon^0, \varepsilon^1, \varepsilon^2, \dots\dots, \varepsilon^{k-1}\}$, where $\varepsilon = \exp(i\,2\pi/k)$ is a primitive kth root of unity. Thus, the k-valued logic becomes the one over the field of complex numbers and any multiple-valued function of k-valued logic becomes a multiple-valued function over the field of complex numbers. The function values and its arguments in this case are the kth roots of unity:

$\varepsilon^j = \exp(i\,2\pi/k), j=0, \dots\dots, k-1$, where i is an imaginary unity. Let us consider the following function (k-valued predicate) proposed in [8]:

$P(z) = \exp(i\,2\pi\, j/k)$, if $2\pi/k \leq \arg z < 2\pi(j+1)/k$, where arg z is the argument of the complex number z. Function divides a complex plane onto k equal sectors and maps the whole complex plane into a subset of points belonging to the unit circle. This is a set of kth

roots of unity. A k-valued function $f(x_1, \dots\dots, x_n)$ is called a k-valued threshold function over the field of complex numbers [8]- [10] if the following condition holds for all $x_1 \dots\dots x_n$ the domain of this function:

$$f(x_1, \dots\dots, x_n) = P(w_0 + w_1 x_1 + \dots\dots + w_n x_n)$$

where

$W = (w_0 + w_1 + \dots\dots + w_n)$ is a weighting vector, P is a function. Multi valued network has been introduced in [7] as a learning element with activation function which implements those mappings described by multiple-valued threshold functions over the field of complex numbers. Multi valued network, its basic properties, and learning are comprehensively observed in [10]. A single discrete-valued Multi valued network performs a mapping between n inputs and a single output. This type of mapping is explained by a multiple-valued (k-valued) function of n variables $f(x_1, \dots.., x_n)$ with n+1 complex-valued weights as parameters. Evidently, the Multi valued network inputs and output are kth roots of unity. The learning of Multi valued network is decreased to the movement along the unit circle. This movement is always free of derivative because any path along the circle always leads to the target. The best method of this movement is determined by an error that is a difference between the desired and actual outputs. Let ε^q be a desired output of the perceptron and $\varepsilon^s = P(z)$ be an actual output of the perceptron. The most efficient MVN learning algorithm is based on the error correction learning rule [10]:

$$W_{r+1} = W_r + \frac{Cr}{(n+1)} (\varepsilon^q - \varepsilon^s)\, \overline{X},$$ where X is an input vector, n is a number of perceptron's inputs, X is a vector with the components complex conjugated to the components of vector X, r is the number of iterations, W_r is a current weighting vector, W_{r+1} is a weighting vector after correction, C_r is a learning rate. The convergence of the learning process based rule is proven in [10]. The regulation confirms such a correction of the weights that the weighted sum is moving from the sector s to the sector q. The correction of the weights changes

the weighted sum exactly on the value. The activation function is discrete. It has been recently proposed in [5],[6], to modify the function in order to generalize it for the continuous case in the following way. If $k \to \infty$ then the angle value of the sector tends to zero. Hence, the function is transformed in this case as follows:

$P(z) = e^{iArg\ z} = \frac{z}{|z|}$ where Arg z is a main value of the argument of the complex number z and |z| is its modulo. The function maps the complex plane into a whole unit circle, while the function maps a complex plane just into a discrete subset of the points belonging to the unit circle. Thus, the activation function determines a continuous-valued MVN. The learning rule is modified for the continuous-valued case in the following way [5],[6]:

$W_{r+1} = W_r + (\varepsilon^q - \varepsilon^{iArg\ z})$

$= W_r + (\varepsilon^q -)$

$= W_r + \delta X$

Where $\delta = (\varepsilon^q -)$ Learning according to the rule makes possible squeezing a space for the possible values of the weighted sum. Thus, this space can be reduced to the respectively narrow ring, which includes the unit circle inside, instead. This approach can be useful in order to eliminate a situation when very small changes either in the weights or the inputs lead to a significant change of z.

Multi-valued approach has three very important advantages in comparison with other perceptrons: 1) its functionality is higher than the one for other perceptrons; 2) its training is simpler; 3) it implements those mappings that are be described by multiple-valued threshold functions. A traditional multi-layer feed forward learning architecture (often referred as a "multilayer perceptron" - MLP) and the propagation-learning algorithm for it are well known. A multilayer architecture of the network with a feed forward dataflow through nodes that requires full connection between consecutive layers and an idea of a propagation learning algorithm was proposed in [3] by D. E. Rumelhart and J. L. McClelland. It is well known fact that multi layer function is based traditionally on the perceptrons with a sigmoid activation function [4]. multi layer function learning is based on the algorithm of error propagation. The error is being sequentially distributed from the "rightmost" layers to the "leftmost" ones. A crucial point of the propagation is that the error of each perceptron of the network is proportional to the derivative of the activation function [3],[4]. A multilayer feed forward learning architecture based on multi valued perceptrons has been recently proposed in [5],[6]. This architecture has at least two principal advantages in comparison with an MLF: derivative-free learning and higher functionality, i.e. an multi value approach with the smaller number of perceptrons outperforms an MLF with the larger number of perceptrons [5],[6].

As it is mentioned above for the single perceptron, the differentiability of the Multi valued network activation function is not required for its learning. Since the Multi valued network learning is reduced to the movement along the unit circle, the correction of weights is completely determined by the perceptron's error. The similar property holds not only for a single Multi valued network, but for any Multi valued network -based architecture. Multi valued network is a multilayer learning architecture with standard feed forward architecture, where the outputs of perceptrons from the preceding layer are connected with the corresponding inputs of perceptrons from the following layer. The architecture contains one input layer, m-1 hidden layers and one output layer. Let us use here the following notations. Let T_{sm} be a desired output of the sth perceptron from the mth (output) layer; be an actual output of the kth perceptron from the mth (output) layer. Then the global error of the architecture taken from the kth perceptron of the mth (output) layer is calculated as follows:

$$\delta^{*}_{sm} = T_{sm} - Y_{sm}$$

The Multi valued network learning algorithm is derived from the consideration that the global error of the architecture expressed in terms of the mean squared error (MSE) is minimized. The square error functional for the rth pattern $Xr = (x^r_1, \ldots\ldots, x^r_n)$ is as follows:

$$E_r = \sum_s (\delta^{*}_{sm})^2 (W)$$

where δ^{*}_{sm} is a global error of the sth perceptron of the mth (output) layer, E_r is a square error of the architecture for the rth pattern, and W denotes all the weighting vectors of all the perceptrons of the architecture. It is fundamental that the error depends not only on the weights of the perceptrons from the output layer but on all perceptrons of the architecture. The mean square error functional for the architecture is defined as follows:

$$E = \frac{1}{N} \sum_{r=1}^{N} E_r$$

where E is a mean square error of the whole architecture and N is a total number of patterns in the training set.

The propagation of the global errors δ^{*}_{sm} through the architecture is used (from the mth (output) layer to the m-1st one, from the m-1st one to the m-2nd one, …, from the 2nd one to the 1st one) in order to express the error of each perceptron , δ_{sj} , $j=1, \ldots\ldots, m,$ by means of the global errors δ^{*}_{sm} of the entire architecture.

Following the propagation learning algorithm for the Multi valued network proposed in [5],[6], the errors of all the perceptrons from Multi valued network are determined by the global errors of the architecture. The Multi valued network learning is based on the minimization of the error function. Let us use the following notations. Let be the weight corresponding to the ith input of the s_j^{th} perceptron (sth perceptron of the jth level), Y_{ij} be the actual output of the ith perceptron from the jth layer (j=1,…,m), and N_j be the number of the perceptrons in the jth layer. It means that the perceptrons from the j+1st layer have exactly N_j inputs. Let 1,..., n be the architecture inputs.

The global errors of the entire architecture are determined. We have to distinguish the global error of the architecture from the local errors of the particular output layer perceptrons. It is important to remember that the global error of the architecture consists not only of the output perceptrons errors, but of the local errors of the output perceptrons and hidden perceptrons. It implies that in order to obtain the local errors for all perceptrons, the global error must be shared among these perceptrons. The weights for all perceptrons of the architecture are corrected after calculation of the errors.

V. Simulation

As it was mentioned above, the genetic code can be considered as the mapping between the four-letter alphabet of the nucleic acids (DNA) and the 20-letter alphabet of the amino acids (proteins). In other words, the genetic code can be considered as a partially defined multiple-valued function of 20-valued logic, which is defined in earlier section. Now we have to build a one-to-one correspondence among the 20 amino acids and the 20th roots of unity and among the 4 nucleic acids and the 4th roots of unity, respectively. It should be mentioned that a set of the 4th roots of unity is a subset of a set of the 20th roots of unity. As it is well known [11] there are two complementary pairs among 4 nucleic acids (A-T and C-G, which means that in two spirals DNA A always bonds only with T and C always bonds only with G). So it would be general practice to locate A, T and C, G in the unit circle in such a way that the complementary nucleic acids will be quite the contrary to each other. Without loss of generality, let us locate them starting from the real 1 with a step 1/2 as follows: A, G, T, C. So A, T and, respectively, G, C are almost exactly the contrary to each other. Table 2 contains all 20 amino acids and their codons. It would be natural and reasonable to distribute the 20 amino acids along the unit circle in such a way that each value of function will be placed as close as it is possible to the corresponding values of the function arguments. Since the nucleic acid A is located at the "real 1" then it is natural to locate the amino acid K at the same point. The same approach leads us to the one-to-one correspondence among the 20 amino acids and a set of the 20th roots of unity. On the other hand, it is possible to distribute the amino acids among the 20th roots of unity randomly (but preserving the distribution of the nucleic acids).

Table 2 Amino acids, their single letter codes and their corresponding DNA codons. [15]

Amino Acid	Code	DNA Codons
Isoleucine	I	ATT, ATC, ATA
Leucine	L	CTT, CTC, CTA, CTG, TTA, TTG
Valine	V	GTT, GTC, GTA, GTG
Phenylalanine	F	TTT, TTC
Methionine	M	ATG
Cysteine	C	TGT, TGC
Alanine	A	GCT, GCC, GCA, GCG
Glycine	G	GGT, GGC, GGA, GGG
Proline	P	CCT, CCC, CCA, CCG
Threonine	T	ACT, ACC, ACA, ACG
Serine	S	TCT, TCC, TCA, TCG, AGT, AGC
Tyrosine	Y	TAT, TAC
Tryptophan	W	TGG
Glutamine	Q	CAA, CAG
Asparagine	N	AAT, AAC
Histidine	H	CAT, CAC
Glutamic acid	E	GAA, GAG
Aspartic acid	D	GAT, GAC
Lysine	K	AAA, AAG
Arginine	R	CGT, CGC, CGA, CGG, AGA, AGG

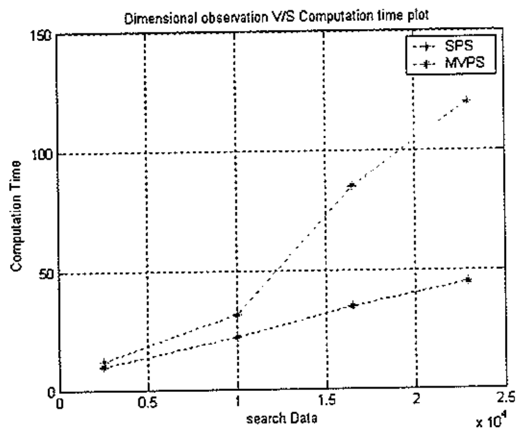

Figure 1. Dimensional observation v/s computation time slot

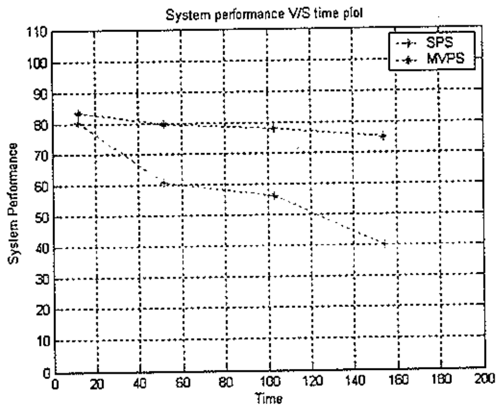

Figure 2. System performance v/s time plot

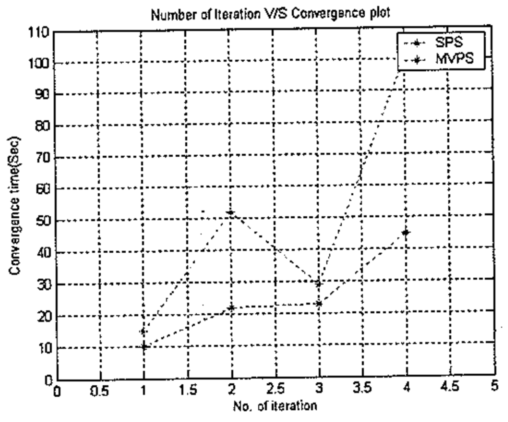

Figure 3. Number of iteration v/s convergence plot

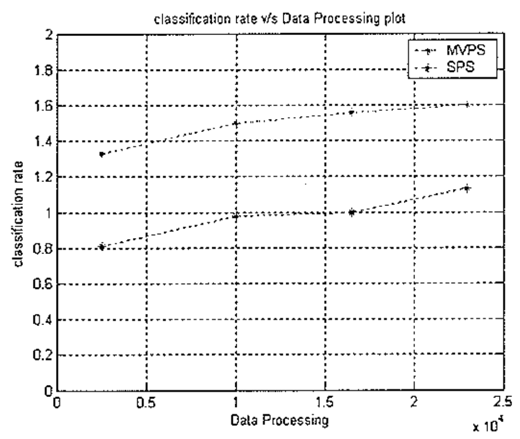

Figure 4. classification rate v/s data processing plot

VI. Conclusion

A multi-valued logic is developed for the efficient processing of codon sequence estimation for genomic signal processing. The suggested approach is developed with the objective of reducing the total computation time based on multi-valued search logic. The updated perceptron for high convergence is developed with the updation of the perceptron modification of updating node with multi level driving function for codon searching. The searching objective of a codon sequence from a large data set is a typical task in GSP as the probable sequence may reach to a very high count in the designed system. It is observed from the implementation that a Genomic processor with single value logic is less efficient than the proposed multi-valued processing system.

References

[1] B. Rost, "*Neural Networks Predict protein Structure: Hype or Hit*", In Artificial Intelligence and Heuristic Methods in Bioinformatics, (P. Frasconi and RonShamir – Eds.), IOS Press, Amsterdam, 2003, pp. 34- 50.

[2] N. Tolstrup, J. Tofgard, J. Engelbrecht and S. Burnak, "*Neural Network Model of the Genetic Code is Strongly Correlated to the GES Scale of Amino Acid Transfer Free Energies*", Journal of Molecular Biology, vol. 243, 1994, pp. 816-820.

[3] D. E. Rumelhart and J. L. McClelland, "*Parallel distributed processing: explorations in the microstructure of cognition*". MIT Press, Cambridge, 1986.

[4] S. Haykin Neural Networks: "*A Comprehensive Foundation*" (2nd Edn.), Prentice Hall, 1998.

[5] I. Aizenberg, C. Moraga, and D. Paliy, "*A Feedforward Neural Network based on Multi-Valued Perceptrons*", In Computational Intelligence, Theory and Applications. Advances in Soft Computing, XIV, (B. Reusch - Ed.), Springer, Berlin, Heidelberg, New York, 2005, pp. 599-612.

[6] I. Aizenberg and C. Moraga "*Multilayer Feedforward Neural Network Based on Multi-Valued Perceptrons (MLMVN) and a Backpropagation Learning Algorithm*" Soft Computing, Vol. 11, No 2, January, 2007, pp.169-183.

[7] N. N. Aizenberg and I. N. Aizenberg, "*CNN Based on Multi-Valued Perceptron as a Model of Associative Memory for Gray-Scale Images*", Proceedings of the Second IEEE Int. Workshop on Cellular Neural Networks and their Applications, Technical University Munich, Germany October 14-16, 1992, pp.36-41.

[8] N. N. Aizenberg, Yu. L. Ivaskiv and D. A. Pospelov, "*About one generalization of the*

threshold function" Doklady Akademii Nauk SSSR (The Reports of the Academy of Sciences of the USSR), vol. 196, No 6, 1971, pp. 1287-1290 (in Russian).

[9] N. N. Aizenberg and Yu. L. Ivaskiv, "*Multiple-Valued Threshold Logic*", Naukova Dumka Publisher House, Kiev, 1977 (in Russian).

[10] I. Aizenberg, N. Aizenberg and J. Vandewalle, "*Multi-valued and universal binary perceptrons: theory, learning, applications*", Kluwer Academic Publishers, Boston/ Dordrecht/London, 2000.

[11] M. Singer and P. Berg, "*Genes & Genomes, University Science Books*", Mill Valley, California, 1991.

[12] Stanley L. Hurst, "*Multiple-Valued Logic – Its status and its future*", IEEE Trans. on computers, Vol. C-33, No.12, December 1984.

[13] Dimitris Anastassion, *Genomic Signal Processing*, IEEE Signal Processing Magazine, 2001, pp. 8-20.

[14] P.P.Vaidyanathan, *Genomics and Proteomics* : A Signal Processor's Tour, IEEE Systems and Circuits Magazine, 2004, pp. 6-29.

[15] *Technical reports from genomic Signal Processing Lab.*, university Texas A&M, Austin, USA

Author's Biography

D. Venkat Reddy was born in India in 1966. He received his B.Tech and M.Tech degrees from Nagarjuna University, Guntur, India and J.N.T.U., Hyderabad in 1989 and 1996 respectively. He is currently working as Associate Professor in Department of Electronics & Communication Engineering, Mahatma Gandhi Institute of Technology, Hyderabad. His main research interests are multivalued logic, Digital Design. He is a member of IEEE Signal Processing Society and Computer Society, member of IETE and member of ISTE.

Prof. Dr. Chamarthy D. V. Paradesi Rao received his B.E in Electronics & Communication Engineering from Osmania University Hyderabad, 1969. He received his M.Tech in Advanced Electronics from JNTU, Hyderabad and Ph.D in Computer Science from Indian Institute of Science, Bangalore, 1983. He worked for more than 23 years as a Professor of Electronics & Communication Engineering, JNTU, Hyderabad. He published more than 12 International and 27 National papers. His main research interests include VLSI, Computers, Communications, Human Resource Development.

Prof. Dr. E. G. Rajan is the Founder President of the Pentagram Research Centre (P) Ltd., Hyderabad, India. He is an Electronics Engineer and a Professor of Signal Processing having about 30 years of experience in teaching, research and administration. He is an editor of the journal of AMSE, Barcelona, Spain. He has supervised a number of M.S and Ph.D students In India and abroad and has published a number of research papers in various international journals.