Cloud-based adaptive exon prediction for DNA analysis

Srinivasareddy Putluri[1], Md Zia Ur Rahman[1] ✉, Shaik Yasmeen Fathima[2]

[1]*Department of ECE, K L University, Green Fields, Guntur DT 522 502, Andhra Pradesh, India*
[2]*Aiboz R&D Pvt. Ltd., Navya Landmark, Lingampally, Hyderabad, T.S., 502032, India*
✉ *E-mail: mdzr22@gmail.com*

Published in Healthcare Technology Letters; Received on 14th May 2017; Revised on 28th October 2017; Accepted on 6th November 2017

Cloud computing offers significant research and economic benefits to healthcare organisations. Cloud services provide a safe place for storing and managing large amounts of such sensitive data. Under conventional flow of gene information, gene sequence laboratories send out raw and inferred information via Internet to several sequence libraries. DNA sequencing storage costs will be minimised by use of cloud service. In this study, the authors put forward a novel genomic informatics system using Amazon Cloud Services, where genomic sequence information is stored and accessed for processing. True identification of exon regions in a DNA sequence is a key task in bioinformatics, which helps in disease identification and design drugs. Three base periodicity property of exons forms the basis of all exon identification techniques. Adaptive signal processing techniques found to be promising in comparison with several other methods. Several adaptive exon predictors (AEPs) are developed using variable normalised least mean square and its maximum normalised variants to reduce computational complexity. Finally, performance evaluation of various AEPs is done based on measures such as sensitivity, specificity and precision using various standard genomic datasets taken from National Center for Biotechnology Information genomic sequence database.

1. Introduction: Cloud computing facilitates the storage and management of huge amounts of data. It is a method of delivering technology to the consumer by using Internet servers for processing and data storage, while the data are used by the client. In genomics research, cloud computing technology provides a way for researchers to enhance their capacity to store and share data, save time and costs of data sharing [1]. By DNA analysis now becoming economical, more swiftly than data computation or data storage, the genome informatics is migrating to the cloud. With next-generation sequencing that yields unparalleled amounts of data, the knack of cloud computing to search for common patterns and to generalise results will accelerate the development of treatments and diagnostic tools.

Genomics deals with the study of genomes which involves the sequencing and analysis of genomes. Cloud computing in genomics is a scalable service, where genetic sequence information is stored and processed virtually usually via networked, large-scale data centres accessible remotely through various clients and platforms over the Internet [2]. Beneath the traditional flow of information, gene sequencing laboratories transmit the data over the Internet to several sequencing archives as shown in Fig. 1.

Using this model, the gene sequence records, value-added integrators and all power users sustain their own storage and compute clusters and remain local copies of the gene sequence datasets. In this Letter, we put forward a new genome informatics cloud-based model that can be used by different healthcare organisations to store and manage large amounts of genomic sequence information of patients using Amazon Cloud Services platform [2]. The gene sequence information can be accessed from the National Center for Biotechnology Information (NCBI) gene database node using Amazon Cloud Services, which is sent as an input to adaptive exon predictors (AEPs) for locating exon regions in a DNA sequence as shown in Fig. 2. Over the past more than 25 years, there is a need for contented and efficient ecosystem for the creation and usage of gene information (Fig. 1). Sequence laboratories present their genomic data to a large collection of databases such as the NCBI, European Molecular Biology Laboratory (EMBL) database of the European Bioinformatics Institute and the DNA Data Bank of Japan.

These databases preserve, manage and provide the sequencing data. The majority of the users get the information either via websites produced by the archival databases or using the value-added integrators of genetic data such as Ensemble database, the University of California at the Santa Cruz Genome Browser. Power users and other biometricians access the genetic data from the primary and secondary sources to high-performance clusters of computers known as 'compute clusters', work along with them and abandon them when they are no longer desired (Fig. 1). Data storage, managing, accessing sequence data and cost challenges of traditional genome informatics are overcome using the cloud-based genome informatics system proposed in this Letter. With the use of this system, the stored DNA sequences can be accessed by the healthcare organisations with high speeds and genome sequence is used for prediction of exon regions which help in disease identification and drug design.

Locating the regions which code for proteins is an immense area of research in genomics. This is due to the importance of exon regions for disease analysis and design drugs. The DNA sequence is a combination of genes and inter-genic sections [3]. The study of prime protein region structure helps the secondary and the tertiary structure of protein coding regions. Once the whole structure of protein region is analysed, it is likely to detect all abnormalities, cure diseases and design drugs [4, 5]. All the living organisms are separated into eukaryotes and prokaryotes. Protein coding segments in eukaryotes are termed as exons, whereas the non-protein coding regions are known as introns. The coding regions in human eukaryotes are only 3% of the sequence and the remaining 97% are non-coding sections. Hence, the identification of coding segments in a DNA sequence is an important task [6, 7]. Almost in all DNA sequences, the coding sections display a three base periodicity (TBP) property. This is obvious by a sharp peak at a frequency $f = 1/3$ in the power spectral density (PSD) plot [8]. Many exon identification techniques presented in the literature are based on various signal processing techniques [9–13]. Adaptive algorithms are able to process very long sequences in several iterations. In this Letter, a novel bioinformatics cloud-based system is proposed to access DNA sequences and an AEP is developed using adaptive algorithms. To improve the performance of AEP than least mean square (LMS) algorithm, a variable normalised LMS (VNLMS) algorithm with its signed versions is considered. Varable least mean square (VLMS) algorithm overcomes the

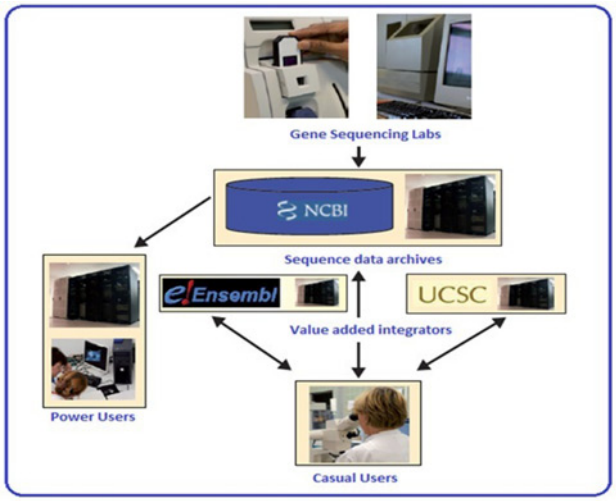

Fig. 1 *Traditional genome informatics system*

drawbacks of LMS and improves tracking ability and convergence speed. This also reduces excess mean square error (EMSE) in the process of exon prediction [14]. To overcome the computational complexity of an AEP in real-time applications, the VN adaptive algorithms are combined with sign-based algorithms. Sign-based algorithms apply signum function and reduce the number of multiplication operations [15]. Owing to normalisation, the larger tap length can be minimised to one, irrespective of tap length by using an approach called maximum variable normalisation. The fixed step size algorithms are data independent and the step size changes do not meet the tracking requirements and results in more error. Best rate of convergence requires larger step size and small EMSE requires smaller step size. To surmount this limitation, the variable step size (VSS) algorithms are used. Here, the step size is forbidden by the error obtained in the iteration process examples of VSS algorithms are seen in [16–19].

These techniques better perform than LMSs counterparts. The proposed AEP uses a hybrid algorithm based on VSS and normalisation strategy. The resultant algorithm is VNLMSs algorithm. To reduce the computational complexity, we combine VNLMSs with signed algorithms. To further minimise the computational burden on the normalisation factor, we performed maximum normalisation [19, 20]. On the basis of the VN and maximum VN algorithms, various AEPs are developed. Hybrid versions of proposed AEPs include VNLMS, VN sign regressor LMS (VNSRLMS), VN sign LMS (VNSLMS), VN sign SLMS (VNSSLMS), maximum VNLMS (MVNLMS), maximum VNSRLMS (MVNSRLMS), maximum VNSLMS (MVNSLMS) and maximum VNSSLMS (MVNSSLMS) algorithms. The performances of proposed AEPs are tested using real standard genomic datasets taken from the NCBI gene sequence database node accessed from the cloud using Amazon Cloud Services [21]. Convergence characteristics, computational complexity (O), sensitivity (Sn), specificity (Sp) and precision (Pr) are considered as performance characteristics to evaluate the performance of the various AEPs. The theory of the adaptive algorithms, results of AEPs and discussion on the performance of various AEPs is presented in the following sections.

2. AEPs using novel genome bioinformatics system based on cloud computing: A significant aspect of cloud computing in the field of genome informatics is the capability of providers of service and their customers for storing large datasets in the cloud [1]. Thanks to the Amazon Infrastructure as a Service by name 'Virtual Machines (VMs)' service, which enables storage and access of datasets as mentioned above. Amazon also provides redundancy to ensure that VMs and the datasets would not disappear due to hardware and disc failures. In the proposed genome bioinformatics model, the compute and storage resources of the community are collocated in the 'cloud' maintained by a huge service provider as shown in Fig. 2. The value-added integrators and sequence archives retain servers and storage systems inside the cloud, and utilise less or more capacity as required for seasonal and daily fluctuations in practise. Untailored users go on to get the data through the websites of the integrators and the archives, though power users at present have the choice of creating essential on demand clusters for computer in the cloud, which have direct admission to the gene sequence datasets. The proposed AEP using a novel bioinformatics genome model is shown in Fig. 2 below.

In the proposed AEP-based cloud-based bioinformatics system, the first step is getting access to the cloud services by healthcare organisations such as hospitals and researchers [2]. The advantage of cloud service is the cost cut in maintaining the data centre and the cost involved in accessing the data. The genome datasets were contributed to Amazon's repository of public datasets by a variety of

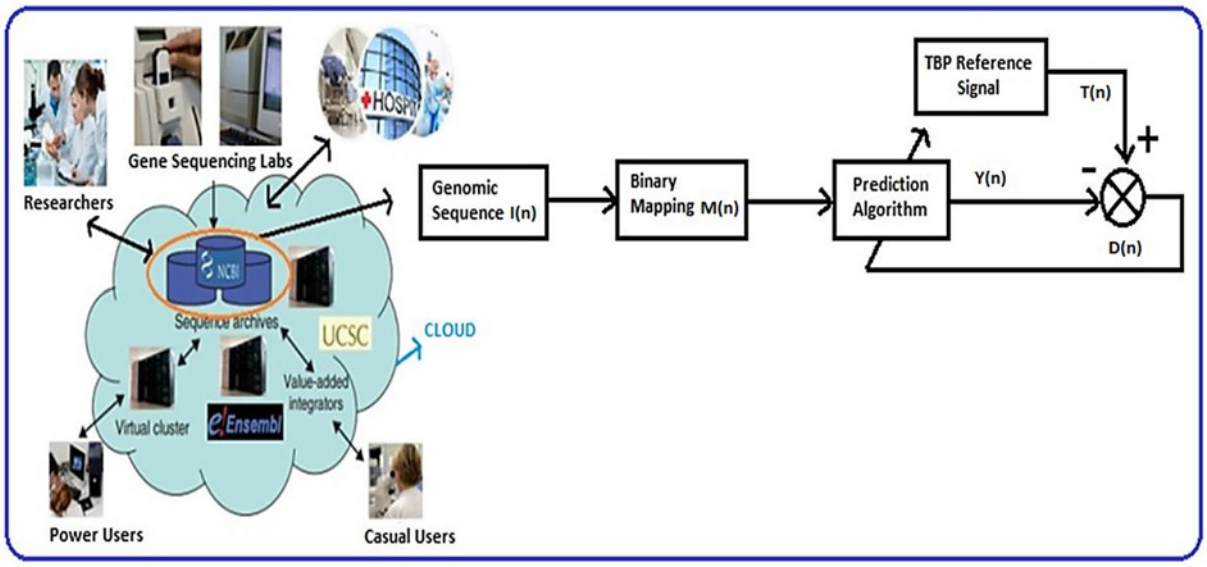

Fig. 2 *Novel genome bioinformatics cloud-based system for exon prediction*

doi: 10.1049/htl.2017.0032

institutions and can be attached to VM images for a nominal fee. They only pay for the time that the VM is running, i.e. by the minute and only while it is running, though there is a minimal storage charge of keeping the VM available. Second, converting the genomic sequences of patients are accessed from the NCBI genome database node present in the cloud into numerical notation as an input to the proposed AEP. This digital conversion is a key task of genomic sequence processing because signal processing techniques can be applied only on digital or discrete signals. Here, this binary mapping is used to convert DNA sequence accessed from NCBI node using Amazon Cloud Services into binary data which represents DNA as four binary sequences. In this, a presence of a nucleotide at a location is indicated by 1 and its absence by 0. Now, the resulting sequence is suitable to give as an input to adaptive algorithm. Consider an AEP that is developed using adaptive signal processing techniques. Let $I(n)$ be genomic sequence, $M(n)$ be mapped digital sequence, $R(n)$ be TBP obeyed genomic sequence, $T(n)$ represents the output obtained by applying the adaptive algorithm and $D(n)$ denotes the feedback signal to update weight coefficients of the adaptive method. The length of LMS adaptive algorithm is considered as 'T'. The subsequent weight coefficient in this algorithm can be predicted based on the present weight coefficient, step size parameter 'S', input sequence sample value $I(n)$ at the instance and feedback signal $D(n)$ generated in the feedback loop. Mathematical expression and analysis of LMS algorithm is presented in [14]. A typical block diagram of an AEP using a novel cloud-based genome bioinformatics system is shown in Fig. 2.

The conventional LMS adaptive algorithm may be used in exon prediction applications because of its simplicity and robustness. The LMS filter requires a priori knowledge of the input power level to select the step size parameter for stability and convergence. Since the input power level is usually one of the statistical unknowns, it is normally estimated from the data prior to beginning of the adaptation process. However, in practical situations LMS algorithm suffers with two major drawbacks. The input data vector is directly proportional to the weight update mechanism. Another one is the step size and is fixed. In real time, an algorithm has to be designed such that it has to handle both strong and weak signals. Hence, based on the filter input and output fluctuations, the tap coefficients should be adjusted accordingly. Therefore, when there is a large input data vector, the amplification intricacy of gradient noise is suffered by LMS adaptive algorithm. To avoid this problem, normalisation has to be applied. By doing so, the change applied to the coefficient of the weight vector of filter is normalised with respect to a squared form of Euclidian norm for the input vector during each iteration. Owing to normalisation, the step size varies iteratively and is proportional to the inverse of the expected total energy of immediate values of the coefficients of the data input vector. The normalised adaptive LMS algorithm generally has quicker convergence than the LMS adaptive method, because it uses an inconsistent convergence factor which aims at minimising the instant output error. Here, data normalisation is adopted in which the step size is normalised with respect to the input data vector [19]. In this case, it is a binary mapped sequence $\boldsymbol{B}(n)$. Now, the resulting algorithm is called as data normalised LMSs (DNLMS) algorithm. The expression of the weight update equation of this algorithm is written as

$$w(n+1) = w(n) + \frac{S}{c + ||I(n)||2} D(n)I(n) \tag{1}$$

The problem of small tap input vector $\boldsymbol{B}(n)$ is introduced by the DNLMS algorithm with numerical difficulties because for a squared norm has to divide by a small value. To get through this problem, its recursion is to be modified by adding a small positive constant c. The constant c is set to avoid step size parameter being too big and the denominator is too small. By means of normalising the step size of LMS by $||I(n)||2$ in the DNLMS adaptive algorithm, the problem of amplification of noise is reduced. Although the problem of amplification of noise is bypassed by the DNLMS adaptive algorithm, less complexity of computations is highly desirable for adaptive algorithms in exon prediction applications for developing nanodevices. This reduction is generally obtainable by clipping either the input data or feedback signal or both. The algorithms based on clipping of error or data are presented in [15]. These are sign regressor algorithm (SRA), sign algorithm (SA) and sign SA (SSA). This combination of three simplified SAs with DNLMSs algorithms provides fast convergence and reduced computational complexity. The convergence rate and a steady-state error of SRA, SA and SSA algorithms are slightly lower to those of the LMS adaptive algorithm for the similar parameter settings. The signum function is written as follows:

$$N\{I(n)\} = \left\{ \begin{array}{l} 1{:}I(n) > 0 \\ 0{:}I(n) = 0 \\ -1{:}I(n) < 0 \end{array} \right\} \tag{2}$$

The step size of the DNLMS algorithm can be selected independent of the number of tap weights and input signal power, which is a major advantage over LMS algorithm. For this reason, the steady-state error and convergence rate of DNLMS algorithm are much better compared with an LMS adaptive algorithm. On the other side, calculation of $S(n)$ requires some extra computations.

The DNLMS improves rate of convergence and stability over LMS algorithm. However, these methods are suitable if and only if the step size is properly chosen. The step size will make the rate of convergence either slower or faster depending on the step size and also the misadjustment. The rates of convergence and misadjustment have quite opposite step size requirements. To overcome it, the VSS algorithms were used. The DNLMS too can be considered as the VSS algorithm, because here the step size is controlled with the help of the signal power. Many proposals exist for the VSS algorithms. In our AEP implementations, we considered the VS strategy mentioned in [19] due to its simplicity and performance of the hybrid algorithms. By combining this, VSS strategy with DNLMS results in VNLMS algorithm. The weight update equations of VLMS and VNLMS algorithm are written as below:

$$w(n\ +\ 1) = w(n) + S_k I(n)D(n) \tag{3}$$

$$w(n\ +\ 1) = w(n) + \frac{S_k}{c + I(n)^2} I(n)D(n) \tag{4}$$

where the term S_k is a ratio of weighted energies updated in the iterations.

It is mathematically written as

$$S_k = \frac{\in i}{\in j} \tag{5}$$

and $\in i = \alpha \in i(n-1) + e2(n)$, $\in j = \beta \in j(n-1) + e2(n)$ and $0 < \alpha < 1$, $0 < \beta < 1$.

The term S_k makes the step size variable. The proposed novel cloud-based AEP structure presented above in Fig. 2 helped to improve the stability and the rate of convergence. Furthermore, to reduce the complexity the signum function is used. By applying this signum function to VNLMS, we can implement four simplified algorithms: VNLMS, VNSRLMS, VNSLMS and VNSSLMS algorithms.

The mathematical expressions can be written as

$$w(n+1) = w(n) + \frac{S_k}{c + I(n)^2} N[I(n)]D(n) \quad (6)$$

$$w(n+1) = w(n) + \frac{S_k}{c + I(n)^2} I(n)N[D(n)] \quad (7)$$

$$w(n+1) = w(n) + \frac{S_k}{c + I(n)^2} N[I(n)]N[D(n)] \quad (8)$$

Further the complexity can be reduced by normalising the VSS with a maximum value of data vector instead of normalising with the entire vector. This reduces the multiplications in the denominator from a length of filter to only one. Now, the modified algorithms are termed as MVNLMS algorithm and its signed versions which include MVNSRLMS, MVNSLMS and MVNSSLMS algorithms.

Mathematically, these recursions can be written as

$$w(n+1) = w(n) + \frac{S_k}{c + \max[I(n)^2]} I(n)D(n) \quad (9)$$

$$w(n+1) = w(n) + \frac{S_k}{c + \max[I(n)^2]} N[I(n)]D(n) \quad (10)$$

$$w(n+1) = w(n) + \frac{S_k}{c + \max[I(n)^2]} I(n)N[D(n)] \quad (11)$$

$$w(n+1) = w(n) + \frac{S_k}{c + \max[I(n)^2]} N[I(n)]N[D(n)] \quad (12)$$

Finally, we choose these algorithms to develop four AEPs and compare their performance with LMS-based AEP. From the performance analysis based on the measures Sn, Sp and Pr, it is evident that MVNSRLMS is just inferior to its non-sign regressor version. Hence, among the algorithms considered for the implementation, MVNSRLMS is found to be better among its signed variants with reference to computational complexity and performance measures.

3. Results and discussion

3.1. Platform and input data: Our cloud-based model uses up to three numbers of computing nodes, each of whom equipped with three cores Intel Xeon X-5550 at 2.67GHz central processing units with 64 GB of random access memory. These nodes include VM images with a complete copy of NCBI datasets (200 GBs) as node 1, datasets from the 1000 genomes project (700 GBs) as node 2 and the genome databases from Ensembl database, which includes the annotated genomes of human and 50 other species (150 GBs of annotations plus 100 GBs of sequence) as node 3. All nodes are connected with 1 GB Ethernet. In this Letter, the input genome sequences are accessed from node 1 using Amazon Cloud Services in fast A file format.

3.2. Task distribution and performance: In the proposed model, the task distribution will be done on one of three available nodes based on the location of the input genomic sequence. In the proposed model, the input gene sequences are considered and accessed from National NCBI node 1. All the three nodes can be accessed in the form of VM images. At present, one can establish an account with Amazon Web Services, launch a VM instance from the available wide variety of three generic and bioinformatics-oriented images and attach any one of available three images for genomic data processing. If the input genomic sequence is from other nodes, the task distribution will be done on that particular node.

In this model, the input genomic sequences are considered and accessed from NCBI of node 1 using Amazon Cloud Services within less time than the actual way of accessing the data. Data is accessed and execution times of gene sequences are computed to analyse the performance. The input genomic sequences then fed as input to the proposed AEPs for prediction of exon locations. The performance in terms of execution times for the five input genomic sequences with the cloud-based model is compared with traditional methods without cloud is tabulated in Table 1.

From Table 1, it was clear that the performance in terms of execution times using the proposed cloud-based model is much improved and it is more efficient than traditional methods without a cloud.

3.3. Results discussion: In this section, performance of various AEPs is compared using the proposed cloud-based model. The novel cloud-based AEP structure is shown in Fig. 2. The MVNLMS algorithm and its sign-based versions are used to develop various AEPs. For purpose of comparison, an LMS-based AEP is also developed. For evaluation purpose, we obtained standard DNA sequences from NCBI database [21]. For consistency of results, to evaluate the performance of various algorithms five DNA sequences are considered from NCBI as our datasets. The accession numbers of the sequences are E15270.1, X77471.1, AB035346.2, AJ225085.1 and AF009962 as shown in Table 2.

The performance measure is carried using parameters such as Sn, Sp and Pr. The theory and expressions for these parameters are given in [13]. The exon prediction results for sequence with accession number AF009962 are shown in Fig. 3. The performance measures Sn, Sp and Pr are measured at threshold values from 0.4 to 0.9 with an interval of 0.05. At threshold 0.8, the exon prediction seems to be better. Hence, at threshold 0.8 values are shown in Table 3. The steps in AE prediction are as follows.

The steps in AE prediction are as follows:

(i) Load the VM image of node 1 and choose input DNA sequences from node 1 genome database using the novel cloud-based genome bioinformatics system [2]. Using a binary mapping technique, convert the DNA sequence to

Table 1 Performance comparison in terms of execution times

Sequence number	Execution time with proposed model, s	Execution time without cloud, s
1	116	286
2	183	314
3	224	375
4	257	412
5	292	483

Table 2 Dataset of DNA sequences from NCBI database on node 1

Sequence number	Accession number	Sequence definition
1	E15270.1	human gene for osteoclastogenesis inhibitory factor gene
2	X77471.1	Homo sapiens human tyrosine aminotransferase gene
3	AB035346.2	Homo sapiens T cell lymphoma/ leukaemia 6 gene
4	AJ225085.1	Homo sapiens Fanconi anaemia group A gene
5	AF009962	Homo sapiens CC chemokine (CCR-5) receptor gene

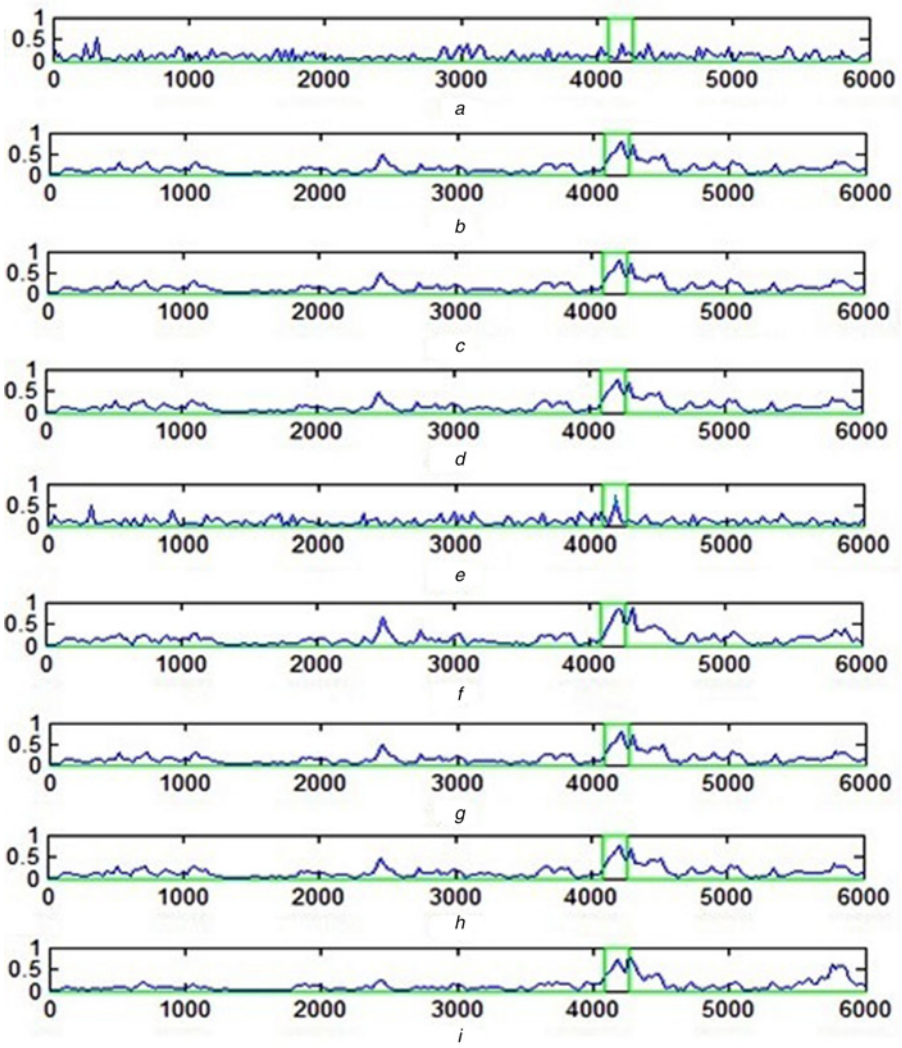

Fig. 3 *Location of exon (3934-4581) for a DNA sequence with accession AF009962 predicted using various AEPs*
a LMS-based AEP
b VNLMS-based AEP
c VNSRLMS-based AEP
d VNSLMS-based AEP
e VNSSLMS-based AEP
f MVNLMS-based AEP
g MVNSRLMS-based AEP
h MVNSLMS-based AEP
i MVNSSLMS-based AEP

Table 3 Performance measures of various AEPs with respect to Sn, Sp and Pr calculations

Sequence number	Parameter	LMS	VNLMS	VNSRLMS	VNSLMS	VNSSLMS	MVNLMS	MVNSRLMS	MVNSLMS	MVNSSLMS
1	Sn	0.6286	0.8128	0.7972	0.7795	0.7581	0.7692	0.7507	0.7416	0.7302
	Sp	0.6435	0.8021	0.7836	0.7732	0.7565	0.7684	0.7423	0.7465	0.7212
	Pr	0.5922	0.8137	0.7783	0.7697	0.7488	0.7595	0.7512	0.7396	0.7323
2	Sn	0.6384	0.8024	0.7835	0.7769	0.7597	0.7691	0.7456	0.7432	0.7318
	Sp	0.6628	0.7992	0.7841	0.7685	0.7586	0.7635	0.7523	0.7476	0.7311
	Pr	0.5894	0.8136	0.7924	0.7715	0.7526	0.7463	0.7392	0.7257	0.7186
3	Sn	0.6457	0.8028	0.7882	0.7793	0.7581	0.7692	0.7517	0.7446	0.7306
	Sp	0.6587	0.8121	0.7936	0.7592	0.7465	0.7682	0.7423	0.7365	0.7212
	Pr	0.5934	0.7994	0.7823	0.7667	0.7488	0.7596	0.7532	0.7456	0.7323
4	Sn	0.6273	0.8145	0.7936	0.7735	0.7557	0.7638	0.7537	0.7374	0.7214
	Sp	0.6405	0.8024	0.7835	0.7529	0.7497	0.7691	0.7476	0.7402	0.7318
	Pr	0.5858	0.8137	0.7941	0.7775	0.7586	0.7598	0.7443	0.7376	0.7251
5	Sn	0.6481	0.7989	0.7884	0.7615	0.7526	0.7663	0.7546	0.7457	0.7306
	Sp	0.6518	0.8058	0.7812	0.7596	0.7461	0.7692	0.7583	0.7446	0.7382
	Pr	0.5904	0.8121	0.7936	0.7782	0.7565	0.7682	0.7525	0.7465	0.7296

binary data. Give the obtained binary data as input to AEP structure.

(ii) A biological sequence obeying TBP is given as reference to the AEP.

(iii) As shown in Fig. 2, a feedback signal is generated and is used to update filter coefficients.

(iv) When the feedback signal becomes minimum, adaptive algorithm predicts the location of coding region accurately.

(v) The locations of exons are plotted using PSD. The performance measures such as Sn, Sp and Pr are measured in Table 3.

Fig. 3 shows the predicted exon locations using various adaptive algorithms. From these plots, it is clear that LMS-based AEP is not predicting the coding regions accurately. This algorithm causes some ambiguities in location prediction by identifying some non-coding regions.

In Fig. 3*a*, some unwanted peaks are identified as locations 800, 1200 and 2400th sample values. At the same time, the actual exon location 3934-4581 is not predicted. In case of VN and its maximum normalised versions, the proposed VNLMS-based algorithms exactly predicted the exon locations in 3934-4581 with good intensity of PSD. These PSDs are shown in Figs. 3*b–i*. Owing to normalisation involved in these algorithms, tracking capabilities of proposed AEPs are better than LMS algorithm. Among these three signed algorithms, MVNSRLMS is found to be better with reference to its computational complexity and convergence characteristics which need only two multiplications, which are independent of tap length of AEP. The performance measures of proposed AEPs are tabulated below in Table 3.

The convergence characteristics of MVNSRLMS are just inferior to MVNLMS, but due to a large number of reduced multiplications this inferior behaviour in convergence can be tolerable. Owing to clipped input sequence and clipped feedback signal in MVNSSLMS, the performance of exon prediction is inferior to other signed versions.

Therefore, based on computational complexity, exon prediction plots, Sn, Sp and Pr calculations shown in Table 3, it is found that MVNSRLMS-based AEP is found to be the better candidate in practical applications such as simplified architecture for the lab on a chip (LOC) or system on chip (SOC).

4. Conclusion: In this Letter, the problem of exon prediction in a DNA sequence is addressed using AEPs proposed using a novel cloud-based novel cloud-based genome informatics system at node 1. The concept of exact location prediction of exons has several applications in modern healthcare technology. Here, a new AE prediction technique is proposed. To fulfil this, cloud services based on Amazon Cloud Services based 'VMs' with custom-designed virtual hard discs for storing and accessing the genome database information and VN adaptive algorithms are considered for processing of DNA sequences. Hence, to reduce computational complexity of the proposed implementation, the concept of maximum variable normalisation is introduced instead of variable normalisation. To further minimise the computational complexity, the proposed MVNLMS algorithm is combined with sign-based algorithms. As a result, three new hybrid algorithms come into the scenario of exon prediction. These are MVNSRLMS, MVNSLMS and MVNSSLMS algorithms. Using these four algorithms, different AEPs are developed and tested on real DNA sequences obtained from NCBI database. From the performance measures shown in Table 3 and performance characteristics that are shown in Fig. 3, it is clear that MVNSRLMS algorithm-based AEP is better in exon prediction applications. This also again evidenced from the performance measures tabulated in Table 3 and PSD of exon locations shown in Fig. 3. Hence, MVNSRLMS-based AEP is suitable for practical genomic applications for the development of LOCs, SOCs and nanodevices.

5 References

[1] Kathleen C., Nocle P., Maria Kmoppers B.: 'The adoption of cloud computing in the field of genomics research: the influence of ethical and legal issues', *PLoS One*, 2016, **11**, (10), pp. 1–33

[2] Lincoln Stein D.: 'The case for cloud computing in genome informatics', *Genome Biol.*, 2010, **11:207**, (5), pp. 1–7

[3] Ning L.W., Lin H., Ding H., *ET AL.*: 'Predicting bacterial essential genes using only sequence composition information', *Genet. Mol. Res.*, 2014, **13**, (2014), pp. 4564–4572

[4] Min L., Qi L., Gamage Upeksha G., *ET AL.*: 'Prioritization of orphan disease-causing genes using topological feature and go similarity between proteins in interaction networks', *Sci. China Life Sci.*, 2014, **57**, (2014), pp. 1064–1071

[5] Inbamalar T.M., Sivakumar R.: 'Study of DNA sequence analysis using DSP techniques', *J. Autom. Control Eng.*, 2013, **1**, (2013), pp. 336–342

[6] Maji S., Garg D.: 'Progress in gene prediction: principles and challenges', *Curr. Bioinf.*, 2013, **8**, (2013), pp. 226–243

[7] Srinivasareddy P., Zia Ur Rahman M.: 'New adaptive exon predictors for identifying protein coding regions in DNA sequence', *ARPN J. Eng. Appl. Sci.*, 2016, **11**, (2016), pp. 13540–13549

[8] Saberkari H., Shamsi M., Hamed H., *ET AL.*: 'A novel fast algorithm for exon prediction in eukaryotes genes using linear predictive coding model and Goertzel algorithm based on the *Z*-curve', *Int. J. Comput. Appl.*, 2013, **67**, (2013), pp. 25–38

[9] Wazim Ismail M., Yuzhen Y., Haixu T.: 'Gene finding in metatranscriptomic sequences', *BMC Bioinf.*, 2014, **15**, (2014), pp. 01–08

[10] Ghorbani M., Hamed K.: 'Bioinformatics approaches for gene finding', *Int. J. Sci. Res. Sci. Technol.*, 2015, **1**, (2015), pp. 12–15

[11] Devendra Kumar S., Rajiv S., Narayan Sharma S.: 'An adaptive window length strategy for eukaryotic CDS prediction', *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 2013, **10**, (2013), pp. 1241–1252

[12] Azuma Y., Onami S.: 'Automatic cell identification in the unique system of invariant embryogenesis in *Caenorhabditis elegans*', *Biomed. Eng. Lett.*, 2014, **4**, (2014), pp. 328–337

[13] Guangchen L., Yihui L.: 'Identification of protein coding regions in the eukaryotic DNA sequences based on Marple algorithm and wavelet packets transform', *Abs. Appl. Anal.*, 2014, **2014**, (2014), pp. 1–14

[14] Simon Haykin O.: 'Adaptive filter theory' (Pearson Education Ltd., Harlow, UK, 2014, 5th edn.), pp. 320–380

[15] Zia Ur Rahman M., Rafi Ahamed S., Rama Koti Reddy D.V.: 'Efficient and simplified adaptive noise cancellers for ECG sensor based remote health monitoring', *IEEE Sens. J.*, 2011, **12**, (2011), pp. 566–573

[16] Nagesh M., Prasad S.V.A.V., Rahman M.Z.: 'Efficient cardiac signal enhancement techniques based on variable step size and data normalized hybrid signed adaptive algorithms', *Int. Rev. Comput. Softw.*, 2016, **11**, (10), pp. 1–13

[17] Kuang J., Yuan Ping L.: 'Variable step size LMS algorithm with a gradient based weighted average', *IEEE Signal Process. Lett.*, 2009, **16**, (12), pp. 1043–1046

[18] Kwong R.H., Edward Johnston W.: 'A variable step size LMS algorithm', *IEEE Trans. Signal Process.*, 1992, **40**, (7), pp. 1633–1642

[19] Chool Shin H., Ali Sayed H., Woo-Jin S.: 'Variable step size – NLMS and affine projection algorithms', *IEEE Signal Process. Lett.*, 2004, **11**, (2), pp. 132–135

[20] Paula Diniz S.R.: 'Adaptive filtering', in Ramirez P.S. (Ed.): 'Algorithms and practical implementation', vol. **3** (Springer Publishers, New York, NY, USA, 2014), pp. 137–207

[21] National Center for Biotechnology Information. Available at www.ncbi.nlm.nih.gov/, accessed August 2016

doi: 10.1049/htl.2017.0032