Advanced Modeling and Optimization, Volume 10, Number 2, 2008

Finite state machine for mutation

Dipankar Mazumdar and Swapan Raha

Department of Mathematics
Visva-Bharati (A Central University)
Santiniketan 731235, India
itsmedip@rediffmail.com, raha_swapan@rediffmail.com

Abstract: In this paper an attempt has been made to represent biological mutational activities through sequential mathematical machines. Begin with a brief theory of finite state sequential machines genetic sequences are presented in a systematic manner. A mutator has been shown as a variety of finite state machines. The mutation is studied systematically and an automaton have been constructed with the biological sequences. It has also been shown that the automaton constructed is the minimal state automaton. DNA sequences are represented as regular expressions. A case study with FMR-1 gene is presented as an illustration.

Keywords: Automaton, Mutation, Mutator, FMR-1 gene.

AMO-Advanced Modeling and Optimization ISSN: 1841-4311

Pages 241-265

1 Introduction

Our motivation for the present work is derived from biomolecular sequence comparison (Allison et al. 1992, Durbin et al. 1998). Given a biomolecular sequence we first represent it as a string over an alphabet. This problem can be addressed using finite state machines. Many researchers have used finite state automata for string matching. Allison and co-workers (Allison et al. 1992) have proposed the use of finite state models for mutation. D.B. Searls and K.P. Murphy (Searls and Murphy 1995, Searls 1995, Searls 1999) proposed an automata theoretic model to compute the relationship between simple mutational models, edit distance and string alignment in a biological context. In this paper, we develop a theory of mathematical machines — both with and without output — in order to represent and manipulate biological mutational activities. In Section 2 we give some ideas about biological mutation. In Sections 3 and 4 we have developed a theory of finite state mathematical machines without and with output. In Section 5 we present a mathematical formulation of biological mutational activities with illustrations. Section 6 of the paper is devoted for a case study with FMR-1 gene for different species. The paper is briefly concluded in Section 7.

2 Mutation

DNA are polymeric molecules made up of linear, unbranched chains of monomeric sub-units called nucleotides. Each nucleotide has three parts: a sugar, a phosphate group and a nitrogenous base. In DNA, the sugar is $2'$-deoxyribose and the bases are: adenine (A), thymine (T), cytosine (C) and guanine (G). Adenine and guanine are purine bases and thymine and cytosine are pyrimidine bases. The particular order of the bases arranged along the sugar phosphate backbone is called the DNA sequence; the sequence specifies the exact genetic instructions required to create a particular organism.

Every organism has an inherent tendency to undergo change from one hereditary state to another. Such hereditary change is called mutation. Genomes are dynamic entities that evolve over time due to the cumulative effects of small-scale sequence alterations caused by *mutation* and larger scale rearrangements arising from *recombination*. Mutation is an alteration in the

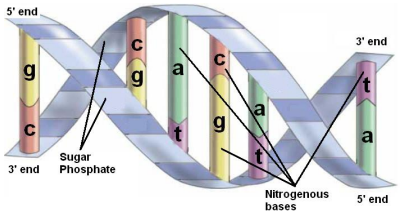

Figure 1: The structure of DNA.

nucleotide sequence of a DNA molecule.

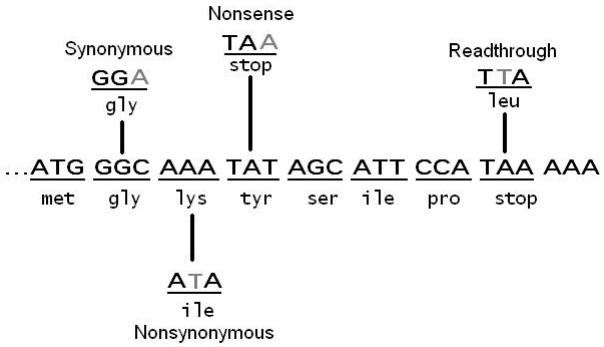

Figure 2: Different effects of mutation.

A *nonsynonymous* or *missense mutation* is a gene mutation in which a base pair change in the DNA causes a change in an mRNA codon so that a different amino acid is inserted into the polypeptide. A phenotypic change may or may not result depending on the amino acid change involved. The mistaken (missense) amino acid may be: acceptable, partially acceptable or unacceptable- with regards to the function of protein molecule. A *nonsense mutation* is a gene mutation in which a base pair change in the DNA results in the change of an mRNA codon from one for an amino acid to one for a stop (nonsense) codon (`UAG`, `UAA` and `UGA`). *Readthrough* is the change that

could convert a termination codon into one specifying an amino acid.

Mutagen is a chemical or physical agent that causes mutations. Many chemicals that occur naturally in the environment have mutagenic properties and these have been supplemented in recent years with other chemical mutatgens that result from human industrial activity. Some important types of physical mutagens are UV radiation, ionizing radiation, heat.

3 Sequential machine without output

Definition 1. Throughout this paper we shall be concerned with a fixed non-empty finite set $\Sigma_k = \{\sigma_0, \sigma_1, \sigma_2, ..., \sigma_{k-1}\}$, called the input alphabet. The elements $\sigma_0, \sigma_1, \sigma_2, \cdots \sigma_{k-1}$ will be called input letters or symbols.
An input word or tape is defined to be any finite sequence of input letters. The set of all possible words will be called the input dictionary and is denoted by Σ_k^*. The null word is denoted by ϵ that contains no letter at all. The length of a word x to be denoted by $lg(x)$ is defined to be the number of letters in x.
Definition 2. $\Delta_l = \{\delta_0, \delta_1, \cdots, \delta_{l-1}\}$ is a fixed non-empty finite set called the output alphabet. The elements $\delta_0, \delta_1, ..., \delta_{l-1}$ will be called the output letters or symbols.
Definition 3. Concatenation of any two words x and y is defined to be the word obtained by writing x followed by y and is denoted by xy. We also write $xx = x^2$, $xxx = x^3$, ... etc, $x^0 = \epsilon$.
Definition 4. A sequential machine without output is a quadruple $S = \langle S, \Sigma_k, M, a\rangle$, where S is a non-empty set called the set of internal states, Σ_k is the input alphabet, M is a function from $S \times \Sigma_k \to S$, called the transition function or the next state function, a is a given element of S, called the initial state. S is called a finite state machine or a finite machine if the set S is finite.
For a finite machine S having internal states $s_0, s_1, ..., s_{n-1}$, the function M is usually given by a table called the transition table which shows $M(s_i, \sigma_j); s_i \in S, i = 0, 1, 2, ...n-1; \sigma_j \in \Sigma_k, j = 0, 1, 2, ..., k-1$. A finite state machine may be represented by means of a directed graph called the transition graph or Moore graph in which every state is represented by a node and node s_i is joined to node s_k by means of a directed edge labeled

σ_j if and only if $s_k = M(s_i, \sigma_j)$.
Definition 5. The domain of definition of the transition function M of *Definition 4* is extended from $S \times \Sigma_k$ to $S \times \Sigma_k^*$ by the following
(i) $(\forall s)_S\ M(s, \epsilon) = s$;
(ii) $(\forall s)_S\ (\forall \sigma)_{\Sigma_k}\ (\forall x)_{\Sigma_k^*}\ M(s, x\sigma) = M(M(s, x), \sigma)$.
Theorem 1. The transition function M of *Definition 4* is extended from $S \times \Sigma_k$ to $S \times \Sigma_k^*$ if and only if the following conditions are satisfied
(i) $(\forall s)_S\ M(s, \epsilon) = s$;
(ii) $(\forall s)_S\ (\forall \sigma)_{\Sigma_k}\ (\forall x)_{\Sigma_k^*}\ M(s, \sigma x) = M(M(s, \sigma), x)$;
Theorem 2. (i) $(\forall s)_S\ (\forall x, y)_{\Sigma_k^*}\ M(s, xy) = M(M(s, x), y)$.
(ii) $(\forall s)_S (\forall x, y, z)_{\Sigma_k^*}\ M(s, x) = M(s, y) \Rightarrow M(s, xz) = M(s, yz)$.
Definition 6. The response function of a sequential machine $S = \langle S, \Sigma_k, M, a\rangle$ is denoted by rp_S, a function from $\Sigma_k^* \rightarrow S$, and is defined by $rp_S(x) = M(a, x)$.
Definition 7. The response tree of a sequential machine $S = \langle S, \Sigma_k, M, a\rangle$ is defined to be a tree over Σ_k having a root such that the root has outdegree k and every other vertices has indegree 1 and outdegree k. There are k^h vertices at height h which are numbered $0, 1, 2, ..., k^h - 1$; where the vertices are assigned labels $rp_S(|(h, w)|)$ where, h is the height and w is the corresponding number of the vertex at that height; $|(h, w)|$ corresponds to a word 'x'.
Definition 8. A state s of a sequential machine $S = \langle S, \Sigma_k, M, a\rangle$ is called accessible if and only if $(\exists x)_{\Sigma_k^*}\ s = rp_S(x)$.
States which are not accessible are redundant and by omitting these redundant states we obtain a more useful sub-machine, called the connected sub-machine.
Definition 9. Connected sub-machine of a given sequential machine $S = \langle S, \Sigma_k, M, a\rangle$ denoted by S^C is defined to be the machine $S^C = \langle S^C, \Sigma_k, M', a\rangle$ where, S^C is the set of all accessible states of S i.e., $S^C = \{s \in S |\ (\exists x)_{\Sigma_k^*}\ s = rp_S(x)\}$ and M' is the restriction of M from $S \times \Sigma_k \rightarrow S$ to $S^C \times \Sigma_k \rightarrow S^C$ given by $(\forall s)_{S^C}\ (\forall \sigma)_{\Sigma_k}\ M'(s, \sigma) = M(s, \sigma)$. Accordingly, a machine S is said to be connected if and only if $S = S^C$. To check the connectivity we do not need to search for infinite length words.
Theorem 3. Let $S = \langle S, \Sigma_k, M, a\rangle$ be a finite machine with n states. Then $(\forall s)_{S^C}\ (\exists x)_{\Sigma_k^*}\ s = rp_S(x)$ and $lg(x) < n$.
Definition 10. ϕ is called a homomorphism from the machine $S =$

$\langle S, \Sigma_k, M, a\rangle$ into (onto) another machine $T = \langle T, \Sigma_k, N, b\rangle$ if and only if ϕ is a homomorphism from the equivalent abstract algebra of S into (onto) the equivalent abstract algebra of T with preservation of initial states i.e., if and only if ϕ is a homomorphism from S into (onto) T such that $(\forall s)_S\ (\forall \sigma)_{\Sigma_k}\ \phi(M(s,\sigma)) = N(\phi(s),\sigma)$ and $\phi(a) = b$.

Theorem 4. If ϕ is a homomorphism from $S = \langle S, \Sigma_k, M, a\rangle$ into (onto) $T = \langle T, \Sigma_k, N, b\rangle$ then $(\forall s)_S\ (\forall x)_{\Sigma_k^*}\ \phi(M(s,x)) = N(\phi(s), x)$.

Theorem 5. If ϕ is a homomorphism from S into (onto) T then $(\forall x)_{\Sigma_k^*}\ \phi(rp_S(x)) = rp_T(x)$.

Theorem 6. A mapping ϕ from a connected machine $S = \langle S, \Sigma_k, M, a\rangle$ into (onto) a machine $T = \langle T, \Sigma_k, N, b\rangle$ is a homomorphism if and only if $(\forall x)_{\Sigma_k^*}\ \phi(rp_S(x)) = rp_T(x)$.

This result at once proved the validity of testing the existence of a homomorphism ϕ from two given finite connected machines S and T.

Definition 11. R is called a congruence relation on a sequential machine $S = \langle S, \Sigma_k, M, a\rangle$ if and only if R is an equivalence relation on S which has substitution property $(\forall u, v)_S\ (\forall \sigma)_{\Sigma_k}\ uRv \rightarrow M(u,\sigma)R\ M(v,\sigma)$. For testing a given equivalence relation R on a finite machine $S = \langle S, \Sigma_k, M, a\rangle$ to be a congruence relation we simply partition S as induced by R. Let $R = \{A_1, A_2, ..., A_m\}$. Then for any fixed j we consider the equivalence class A_j. We take two elements $u, v \in A_j$ and test u against v, i.e., we look whether $M(u,\sigma_i)$ and $M(v,\sigma_i)$ belong to the same equivalence class for $i = 0, 1, 2, ..., k-1$. We do the same for all other elements $w \in A_j$. If any check fails we conclude that R is not a congruence relation. Otherwise we conclude that R is a congruence relation.

Definition 12. Let R be a congruence relation on a sequential machine $S = \langle S, \Sigma_k, M, a\rangle$. The quotient machine of S modulo R is a sequential machine denoted by S/R and is defined by $S/R = \langle T, \Sigma_k, N, b\rangle$ where $T = \{R[s],$ class induced by R represented by $s | s \in S\}$ and $(\forall s)_S\ (\forall \sigma)_{\Sigma_k}\ N(R[s],\sigma) = R[M(s,\sigma)]$ and $b = R[a]$.

Theorem 7. *Definition 12* implies that $(\forall s)_S\ (\forall x)_{\Sigma_k^*}\ N(R[s], x) = R[M(s,x)]$.

Theorem 8. i) $rp_{S/R}(x) = R[rp_S(x)]$.

ii) If S is connected then S/R is also so.

Definition 13. Let ϕ be a homomorphism from a machine S into a machine T. Then we define a relation R_ϕ on S by $sR_\phi s'$ if and only if $\phi(s) = \phi(s')$. Clearly R_ϕ is a congruence relation on S.

Theorem 9. Let ϕ be a homomorphism from a machine S onto a machine T and R_ϕ be defined as in *Definition 13*. Then T is isomorphic to S/R_ϕ.
Theorem 10. If R is a congruence relation on a machine S then there exists a homomorphism from S onto S/R.
Definition 14. The equiresponse relation of a sequential machine S to be denoted by $\rho(S)$ is a relation on Σ_k^* defined by $x\rho(S)y$ if and only if $rp_S(x) = rp_S(y)$.
Theorem 11. $\rho(S)$ is a congruence relation on Σ_k^*.
Theorem 12. For any sequential machine S, $T(\rho(S))$ is isomorphic to S^C where $T(\rho(S)) = S/\rho(S)$.
Theorem 13. For any congruence relation R on Σ_k^*, $\rho(T(R)) = R$.
Theorem 14. Let R_1 and R_2 be two congruence relations on Σ_k^*. There exists a homomorphism from $T(R_1)$ onto $T(R_2)$ if and only if $R_1 \subseteq R_2$.
Theorem 15. There exists a homomorphism from a connected machine S onto a connected machine T if and only if $\rho(S) \subseteq \rho(T)$.
Theorem 16. Two connected machines S and T are isomorphic if and only if $\rho(S) = \rho(T)$.

4 Sequential machine with output

Definition 15. A sequential machine with output or a Mealy machine is a six-tuple $S = \langle S, \Sigma_k, \Delta_l, M, Z, a\rangle$ where $S = \langle S, \Sigma_k, M, a\rangle$ is a sequential machine without output; Δ_l is the output alphabet and Z is a function from $S \times \Sigma_k \to \Delta_l$.
A Mealy machine may not have an initial state. S is called a finite state machine if the set S is finite. If $S = \{s_0, s_1, \cdots, s_{n-1}\}$ then the functions M and Z are given by transition tables in which the ordered pair $(M(s_i, \sigma_j), Z(s_i, \sigma_j))$ appeared $s_i \in S$ rowwise and $\sigma_j \in \Sigma_k$ columnwise. In the Moore Graph representation, node s_i is joined to node s_k by means of a directed line labelled $\sigma_j|\delta_p$ if and only if $s_k = M(s_i, \sigma_j)$; $\delta_p = Z(s_i, \sigma_j)$.
If a Mealy machine is such that its output function Z depends only on the internal states and not on the inputs explicitly then the machine is called a Moore machine.
Definition 16. A Moore machine is a six-tuple $S = \langle S, \Sigma_k, \Delta_l, M, Z, a\rangle$ where $\langle S, \Sigma_k, M, a\rangle$ is a sequential machine without output; Δ_l is the output alphabet and $Z : S \to \Delta_l$ is the output function. A Moore machine too may

not have a specified initial state.
Remark 1. Although a Moore machine appears restricted it can be shown that every Mealy machine may be represented by a Moore machine.
Theorem 17. Given a Mealy machine (without initial state) $S = \langle S, \Sigma_k, \Delta_l, M, Z\rangle$, we can construct a Moore machine $S' = \langle S', \Sigma_k, \Delta_l, M', Z'\rangle$ such that S' determines S uniquely.
Remark 2. The Moore machine is theoretically simpler to study but has a practical disadvantage namely that the size of the equivalent Moore machine increases k-times. If in a Moore machine we restrict the output alphabet to $\Delta_2 = \{0, 1\}$ which does not however, curtail the generality to a great extent, the output function Z may be replaced by a set F defined by $F = \{s \in S | Z(s) = 1\}$.
An input word is said to be recognized by the machine if the final output after feeding the input is 1, i.e., the final state resulting from feeding the input belongs to F. Thus, formally we define an automaton in the following.

4.1 Finite state automaton

Definition 17. An automaton is a five-tuple $S = \langle S, \Sigma_k, M, a, F\rangle$ where $S = \langle S, \Sigma_k, M, a\rangle$ is a sequential machine without output and F is a given subset of S called the set of final states of S or the output set. An automaton is called finite if it has a finite number of internal states. A finite automaton may be represented by a Moore Graph of the associated sequential machine without output together with a list of elements of F.
Remark 3. Properties of automata which are not connected to the output set are exactly the same as for the associated machine without output. Thus, the concept of transition function, response function, equi-response relation, connectedness, congruence relation will be taken over unchanged from these results of sequential machines without output.
Definition 18. A word or tape x is said to be recognized or accepted by an automaton $S = \langle S, \Sigma_k, M, a, F\rangle$ if and only if $rp_S(x) \in F$.
Definition 19. The behaviour of an automaton $S = \langle S, \Sigma_k, M, a, F\rangle$ is the set of all words recognized by S denoted by β_S and is defined by $\beta_S = \{x \in \Sigma_k^* | rp_S(x) \in F\}$.
Definition 20. Two automata are said to be behaviourally equivalent, written as $S \equiv T$ if and only if $\beta_S = \beta_T$.
Theorem 18. Behavioural equivalence is an equivalence relation on the set

of all machines.

Definition 21. A connected sub automaton of a given automaton $S = \langle S, \Sigma_k, M, a, F\rangle$ is denoted by S^C and is defined by $S^C = \langle S^C, \Sigma_k, M^C, a, F^C\rangle$ where $\langle S^C, \Sigma_k, M^C, a\rangle$ is a connected sub-machine of the associated sequential machine without output $\langle S, \Sigma_k, M, a\rangle$ and $F^C = F \cap S^C$.

Theorem 19. $\beta_{S^C} = \beta_S$.

Definition 22. ϕ is a homomorphism from an automaton $S = \langle S, \Sigma_k, M, a, F\rangle$ into (onto) an automaton $T = \langle T, \Sigma_k, N, b, G\rangle$ if and only if ϕ is a homomorphism from the associated sequential machine without output $\langle S, \Sigma_k, M, a\rangle$ into (onto) the associated sequential machine without output $\langle T, \Sigma_k, N, b\rangle$ and $(\forall s)_S \ \ if \ \ s \in F \ \ then \ \ \phi(s) \in G$.

Definition 23. ϕ is called a strong homomorphism from an automaton $\langle S, \Sigma_k, M, a, F\rangle$ into (onto) an automaton $\langle T, \Sigma_k, N, b, G\rangle$ if and only if ϕ is a homomorphism from $\langle S, \Sigma_k, M, a, \rangle$ into (onto) $\langle T, \Sigma_k, N, b\rangle$ without output and $(\forall s)_S \ s \in F$ if and only if $\phi(s) \in G$.

Definition 24. An isomorphism from S into T is a one-to-one strong homomorphism from S into T. Automaton S is said to be isomorphic to automaton T if and only if there exists an isomorphism from S into T.

Definition 25. Let R be an equivalence relation on a set S. Then R is said to refine a set $F \subseteq S$ if $(\forall u, v)_S \ uRv$ then ($u \in F$ if and only if $v \in F$).

Remark 4. A partition of F induced by R is obtained by further partitioning the sets F and $S - F$.

Definition 26. Let R be a congruence relation on an automaton $S = \langle S, \Sigma_k, M, a, F\rangle$ such that R refines F. The quotient automaton of S modulo-R is denoted by S/R and is defined by $S/R = \langle T, \Sigma_k, N, b, G\rangle$ where $\langle T, \Sigma_k, N, b\rangle = \langle S, \Sigma_k, M, a\rangle/R$ and $G = \{R[u] | u \in F\}$.

Theorem 20. $\beta_{S/R} = \beta_S$.

Theorem 21. There exists a strong homomorphism from S onto S/R.

Theorem 22. For any automaton $S = \langle S, \Sigma_k, M, a, F\rangle$, $\rho(S)$ refines β_S.

Let S be a given automaton whose behaviour is β_S. Suppose our problem is to construct an automaton having behaviour β_S and having the fewest possible internal states. We have already noticed that if R is any congruence relation on S such that R refines F, the output of S, then the quotient automaton S/R has the same behaviour as S but has a lesser number of internal states. So in order to compute the minimum state automaton we must look for the largest such congruence relation.

4.2 State Equivalence

Definition 27. Let $S = \langle S, \Sigma_k, M, a, F\rangle$ and $T = \langle T, \Sigma_k, N, b, G\rangle$ be two automata. A state $s \in S$ is said to be equivalent to a state $t \in T$, denoted as $s \equiv t$, if and only if $(\forall x)_{\Sigma_k^*}$ $(M(s,x) \in F$ if and only if$N(t,x) \in G)$. If $s \not\equiv t$ then automata S and T are said to be distinct.

Theorem 23. Connected machine $S = \langle S, \Sigma_k, M, a, F\rangle$ is equivalent to connected machine $T = \langle T, \Sigma_k, N, b, G\rangle$ if and only if $a \equiv b$.

Theorem 24. Let $S = \langle S, \Sigma_k, M, a, F\rangle$ and $T = \langle T, \Sigma_k, N, b, G\rangle$ be two automata in which $s \in S$ and $t \in T$. $s \equiv t$ if and only if $(\forall x)_{\Sigma_k^*}$ $M(s,x) \equiv N(t,x)$.

Definition 28. If we consider equivalence of states of a single automaton $S = \langle S, \Sigma_k, M, a, F\rangle$ then, for convenience, the notation $\equiv$ will be denoted by R_F and is defined by $(\forall u, v)_S$ uR_Fv if and only if $(\forall x)_{\Sigma_k^*}$ $(M(u,x) \in F$ if and only if $M(v,x) \in F)$.

Theorem 25. i) R_F is a congruence relation on S induced by F.
ii) R_F refines F.

Definition 29. R_F is called the congruence relation on S induced by F.

Theorem 26. R_F is the largest congruence relation which refines F.

Definition 30. Let $S = \langle S, \Sigma_k, M, a, F\rangle$ be a given connected automaton and R_F, the congruence relation on S induced by F. The minimal automaton associated with S, denoted as S^M, is defined by $S^M = S/R_F$ i.e. $S^M = \langle S^M, \Sigma_k, M', a, F\rangle$ where $S^M = \{R_F[s] | s \in S\}$ and $(\forall s)_{S^M}$ $(\forall \sigma)_{\Sigma_k}$ $M'(R_F[s], \sigma) = R_F[M(s,\sigma)]$ with $a = R_F[a]$ and $F = \{R_F[u] | u \in F\}$.

Theorem 27. $\beta_{S^M} = \beta_S$.

Theorem 28. There exists a strong homomorphism from S onto S^M.

Theorem 29. Let S and T be two connected automata. $S \equiv T$ if and only if automata S^M and T^M are isomorphic.

Theorem 30. Let S be a given connected automaton and T is an automaton such that $T \equiv S$. Then the number of states of T is greater than or equal to the number of states of S^M i.e., among all automata having behaviour β_S. S^M has the minimum number of states. If T has the same no of states as S^M then T is isomorphic to S^M, i.e., the minimal automaton is unique upto isomorphism.

Theorem 31. If S is a connected automaton then any two states of S^M are distinguishable.

4.3 Regular Set

Definition 31. A set $\beta \in \Sigma_k^*$ is said to be regular if and only if there exists a finite automaton S whose behaviour is β.
Theorem 32. $\beta \in \Sigma_k^*$ is regular if and only if one of the following conditions hold,
(i) The induced congruence relation R_β is of finite rank;
(ii) There exists a congruence relation R on Σ_k^* of finite rank which refines β.
Theorem 33. (i) Φ and Σ_k^* are regular sets.
(ii) If β is a regular set then $\bar{\beta} = \Sigma_k^* - \beta$ is a regular set.
(iii) If β_1 and β_2 are regular sets then $\beta_1 \cup \beta_2$, $\beta_1 \cap \beta_2$ are regular sets.
(iv) If R_k denote the set of all regular sub-sets of Σ_k^* then $R_k = \langle R_k, \cap, \cup, ^- , \Phi, \Sigma_k^* \rangle$ is a Boolean algebra.
Theorem 34. i) $\{\epsilon\}$ is a regular set.
ii) If $x \in \Sigma_k^*$ then $\{x\}$ is a regular set. Any finite sub-set of Σ_k^* is a regular set.
Definition 32. The transpose of a set $\alpha \subseteq \Sigma_k^*$ is denoted by $\alpha^T = \{x^T | x \in \alpha\}$.
Theorem 35. For $\alpha, \beta \subseteq \Sigma_k^*$, $\left(\alpha^T\right)^T = \alpha$; $(\alpha.\beta)^T = \beta^T.\alpha^T$.
Theorem 36. If β is a regular set then β^T is a regular set.
In order to facilitate the study of regular sets we introduce the following generalization of a finite automaton.
Definition 33. A transition system is a six-tuple $S = \langle S, \Sigma_k, M, A, F, P \rangle$ where S is the set of internal states, Σ_k is the input alphabet , M is a function from $S \times \Sigma_k \rightarrow 2^S$, where 2^S is the set of all subsets of S , called the transition function such that for $s \in S$ and $\sigma \in \Sigma_k$, $M(s, \sigma)$ is the set of states at which the machine can go to when in state s and consumes σ as input; A is a non-empty subset of S called the set of initial states, F is a given subset of S , called the set of final states; P is a binary relation on S such that for $u, v \in S$ if uPv then we say that there is a spontaneous transition from u to v and P is called the spontaneous transition relation. A transition system S is called finite if and only if the set S is finite.
A finite state transition system may be represented by a transition table together with the prescription of the sets A, F and the relation P or by means of a transition graph in which vertex u is joined to vertex v by means of a directed line bearing label ϵ iff uPv, the sets A and F being given separately.

Example 1. $S = \langle S, \Sigma_{DNA}, M, A, F, P\rangle$ is a transition system where $S = \{s_0, s_1, s_2, s_3, s_4\}$; $\Sigma_{DNA} = \{a, t, g, c\}$; $A = \{s_0, s_1\}$; $F = \{s_3\}$; $P = \{(s_2, s_3)\}$. The transition table is given by table 1.

	a	t	g	c
s_0	$\{s_3\}$	$\{s_2\}$	Φ	$\{s_1\}$
s_1	Φ	$\{s_3\}$	$\{s_4\}$	Φ
s_2	Φ	Φ	$\{s_0\}$	$\{s_4\}$
s_3	Φ	Φ	Φ	Φ
s_4	$\{s_3, s_0\}$	Φ	Φ	Φ

Table 1: The transition table of the *Example 1.*

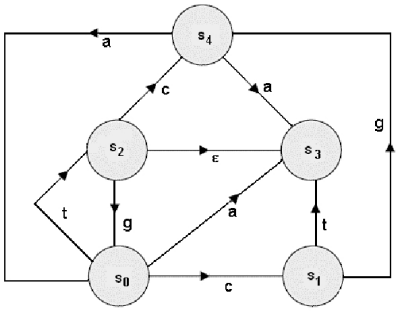

Figure 3: The transition graph corresponding to the transition system S of *Example 1.*

Definition 34. Any (sequential machine) automaton may be recognised as a transition system.

Definition 35. The transition relation of a transition system $S = \langle S, \Sigma_k, M, A, F, P\rangle$ is a relation between three states S, Σ_k and S, i.e., a subset of $S \times \Sigma_k \times S$, denoted by $|u, \sigma, v|$ and is defined by $(\forall u, v)_S\ (\forall \sigma)_{\Sigma_k}\ |u, \sigma, v|$ if and only if $(\exists s, w)_S\ u\widehat{P}s \wedge w \in M(s, \sigma) \wedge w\widehat{P}v$ where $\widehat{P}$ is the reflexive and

transitive closure of P ($\widehat{P} = \bigcup_{i=1}^{\infty} P^i \cup O$).

Definition 36. The response relation of a transition system $S = \langle S, \Sigma_k, M, A, F, P\rangle$ to be denoted by ρ_S is a relation between Σ_k^* and S and is defined by $\rho_S = \{(x, s)|(\exists a)_A|a, x, s|\}$ i.e. $x\rho_S s$ if and only if $(\exists a)_A|a, x, s|$.

Definition 37. A word x is said to be accepted or recognized by a transition system $S = \langle S, \Sigma_k, M, A, F, P\rangle$ if and only if $(\exists u)_F\ x\rho_S u$, i.e., if and only if $(\exists a)_A\ (\exists u)_F\ |a, x, u|$.

Definition 38. The behaviour of a transition system S is defined to be the set of all words recognized by S and is denoted by β_S, i.e., $\beta_S = \{x|(\exists u)_F\ x\rho_S u\} = \{x|(\exists a)_A\ (\exists u)_F|a, x, u|\}$.

Theorem 37. The behaviour of a finite transition system is a regular set.

Definition 39. Regular Expressions are abstract notations used to denote regular sets. Regular expressions over alphabet Σ_k are exactly those expressions that can be constructed recursively from the following rules

i) ϵ is a regular expression;

ii) For each $\sigma \in \Sigma_k$, σ is a regular expression;

iii) If e_1 and e_2 are regular expressions then $e_1|e_2$, e_1e_2 are regular expressions;

iv) If e is a regular expression then e^* is a regular expression.

Each regular expression e over Σ_k^* actually denotes a language, called the valuation of the regular expression and is a regular set denoted by $v(e)$ where

i) $v(\epsilon) = \{\epsilon\}$;

ii) $v(\sigma) = \{\sigma\}$;

iii) $v(e_1|e_2) = v(e_1) \cup v(e_2)$;

iv) $v(e_1e_2) = v(e_1).v(e_2)$;

v) $v(e^*) = \{v(e)\}^*$.

Example 2: Human FMR-1 gene sequence fragment containing a triplet repeat region given by,

... GCG CGG CGG CGG CGG CGG CGG CGG CGG CGG CGG AGG CGG CGG CGG CGG CGG CGG CGG CGG CGG AGG CGG CGG CGG CGG CGG CGG CGG CGG CGG CTG ...

The regular expression representing the pattern would be $gcg((c|a)gg)^*ctg$.

The transition graph corresponding to the regular expression is given in figure 4.

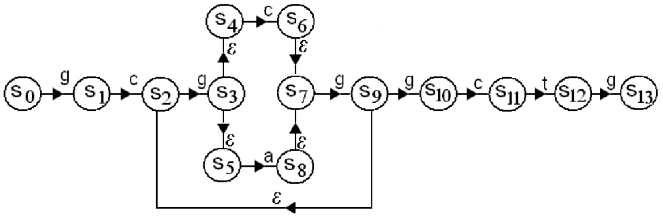

Figure 4: Transition graph of the FMR-1 gene fragment given in *Example 2*.

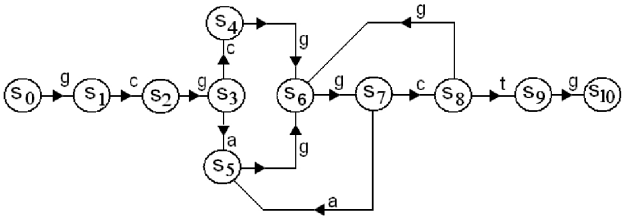

Figure 5: The deterministic finite automata (DFA) representing FMR-1 gene fragment of *Example 2*.

5 Mathematical formulation of Biological mutational operations

DNA sequence specifies the exact genetic instruction required to create a particular organism. Every organism has an inherent tendency to undergo change from one hereditary state to another. Such hereditary change is called mutation. Actually mutation is an alteration in the nucleotide sequence of a DNA molecule. The bases constituting DNA molecules are adenine, thymine, guanine and cytosine and are represented with the symbols a, t, g, c. Let us set $\Sigma_{DNA} = \{a, t, g, c\}$, the alphabet of DNA.

Definition 40. A finite transducer is a 7-tuple $S = \langle S, \Sigma_k, \Delta_l, M, Z, a, F \rangle$

where $\langle S, \Sigma_k, \Delta_l, M, Z, a\rangle$ is a finite state machine with output and F is a finite subset of the set of internal states S, called the set of final states.

Remark 5. Properties of finite transducers which are not connected to the output set are exactly the same as for the associated finite state machine without output. Results of finite transducers which are not connected to the output function Z are exactly the same for the associated finite state automaton. Thus the concepts of extension of definition of M and Z, the response function, equi-response relation, connectedness, congruence relation will be taken over unchanged from these results of sequential machines without output.

Definition 41. An input word $x \in \Sigma_k^*$ is said to be accepted by a finite transducer S if and only if $M(a, x) \in F$ *i.e.*, $rp_S(x) \in F$. The collection of all words x acceptable to a finite transducer is called the behaviour of the transducer.

Definition 42. Let $x \in \Sigma_k^*$ be acceptable to a finite transducer S. Then there correspond an output word $y \in \Delta_l^*$, called the output corresponding to the input x. The collection of all output words (non-determinism) corresponding to the input word 'x' is denoted by $\delta_S(x)$, and is defined as $\delta_S(x) = \{y \in \Delta_L^* | rp_S(x) \in F \wedge Z(a, x) = y\}$.

Definition 43. The sequence of transitions employed to reach a final state starting from the initial state after consuming an input word (string) $x \in \Sigma_k^*$ is called a derivation and is denoted by $D(x)$. When $D(x)$ is unique it is called a deterministic transducer. A transducer which is not deterministic will be called non-deterministic.

Example 3. Let $S = \langle S, \Sigma_k, \Delta_l, M, Z, a, F\rangle$ be a finite transducer. Let S=$\{s_0, s_1, s_2\}$, Σ_2=$\{\sigma_0, \sigma_1\}$, Δ_2=$\{\delta_0, \delta_1\}$, and $a = \{s_0\}$ and $F = \{s_2\}$. The state transition matrix and the output are as shown in the table 2.

M=

	ϵ	σ_0	σ_1
s_0	s_0	s_1	s_2
s_1	s_0	s_2	s_0
s_2	$-$	s_2	s_2

Z=

	ϵ	σ_0	σ_1
s_0	$-$	δ_0	δ_1
s_1	$-$	δ_1	δ_1
s_2	$-$	δ_0	δ_0

Table 2: The state transition matrix and the output of *Example 3*.

$\delta_S = \bigcup_{x \in \Sigma_k^*} \delta_S(x)$ is called the derivation of S. Thus $y \in \delta_S \rightarrow (\exists x)_{\Sigma_k^*}$ s.t. $y \in \Delta_l^* \wedge rp_S(x) \in F \wedge Z(a, x) = y$.

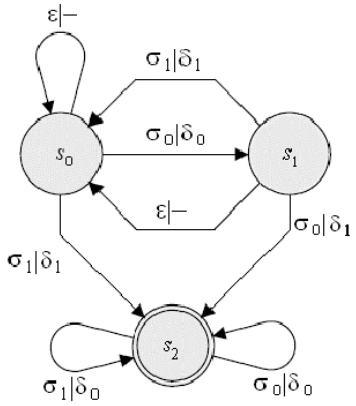

Figure 6: Transition diagram of the finite transducer of *Example 3*.

It is easy to see that the word $\sigma_0\sigma_1\sigma_1$ is acceptable as $M(s_0, \sigma_0\sigma_1\sigma_1) = s_2 \in F$. Whereas the set of words represented by the regular expression $(\sigma_0\sigma_1)^*$ is not acceptable. From the transition table the response function can be represented by means of a response tree as in the figure 5. Here we have $Z(s_0, \sigma_0\sigma_1^2) = \delta_0\delta_1^2$.
The set of all words acceptable to the finite transducer will be given by the valuation of the regular expression $(\sigma_1|\sigma_0^2|(\sigma_0|\sigma_0\sigma_1)(\sigma_1|\sigma_0^2))(\sigma_1|\sigma_0)^*$ and is the derivation.
$D(\sigma_1) = \{s_2\}$; $D(\sigma_0^2) = \{s_1s_2\}$; $D(\sigma_0^3) = \{s_1s_0s_1s_2, s_1s_2s_2\}$; ... etc.
$\delta_S(\sigma_0) = \{\delta_0\}$; $\delta_S(\sigma_1) = \{\delta_1\}$; $\delta_S(\sigma_0^2) = \{\delta_0\delta_1\}$; $\delta_S(\sigma_0^3) = \{\delta_0\delta_1^2\}$; $\delta_0\delta_1^* \in \delta_S$.
Definition 44. A weighted finite transducer is a finite transducer $S = \langle S, \Sigma_k, \Delta_l, M', Z, a, F\rangle$ where $\langle S, \Sigma_k, \Delta_l, Z, a\rangle$ is the same as in *Definition 41* and the transition function $M' : S \times \Sigma_k \rightarrow S \times U$; where $U \subset \Re$, the real number set. More explicitly, $M'(s, \sigma) = (M(s, \sigma), u)$.
This number associated with a state transition is called the weight of the transition. Accordingly, the weight of a derivation is the sum of weights of

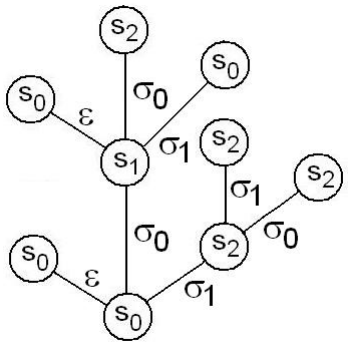

Figure 7: Response Tree of *Example 3*.

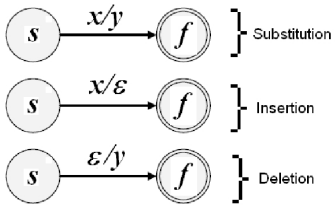

Figure 8: Mechanisms using finite state machines.

the transitions employed.

6 A case study with FMR-1 gene

The degree to which the genetic factors influence human intelligence remains a matter of controversy. Mutation affecting the FMR1 gene cause the fragile X syndrome, the most prevalent known inherited cause of intellectual dysfunction. The most common mutation occurring in the FMR1 locus involves expansion of a triplate $(\texttt{CGG})_n$ repeat sequence within the promoter region of the gene. When more than 200 `CGG` repeats are present, the expanded repeat sequence and an adjacent CpG island are usually hypermethylated, a phe-

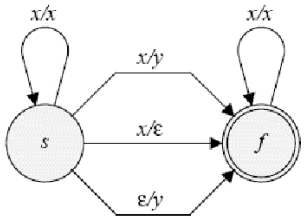

Figure 9: Transducer corresponding to biological mutational activities.

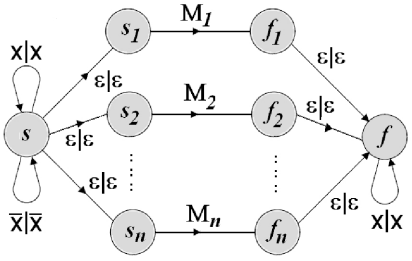

Figure 10: The mutator corresponding to the mechanisms $M_1, M_2, ..., M_n$.

nomenon associated with transcriptional silencing of the gene and commonly referred to as the FMR1 full mutation. Most males with the FMR1 full mutation function show mentally retarded range of intelligence; in contrast, females with the FMR1 full mutation show a broader range of intelligence, from mental retardation to normal intelligence. Despite differnces in severity of intellectual dysfunction, both males and females with the FMR1 full mutation manifest a similar cognitive profile with weakness in the visual-spatial and attentional-organizational domains and relatively preserved verbal abilities (Reiss et al. 1995, Reyniers et al. 1993, Fu et al. 1991).

The function $\delta(\texttt{CGG})= 0+0+1=1$ may be used as a measure of the occurrence of the triplet `CGG`. The M and Z functions of the DFA of FMR-1 gene (figure 16) has been given in table 3.

Definition 45. The mutation ratio, denoted as μ, is defined as a measure of the number of occurrence of a subsequence of length (say) m in a sequence

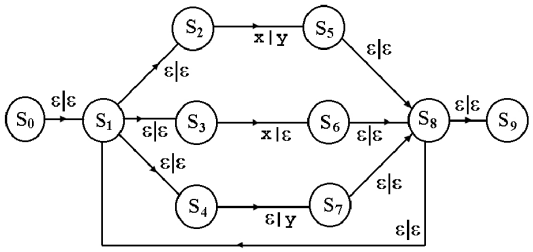

Figure 11: The mutator corresponding to the biological mutational mechanisms.

M=

	a	t	g	c
s_0	s_0	s_0	s_0	s_1
s_1	s_0	s_0	s_2	s_1
s_2	s_0	s_0	s_3	s_1
s_3	$-$	$-$	$-$	$-$

Z=

	a	t	g	c
s_0	0	0	0	0
s_1	0	0	0	0
s_2	0	0	1	0
s_3	0	0	0	0

Table 3: The M and Z functions of the DFA to calculate the occurrence of the motif `CGG` given in *Figure 13*. The output calculates the occurrence of the motif.

of length (say) n. Specifically, if r occurrence of a subsequence of length m is observed in a sequence of length n, then $\mu = \frac{rm}{n}$. Obviously, $0 \leq \mu \leq 1$. Accordingly, the value of μ for the subsequences `CGG` and `AGG` are calculated for the FMR-1 gene of different species is represented in table 5. The count shows a difference between the normal and the genomic mutant.

7 Conclusion

Our aim was to develop a mathematical machine that simulates biological mutational operations. We have developed a Mutator that corresponds to a finite state sequential machine with output and then the operations were described by means of transition diagram. We have developed a theory to

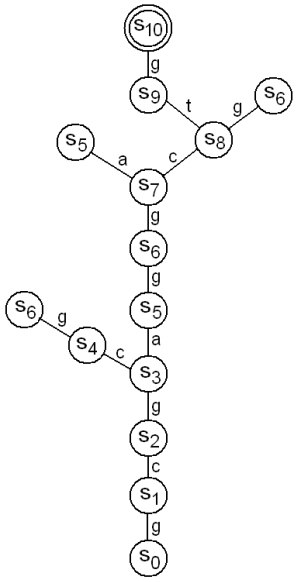

Figure 12: The tree corresponding to the FMR-1 gene.

obtain the minimal state automata. We have studied the FMR-1 gene. A DFA is constructed with the gene sequence. The mutation ratio is studied for different species. We hope that this will help us to model different concepts of biomolecular sequence analysis.

Acknowledgement

The research has been partially supported by the UGC-SAP DRS Phase-I project under the Department of Mathematics, Visva-Bharati. The authors also thank to Abul Hossain, Hirak Kumar Bandyopadhyay and Dr. Arani Chakraborty of Visva-Bharati for their valuable support in preparing the manuscript. The authors also thank to Dr. Leen Torenvliet and Dr. Jaap Kandoorp of Universiteit van Amsterdam for the long discussions and comments on the manuscript.

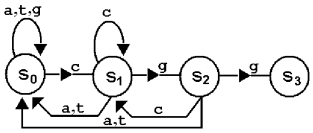

Figure 13: The DFA to calculate the occurrence of the motif `CGG`. A counter is added at the state s_3 to calculate the occurrence of the motif.

M=

	a	t	g	c
s_0	–	–	s_1	–
s_1	–	–	–	s_2
s_2	–	–	s_3	–
s_3	s_5	–	–	–
s_4	–	–	s_6	–
s_5	–	–	s_6	–
s_6	–	–	s_7	–
s_7	s_5	–	–	s_8
s_8	–	s_9	s_6	–
s_9	–	–	s_{10}	–

Z=

	a	t	g	c
s_0	0	0	0	0
s_1	0	0	0	0
s_2	0	0	0	0
s_3	0	0	0	0
s_4	0	0	0	0
s_5	0	0	0	0
s_6	0	0	1	0
s_7	0	0	0	0
s_8	0	0	0	0
s_9	0	0	0	0

Table 4: The M and Z functions of the DFA of FMR-1 gene given in *Figure 16*.

References

[1] Allison, L., Wallace, C.S., Yee, C.N. 1992. Finite state models in the alignment of macromolecules. *J. Molecular Evolution*. 35(1),77-89.

[2] Arbib, M.A. 1970. *Theories of abstract automata*. Prentice Hall, Englewood Cliffs, NJ.

[3] Brown, T.A. 1999. *Genomes*. John Wiley & Sons (Asia) Pte. Ltd.

[4] Durbin, R., Eddy, S., Krogh, A., Mitchison, G. 1998. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press, UK.

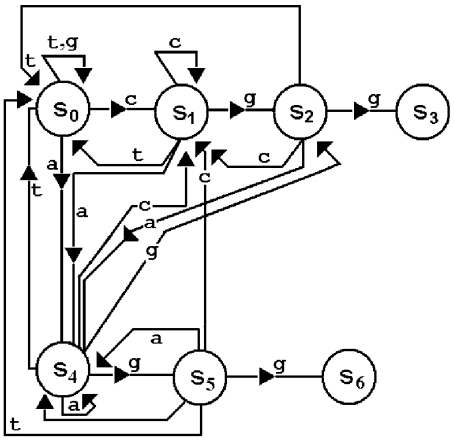

Figure 14: The DFA to calculate the occurrence of the motif `CGG` and as well as the mutated motif `AGG`. A counter may be added at the state s_3 for `CGG` and at the state s_6 for `AGG`.

[5] Ginsburg, S. 1962. Examples of abstract machines. *IEEE Transactions on Electronic Computers*. 11(2), 132-135.

[6] Fu, Y.H., et al. 1991. Variation of the CGG repeat at the fragile X site results in genetic instability: resolution of the Sherman paradox. *Cell* 67, 1047-1058.

[7] Goodman, M.F. 1998. Purposeful mutations. *Nature*. 395, 221-223.

[8] Griffiths, A.J.F., Miller, J.H., Suzuki, D.T., Lawontin, R.C., Gelbart, W.M. 2000. *An introduction to genetic analysis*. W.H. Freeman and Company, NY.

[9] Head, T. 1987. Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviours. *Bulletin of Mathematical Biology*. 19(6), 737-759.

[10] Hopcroft, J.E., Ullman, J.D. 1979. *Introduction to automata theory, Languages, and Computation*. Addison-Wesley Publishing Company Inc. USA.

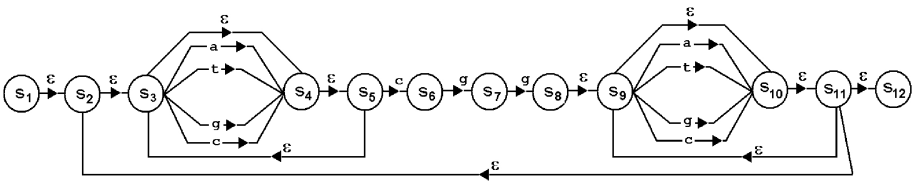

Figure 15: The NDFA of the FMR-1 gene.

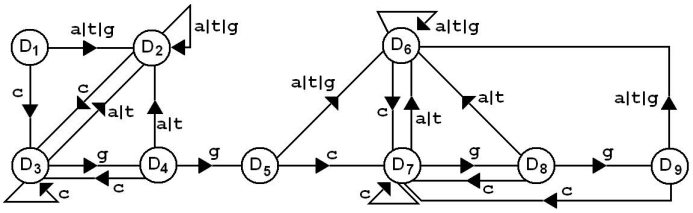

Figure 16: The DFA of the FMR-1 gene. Where $D_1 = \{s_1, s_2, s_3, s_4, s_5\}$, $D_2 = \{s_3, s_4, s_5\}$, $D_3 = \{s_3, s_4, s_5, s_6\}$, $D_4 = \{s_3, s_4, s_5, s_7\}$, $D_5 = \{s_3, s_4, s_5, s_8, s_9, s_{10}, s_{11}, s_{12}\}$, $D_6 = \{s_2, s_3, s_4, s_5, s_9, s_{10}, s_{11}, s_{12}\}$, $D_7 = \{s_2, s_3, s_4, s_5, s_6, s_9, s_{10}, s_{11}, s_{12}\}$, $D_8 = \{s_2, s_3, s_4, s_5, s_7, s_9, s_{10}, s_{11}, s_{12}\}$, $D_9 = \{s_2, s_3, s_4, s_5, s_8, s_9, s_{10}, s_{11}, s_{12}\}$.

[11] Hunter, L. 1993. *Artificial intelligence and molecular biology.* AAAI Press.

[12] Reyniers, E., Vits, L., Boulle, K.D, Roy, B.V., Velzen, D.V., Graff, E.D., Verkerk, A.J.M.H., Jorens, H.Z.J., Darby, J.K., Oostra, B., Willems, P.J. 1993. The full mutation in the FMR-1 gene of male fragile X patients is absent in their sperm. *Nature Genetics.* 4, 143-146.

[13] Reiss, A.L., Freund, L.S., Baumgardner, T.L., Abrams, M.T., Denckla, M.B. 1995. Contribution of the FMR1 gene mutation to human intellectual dysfunction. *Nature Genetics.* 11, 331-334.

[14] Searls, D.B. 1992. The linguistics of DNA. *American Scientist.* 80(6), 579-591.

GenBank Accession Number	*Species*	μ_{CGG}	μ_{AGG}
NM_002024	*Homo sapiens* (Human)	0.03438	0.05776
XM_584093	*Bos taurus* (Cattle)	0.03033	0.06129
XM_001368509	*Monodelphis domestica* (Gray short-tailed opossum)	0.01657	0.05178
AJ875178	*Coniochaeta tetraspora*	0.06834	0.06151
S73754	*Homo sapiens* (Fragile X mental retardation syndrome patient)	0.12852	0.07631

Table 5: The values of the *mutation ratio* μ for the occurrences of the subsequences `CGG` and `AGG` in the FMR-1 gene for different species.

[15] Searls, D.B., Murphy, K.P. 1995. Automata theoretic models of mutation and alignment. *Proceedings of International Conference on Intelligent Systems in Molecular Biology*. 3, 341-349.

[16] Searls, D.B. 1995. String variable grammar: a logic grammar formalism for the biological language of DNA. *Journal of Logic Programming*. 21(1-2).

[17] Searls, D.B. 1999. Formal language theory and biological macromolecules. *Series in discrete mathematics and theoretical computer science*. 47, 117-140.

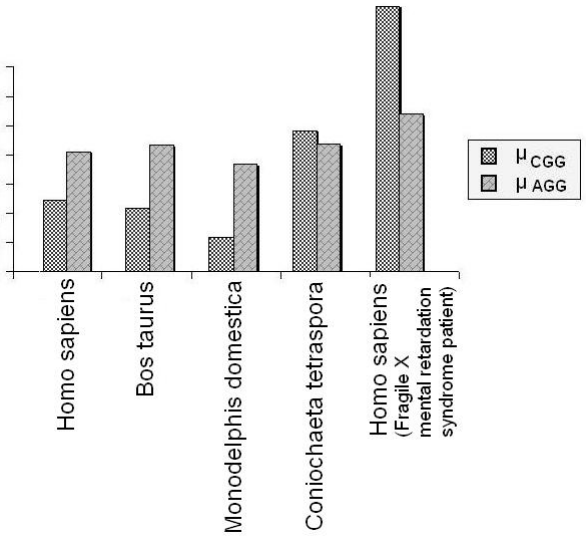

Figure 17: The μ count plot for different species and human genomic mutant for FMR1 gene.