Advances and Applications in Mathematical Sciences
Volume 20, Issue 9, July 2021, Pages 1923-1936

STOCHASTIC MODELLING FOR IDENTIFYING MALIGNANT DISEASES

S. D. JENIFFER and K. SENTHAMARAI KANNAN

Department of Statistics
Manonmaniam Sundaranar University
Tirunelveli - 627012 India
E-mail: jenifferdavid1996@gmail.com
senkannan2002@gmail.com

Abstract

The tumour suppressor gene gives instructions on how to make the tumour protein that controls cell division in a specified manner. If any changes in its structure or function occur, the cell division results in either malignant cell growth or benign cell growth. Malignant cell growth can cause many forms of cancer. A model for mutated genes has been established in this paper, which will help to detect whether or not the embedded gene is mutated. If detected earlier, steps can be taken to reduce the malignancy of the disease in advance. Hidden Markov Models are a stochastic model commonly used to analyse biological sequences. Profile Hidden Markov Model can be used to compare a single sequence to a profile or to coordinate multiple sequences. Modeling the mutant genes and matching the new gene with them would become a cost-effective primary method of prevention against various chronic diseases and drug resistance.

Introduction

There may be 20000 to 25000 genes in the human body, according to Human Genome Project report. Every gene in every human body has two copies of it. Gene on chromosome 17 associated with cancer that insisted on making protein is called a tumour suppressor. The tumour protein p53 controls the division of cells and uncontrollably prevents growth and division (proliferation) this protein is attached directly to the DNA in the nucleus of every cell. If the hereditary material can damage agents such as toxic. Chemicals, the abnormal division of cell will result in UV rays of sunlight. This leads to malignant or benign cell growth if the gene is damaged. The

2010 Mathematics Subject Classification: 92D20.
Keywords: Baum-Welch Algorithm HMM, MSA, Profile HMM, Viterbi Algorithm.
Received February 28, 2021; Accepted March 25, 2021

Tp53 gene mutations could occur primarily through missense substitutions. This adds to the number, 75 percent. Nonsense mutations then contribute 9 percent and 7 percent to the insertion and deletion of frame shift. Silent mutations and several other rare modifications are the remainder of the mutations [1], [6], [15].

Cancer is the second leading cause of death in the world and is responsible for an estimated 9.6 million deaths in 2018. Globally, about 1 in 6 fatalities have been linked to cancer. In nations with low and medium incomes, about 70 percent of cancer deaths occur. Lung cancer, breast cancer, colorectal cancer and skin cancer are the most common cancers. (WHO report).

Review of Related Research

Analysis of Biological sequence by Hidden Markov Models has improved from the work of Rabiner [8]. In this paper, Hidden Markov Models for speech recognition are discussed in the theoretical practical aspect and it implemented. On the basis of this work, Eddy et al. proposed the uses of Hidden Markov models to the Computational Biology [3]. He has elucidated Hidden Markov Models and also discussed the applications of HMM for multiple sequence alignment, homolog recognition and its assumptions while using HMM for computational biology as well as biological sequence analysis. Hughey et al. Also discussed about the applications of HMM in computational biology. In [4] they have examined about the mathematical extensions and heuristics of HMM and explore it to the SH2 domain. By finding the three major principles in [2], Eddy proposed Profile Hidden Markov models to analyse the biological sequence. After few years, some of the additional features of HMM is discussed and introduced gene-HMM [11]. The usage of HMMER and SAM packages are discussed and some other open areas for biological research are listed. The study of Yoon perceiving the improvements made in HMM [14]. Many researchers have introduced and improvised the works discussed above for different diseases and have contributed their ideas.

Data Description

Mutated TP53 gene sequences of five cancer patients are used for this study. This dataset was collected from the HGMD database, which is available with some limitations.

The above-mentioned sequence is the combination of amino acids in the mutated TP53 gene. The alphabet '*A*' represents the amino acid alanine. '*C*', '*G*', '*T*' is represented Cytosine, Guanine, Thymine respectively.

Methodology

Multiple Sequence Alignment

Progressive alignment is the widely used method to align the DNA sequence. These alignment methods are heuristic algorithms where the optimization process is governed by the objective for minimization of overall pairwise scores. Most algorithms implementing progressive alignment methods use a guide tree for establishing an order in which the sequences are merged into the progressively growing multiple alignment. A guide tree is formed by taking all the sequences and applying the principles of agglomerative clustering to construct a binary tree. The leaves and internal nodes represent sequences and alignments respectively

The construction of the guide tree for a set of N sequences essentially proceeds as follows:

Step 1. The pair wise similarity (or distance) score matrix is computed.

Step 2. Each of the N sequences is considered to be a singleton group. The intergroup similarity (or distance) is identical to the pair wise similarity computed in the previous step.

Step 3. Groups are merged such that each successive merge step chooses the most similar groups and recomputes the new group's similarity (or distance) to all of the other groups.

Step 4. The merging process stops when all sequences belong to one large group containing all N sequences.

Step 5. The order in which the sequence and groups are merged provides the guide tree. Sequence alignments are performed as dictated by the guide tree.

Upon the alignment of the original seed pair wise alignment, any stage of following the guide tree will result in requiring one of the two possible alignments to be performed. Either a sequence might need to be aligned to a

group of sequences, or a group of sequences might need to be aligned to another group of sequences. When a single sequence is required to be aligned to a group of sequences, dynamic programming algorithm is applies to compute the score (or distance) of the new sequence and all the sequences in the group. The highest scoring alignment is used to determine how the new sequence is subsequently aligned to the group. And when a group of sequence is to be aligned with another group, the highest pair wise score between each member of the two groups is used to establish how the two groups align with each other. As progressive alignments are formed, the gaps introduced in a pair wise alignment are replaced with a special character, such as an *X*. This allows the gaps to progress till the end when all *X's* in the alignment constructed are replaced with the gap character -. Underlying principle in progressive alignment may therefore be stated as "once a gap, always a gap." The dynamic programming alignment algorithms must also be adjusted to accommodate the special symbol *X* such that there is no cost associated with aligning an *X* with anything including other *X* characters [12].

Hidden Markov Model

In a Markov model, all states in a linear sequence are directly observable. In some situations, some non-observed factors influence state transition calculations. To include such factors in calculations requires the use of more sophisticated models: HMMs. An HMM combines two or more Markov chains with only one chain consisting of observed states and the other chains made up of unobserved states that influence the outcome of the observed states [13].

HMM utilizes a set of hidden states with an emission of the symbols associated with each state. From a symbol generation perspective, the state sequence executed by the model is not observed. An *N*-state HMM is parameterized using the set $\lambda = \{A, B, \pi\}$. Individual elements of this set are defined as follows:

1. *A*: The $N \times N$ matrix $A = \{a_{ij}\}$ represents the state transition probabilities.

$$a_i = \Pr[q_x + 1 = s_j \mid q_x = S_i] 1 \le i, j \le N.$$

2. B: The $Q \times |\Sigma|$ emission probabilities corresponding to the emitting states. As we discuss below a subset of the N states have emissions associated with them. The elements of this matrix, $E = \{e_k(b)\}$, are defined as follows:

$$e_k(b) = \Pr[O_x = b \mid q_x = S_k] 1 \le k \le Q, 1 \le b \le |\Sigma|.$$

3. π : The initial state distribution probabilities,

$$\pi_i = \{\pi_i\}, \pi_i = \Pr[q_1 = S_i] \pi = \{\pi_i\}. 1 \le i \le N.$$

As a first step towards inducing the model, the topology of the HMM is established using the consensus sequence. The aligned columns of symbols correspond to either emission from the same match state or to emissions from the same insert state. In this formalism therefore, the columns that correspond to the match state are established to define the match states of the HMM architecture [2], [12].

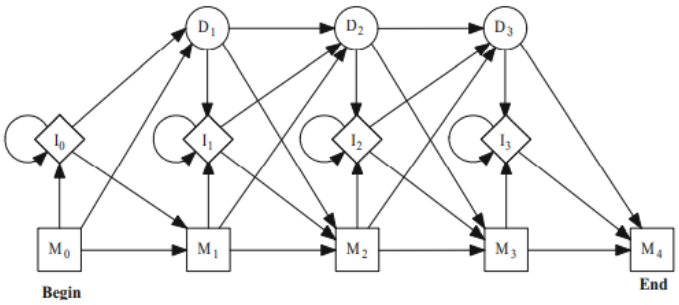

Figure 3.1. The consensus columns are used to define the match states M1, M2 and M3 for the HMM.

Transition Probabilities. The value of each transition probability is computed using the frequency of the transitions as each sequence is considered. The model parameters are computed using the state transition sequences.

Emission Probabilities. Having thus specified the state transition sequence, the emission probabilities for each of the symbol, $\alpha \in |\Sigma|$ is computed for each match and insert state, k, in the model. The emission probability is computed using the formula. Thus an emission probability is associated with each state, and specifies the probability of emitting each of

the symbols in $|\Sigma|$ in the state k [12].

$$e_k(\alpha) = \frac{Freq_k^{(\alpha)}}{\sum_{v} (Freq_k^{(v)})}.$$

Viterbi Algorithm

Once the HMM topology is set and its parameters trained, we can use it to find genes in a newly unlabelled DNA sequence X. In other words, we seek an appropriate state path H^* that best explains how the model could have produced X; this process is called HMM decoding. The simplest measure of "best" is to find the path that has the maximum probability in the HMM, given the sequence X. Recall that the model gives the joint probabilities $\Pr(H, X)$ for all sequence/annotation pairs, and as such, it also gives the posterior probability $\Pr(H/X) = \Pr(H, X)/\Pr(X)$, for every possible state path H through the model, conditioned on the sequence X. We will seek the path with maximum posterior probability. Given that the denominator $\Pr(X)$ is constant in the conditional probability formula for a given sequence X, maximizing the posterior probability is equivalent to finding the state path H^* that maximizes the joint probability $\Pr(H^*, X)$. The most probable state path can be found in time linear in the sequence length by the Viterbi algorithm. This simple dynamic programming algorithm computes the optimal paths for all prefixes of X; when we move from the i-length prefix to the $(i+1)$-length prefix, we need only add one edge to one of the recomputed optimal paths for the i-length prefix. For every position i in the sequence and every state k, the algorithm finds the most probable state path $h_1, \ldots, h_i$ to generate the first i symbols of X, provided that $h_i = k$. The value $V[i, k]$ stores the joint probability $\Pr(h_1, \ldots, h_i, x_1, \ldots, x_i)$ of this optimal state path. Again, if $h_1, \ldots, h_i$ is the most probable state path generating $x_1, \ldots, x_i$ that ends in state hi, then $h_1, \ldots, h_{i-1}$ must be the most probable state path generating $x_1, \ldots, x_{i-1}$ and ending in state h_{i-1}. To compute $V[i, k]$, we consider all possible states as candidates for the second-to-last state, h_{i-1} and select the one that leads to the most probable state path, as

expressed in the following recurrence [5]:

$$V[i, k] = \begin{cases} s_k \cdot e_k, x_1 & \text{if } i = 1 \\ \max\limits_l V[i-1, l] a_{l,k} e_k, x_i & \text{otherwise.} \end{cases}$$

Baum-Welch Algorithm

The Baum-Welch algorithm starts from an initial set of model parameters θ_0. In each iteration, it changes the parameters as follows:

Step 1. Calculate the expected number of times each transition and emission is used to generate the training set T in an HMM whose parameters are θ_k.

Step 2. Use the frequencies obtained in step 1 to re-estimate the parameters of the model, resulting in a new set of parameters θ_{k+1}.

The first step of the algorithm can be viewed as creating a new annotated training set $T(k)$, where for each unannotated sequence $X \in T$, we add every possible pair (X, H) of the sequence X and any state path, weighted by the conditional probability $\Pr(H/X, \theta_k)$ of the path H in the model with parameters θ_k, given the sequence X. The second step then estimates new parameters θ_{k+1} as in the supervised scenario, based on the new training set $T(k)$. The Baum-Welch algorithm achieves the same result in $O(nm^2)$ time per iteration using the forward and backward algorithms to avoid explicitly creating this exponentially large training set [5].

Profile Hidden Markov Model

Profile HMM use position-specific scoring. It allows HMMs to characterize entire families of sequences by modelling the extent to which the regions should be conserved in a multiple alignment. The probability of a gap or insertion is position specific. *A* 'profile' can be thought of as a series of amino acid probability distributions, one for each *M*-state. Each delete state adds a delete character to the sequence with probability 1.

In the standard notation view,

m-Match state (output), *I*-insert state (output), *d*-delete state (no output)

In addition to the start and end states, there can be 3 other classes of states: main, delete, insert. The graphical representation of a typical profile HMM is shown below.

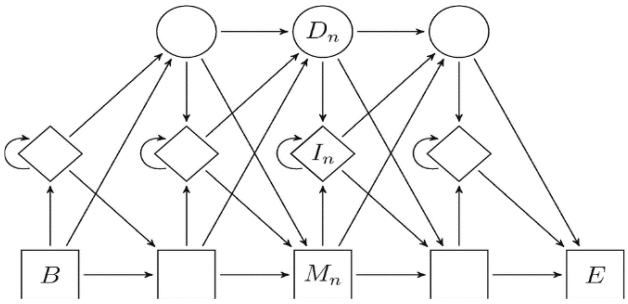

Figure 3.1. Graphical representation of profile HMM.

The model generated from multiple alignment consists of a linear sequence of nodes with a begin state (*B*) and an end state (*E*). Each node between the beginning and end states corresponds to a column in a multiple alignment. Each node has a match state (*M*), insert state (*I*) and delete state (*D*) with position-specific probabilities for transitioning into each of these states from the previous node. Each match, insert, delete state have position-specific probabilities for match, insert, delete a particular residue. These probabilities indicate the probability of transitioning.

The overall training method used is as follows:

Step 1. The model is initialized with estimates of transition probabilities and amino acid composition for each match and insert state.

Step 2. All possible paths through the model for generating each sequence are examined.

Step 3. A new version of the HMM is produced that uses the results in the previous step to generate new transition probabilities and match-insert state compositions.

Step 4. The previous two steps are repeated up to 10 more times until the parameters do not change significantly.

Step 5. The trained model is used to provide the most likely path for each sequence.

The model is then used to search a sequence database for additional sequences that share the same sequence variation. This gives a type of distance score of the sequence from the model, thus providing an indication of how well a new sequence fits the model and whether the sequence may be related to the sequences used to train the model [5].

Results and Discussion

Initially, DNA sequences are aligned using the Clustal Omega alignment tool. The multiple sequence alignment by the tool has been based on the distance matrix. Which is also organized as a guide tree. The phylogenetic tree for aligning the taken DNA sequences is shown below.

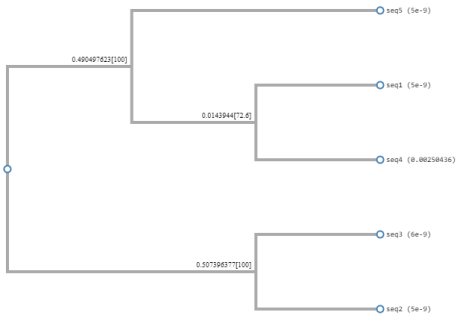

Figure 4.1. phylogenetic trees for the alignment of five DNA sequences.

There is sequence-sequence, sequence-profile, profile-profile alignment might be done respectively.

Based on the above-mentioned phylogenetic tree, the multiple sequence alignment might have been done as follows: the first mutated sequence and the fourth one was going under the sequence-sequence pair wise alignment, and the second and third sequence also aligned using the sequence-sequence alignment. Then the remaining sequence has been inclusively aligned with the first generated profile. We might have two profiles of DNA sequence alignment. With this, the profile-profile alignments have been applied and the below mentioned alignment has been generated.

```
seq4    CCGTGGCCCCTGCACCAGCAG---CTCCTACACCGGCGGCCCCTGCACCAGCCCCCTCCT
seq5    CCGTGGCCCCTGCACCAGCAG---CTCCTACACCGGCGGCCCCTGCACCAGCCCCCTCCT
seq1    CCGTGGCCCCTGCACCAGCAG---CTCCTACACCGGCGGCCCCTGCACCAGCCCCCTCCT
seq2    GCAG--TCACAGCACATGACGGAGGTTGTGAGGCGCTGCCCCCACCATGAGCGCTGCTCA
seq3    CCACTACAACTACATGTGTAACAGTTCCTGCATGGGCGGCATGAACCGGAGGCCCATCCT
                *.    ..*:.**  :* ..    *  *... *  * *.  : *.  **  *    *:

seq4    GGCCCCTGTCATCTTCTGTCCCTTCCCAGAAAACCTACCAGGGCAGCT-ACGGTTTCCGT
seq5    GGCCCCTGTCATCTTCTGTCCCTTCCCAGAAAACCTACCAGGGCAGCT-ACGGTTTCCGT
seq1    GGCCCCTGTCATCTTCTGTCCCTTCCCAGAAAACCTACCAGGGCAGCT-ACGGTTTCCGT
seq2    GATAGCG---ATGGTCTGGCCCCTCCTCAGCATCTTATCCGAGTGGAAGGAAATTTGCGT
seq3    CACCATCATCACACTGGAAGACTCCAGTGGTAATCTACTGGGACGGAACAGCTTTGAGGT
         . .      *   *  .  .*  *.  .. *:  **   *.. .*.: .   **   **
```

Figure 4.2. Multiple sequences alignment for five DNA sequences.

The aligned sequences might be visualized using the consensus logos which were introduced by Schneider and Stephen. There are several tools are available to generate to sequence logo. Skylign, online tool in the visual representation of DNA sequences is used to visualize the MSA. How each residue at each position be predominating and how much predominating can be studies from this logos. The below mentioned logo anticipating much information. Frequently occurred residues are identifying and allows to form consensus sequences.

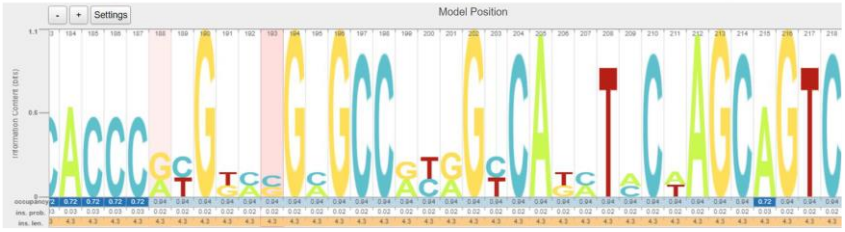

Figure 4.3. Consensus sequence Logo for Multiple Sequence Alignment.

Also, we can construct the percentage identity matrix for the DNA sequence alignment. It represents the similarity among the taken DNA sequences.

seq1	100%				
seq4	99.53%	100%			
seq5	97.91%	100%	100%		
seq2	52.91%	50.96%	55.35%	100%	
seq3	28.9%	31.65%	28.45%	74.94%	100%
	seq1	**seq4**	**seq5**	**seq2**	**seq3**

Figure 4.4. Percentage identity matrix of the DNA sequences.

From this, the similarity of DNA sequences among the five cancer patients can be studied. Also, it represents that there may be mutations might happen in any position of a particular gene but the same thing among all is the mutation leads to malignant which tends to cancer. The next step is to develop Hidden Markov Model.

The Hidden Markov Model deals with the visible states and hidden states. In the DNA sequence alignment, the visible states are nucleotides such as Adenine, Thymine, Guanine, and Cytosine. The hidden states are Match state, insert state, and Delete State respectively. We can estimate the Transition Probability Matrix and Emission Probability Matrix for the visible states and hidden states using Baum-Welch algorithm.

TPM gives the state wise transition probability between each visible state. On the other hand, EPM expresses the transition probability between each visible state to each hidden state. Generally, In Markov model the transition probability for one state to another state is strictly depends upon the previous state only.

We may predict the next aligned state using the Viterbi algorithm. The subsequent hidden states which have been computed are listed below.

D DD M D D M I M I I M M I I M MMM I M D M D M MMM D M I M MMM D I
I M MMM D DDD M M D I I M MMM D DD M M D DD M M D DDDD M MM I
M D DDDD M MMM D D M MM D I I M M D D I D DDD M M I I

Figure 4.5. Predicted Viterbi Path of DNA sequences.

Using the aligned sequence which has been done by multiple sequence alignment, we have to construct the profile. In this profile, a part of aligned sequences has been shown. There are ten Match, Insert and Delete states. The emission probability is noted in the square, which can be measured using the thickness of the line near the residual. The match, delete and insert states are joined with other by straight lines of thickness according to their probability.

The profile HMM derived from the Multiple Sequence Alignment by Laplace's rule is visualized below:

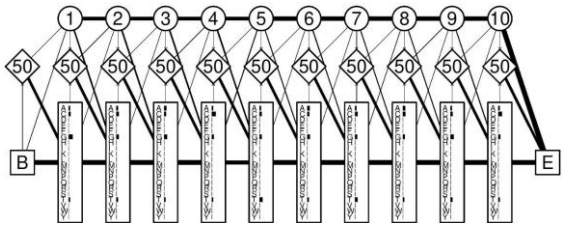

Figure 4.6. Profile HMM for the aligned DNA sequences.

The derived profile HMM from the MSA has a match state for each residue y_i. Here we show only ten aligned amino acids of the entire DNA sequence of our study. This is the alignment of a part of our DNA data; Emission probabilities are shown as bars opposite the different amino acids for each match states, and transition probabilities are indicated by the thickness of the lines. The bar near each amino acid indicates the emission probability.

Once the profile was constructed, we have to train the model. Among the five DNA sequences, we randomly choose one sequence as training set. In our model we have select the DNA sequence of the first patient to train. Training of model and iterations are as follow.

Iteration 1 log likelihood = – 241.0646

Iteration 2 log likelihood = – 221.3798

Iteration 3 log likelihood = – 207.8868

Iteration 4 log likelihood = – 202.9593

Iteration 5 log likelihood = – 201.2624

Iteration 6 log likelihood = – 200.2661

Iteration 7 log likelihood = – 199.4311

Iteration 8 log likelihood = – 198.6151

Iteration 9 log likelihood = – 197.7914

Iteration 10 log likelihood = – 197.0278

Iteration 11 log likelihood = – 196.4073

Iteration 12 log likelihood = – 195.8262

Iteration 13 log likelihood = – 195.1387

Iteration 14 log likelihood = – 194.6063

Iteration 15 log likelihood = – 194.3908

Iteration 16 log likelihood = – 194.3195

Iteration 17 log likelihood = – 194.2907

Iteration 18 log likelihood = – 194.2738

Iteration 19 log likelihood = – 194.2604

Iteration 20 log likelihood = – 194.2486

Iteration 21 log likelihood = – 194.2378

Iteration 22 log likelihood = – 194.228

Convergence threshold reached after 22 EM iterations

This shows that the 22nd iteration we can have the profile. By following, the above-mentioned procedure, we might have been identified and detect the DNA sequence of a new person whether the mutation in TP53 has leads to malignant disease or not.

The trained model is used to provide the most likely path for each sequence by using the Viterbi algorithm. The probability is computed by the forward-backward algorithm. Thus providing an indication of how well a new sequence fits the model and whether the sequence may be related to the sequences used to train the model [11].

Conclusion

In biological sequence Analysis, multiple sequence alignment is the vast using technique to analyze the DNA/Protein sequences. It will be helpful to detect and identify various diseases. If the gene mutation is identified earlier, it will become a cost-effective primary prevention method from malignant diseases and resilience to drugs. When we consider the population, which require this testing, the time and the cost reduction will work out to be huge savings for the policymaker and society, says the study. The model developed by this study using stochastic methods will make the gene-sequence teaching faster, cost effective and simple, it says. The study will be useful to analyze the Genome, identifying and detecting the malignant disease arise by gene mutation. The DNA bar-coding, DNA recognition and speech recognition will follow the same method. The launch of a profile that hides the Markov model [10] also detects Android malware recently. In speaking recognition, the model can also be useful.

Financial Support and Funding

This work was financially supported by Manonmaniam Sundaranar University, Tirunelveli (MSU/RES/Fellowship/19214012152039/2019).

References

[1] L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, S. A. Aparicio, S. Behjati, A. V. Biankin and M. R. Stratton, Signatures of mutational processes in human cancer, Nature 500(7463) (2013), 415-421.

[2] R. Durbin, S. R. Eddy, A. Krogh and G. Mitchison, Biological sequence analysis: probabilistic models of proteins and nucleic acids, Cambridge university press (1998).

[3] S. R. Eddy, Multiple alignment using hidden Markov models, ISMB 3 (1995), 114-120.

[4] R. Hughey and A. Krogh, Hidden Markov models for sequence analysis: extension and analysis of the basic method, Bioinformatics 12(2) (1996), 95-107.

[5] I. Mandoiu and A. Zelikovsky Bioinformatics algorithms: techniques and applications 3 (2008), John Wiley and Sons

[6] S. S. McDade and M. Fischer, TP53. (2019)

[7] A. Petitjean, M. I. W. Achatz, A. L. Borresen-Dale, P. Hainaut and M. Olivier, TP53 mutations in human cancers: functional selection and impact on cancer prognosis and outcomes, Oncogene 26(15) (2007), 2157-2165.

[8] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77(2) (1989), 257-286.

[9] S. C. Rastogi, N. Mendiratta and P. Rastogi, Bioinformatics Methods and Applications: Genomics, Proteomics and Drug Discovery, PHI Learning Private Limited, (2008), New Delhi.

[10] S. K. Sasidharan and C. Thomas, Pro Droid-An Android malware detection framework based on profile hidden Markov model, Pervasive and Mobile Computing (2021), 101-336.

[11] B. Schuster-Bockler and A. Bateman, An introduction to hidden Markov models, Current protocols in bioinformatics 18(1) (2007), A-3A.

[12] G. B. Singh, Fundamentals of bioinformatics and computational biology (2015),

[13] J. Xiong, Essential Bioinformatics, Cambridge: Cambridge University Press doi:10.1017/CBO9780511806087 (2006).

[14] B. J. Yoon, Hidden Markov models and their applications in biological sequence analysis, Current genomics 10(6) (2009), 402-415.

[15] T. Zenz, B. Eichhorst, R. Busch, T. Denzel, S. Habe, D. Winkler and S. Stilgenbauer, TP53 mutation and survival in chronic lymphocytic leukemia, Journal of Clinical Oncology 28(29) (2010), 4473-4479.