IDENTIFICATION AND ANALYSIS OF FRAMESHIFT MUTATIONS AND THEIR DISEASE SPECIFIC CONSEQUENCES

Thesis submitted in partial fulfillment of the Degree of

Bachelor of Technology

In

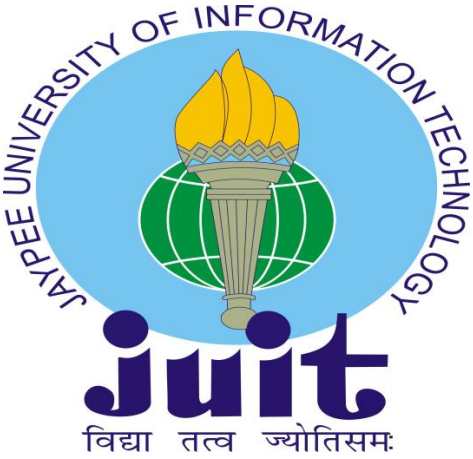
Bioinformatics

By

Jyoti Thakur (101508)

Under the Supervision of

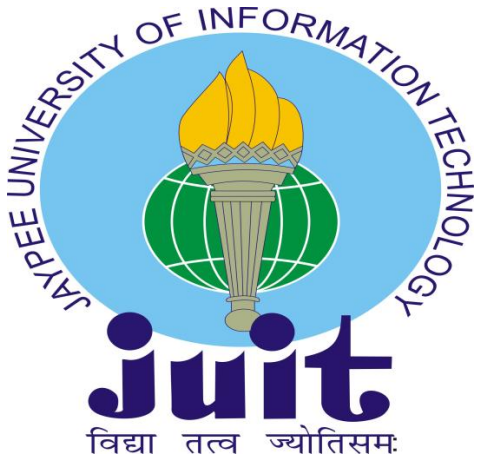
Dr. Tiratha Raj Singh



Department of Biotechnology/Bioinformatics
Jaypee University of Information Technology
Waknaghat, Solan – 173234, Himachal Pradesh
2013-2014

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

WAKNAGHAT, HIMACHAL PRADESH



CERTIFICATE

This is to certify that project report entitled "**Identification and analysis of frame-shift mutation and their disease specific consequences**", submitted by **Jyoti Thakur(101508)** in partial fulfilment for the award of degree of Bachelor of Technology in Bioinformatics of Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

_____

Date:

Dr. Tiratha Raj Singh
Assistant Professor,

Dept. of BT/BI, JUIT

ACKNOWLEDGEMENT

I am very thankful and express my deepest gratitude to **Dr. Tiratha Raj Singh**, Assistant Professor, Department of Biotechnology and Bioinformatics, Jaypee University of Information Technology, Waknaghat under whose supervision and guidance this work has been carried out. His whole hearted involvement, advice, support and constant encouragement throughout, have been responsible for carrying out this project work with confidence. I am thankful to him for showing confidence in me to take up this project. It was due to his planning and guidance that I am able to complete this project on time.

I like to pay my most sincere thanks to **Prof. R. S. Chauhan**, Dean, Biotechnology and Head of Department, Department of Biotechnology and Bioinformatics, for providing me with opportunities and facilities to carry out the project.

I also thank **Mrs. Somlata Sharma** (Bioinformatics Laboratory Incharge) for providing her full cooperation with keen interest. I am indebted to other faculty members and my friends and all those who provided their reviews and suggestions for improvising my project.

Lastly I want to thank my friends and family for being there and supporting me throughout.

DATE: Jyoti Thakur (101508)

ABSTRACT

Background

Frameshift is one of the three classes of recoding. Frame-shifts lead to waste of energy, resources and activity of the biosynthetic machinery. In addition, some peptides synthesized after frame-shifts are probably cytotoxic which serve as plausible cause for innumerable number of diseases and disorders such as muscular dystrophies, lysosomal storage disorders, and cancer. Hidden stop codons occur naturally in coding sequences among all organisms. These codons are associated with the early termination of translation for incorrect reading frame selection and help to reduce the metabolic cost related to the frame-shift events. Researchers have identified several consequences of hidden stop codons and their association with myriad disorders. However the wealth of information available is speckled and not effortlessly acquiescent to data-mining. There are lots od appearances of hidden stops in mitochondrial genomes and we tried to study this putative event in mitochondrial genomes of vertebrates. To reduce this gap, this work presents an algorithmic web based tool to study hidden stops in frame-shifted translation for vertebrate mitochondrial genomes through respective genetic code system.

FrameOUT (FO)

FO is an algorithmic web based application tool that provides a user-friendly interface for prediction of mutations in a user input sequence, be it a diseased or a normal sequence. This mutation probability calculation of user input sequence is done by implementing Hidden Markov Model (which is equivalent to stochastic regular grammars), particularly HMM Forward Algorithm, in which we calculate the probability based on certain training set.

FrameOUT DB (FODB)

FODB is a collection of all available Frameshift events and their association with various diseases. This information has been collected and curated manually from various resources and available literature.

Availability FrameOUT and FrameOUT DB is available through a user friendly web based application for academic and research purpose at:

- http:\\www.bioinfoindia.org\frameout

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS AND TERMS USED

1. **HMM:** Hidden Markov Model

2. **FO:** FrameOUT

3. **FODB:** FrameOUT Database

4. **HSC:** Hidden Stop Codons

5. **Indel:** Term for the insertion or the deletion of bases in the DNA of an organism.

6. **PMID:** PubMed ID

7. **ND1:** NADH dehydrogenase

8. **ND2:** NADH dehydrogenase 2

9. **COX1:** Cytochrome c oxidase I

10. **COX2:** Cytochrome c oxidase 2

11. **ATP8:** ATPase8

12. **COX3:** Cytochrome c oxidase 3

13. **ND3:** NADH dehydrogenase 3

14. **ND4L:** NADH-ubiquinone oxidoreductase chain 4L

15. **ND4:** NADH dehydrogenase 4

16. **ND5:** NADH dehydrogenase 5

17. **ND6:** NADH dehydrogenase 6

18. **CYTB:** Cytochrome b

DEDICATION

This project is dedicated to my project guide & mentor Dr. Tiratha Raj Singh for teaching and guiding me in every part of this project.

CONTENTS

CHAPTER 1: SCOPE & OBJECTIVE

Frameshift is one of the three classes of recoding. Frame-shifts lead to waste of energy, resources and activity of the biosynthetic machinery. In addition, some peptides synthesized after frame-shifts are probably cytotoxic which serve as plausible cause for innumerable number of diseases and disorders such as muscular dystrophies, lysosomal storage disorders, and cancer. Hidden stop codons occur naturally in coding sequences among all organisms. These codons are associated with the early termination of translation for incorrect reading frame selection and help to reduce the metabolic cost related to the frame-shift events. Researchers have identified several consequences of hidden stop codons and their association with myriad disorders [1, 7].

However the wealth of information available is speckled and not effortlessly acquiescent to data-mining. To reduce this gap, this work describes an algorithmic web based tool to study hidden stops in frame-shifted translation for vertebrate mitochondrial genome through respective genetic code system. Also it helps to predict the mutational probability for various associated coding regions and thereby helps to analyze in which region the probability of mutation is maximum. As there are thirteen protein coding genes in vertebrate mitochondrial genomes, input data was trained on these sequences to predict the mutability of query sequences and their frameshift consequences towards their likeliness for these protein coding genes.

Additionally we planned to compile information on various frameshift events and their direct consequences leading to many diseases. By enlisting various diseases due to frameshift events it can serve as a classification method based on various traits. The information being collected manually form the literature and available resources will be of utmost use to the scientific community.

CHAPTER 2: INTRODUCTION

Reading frames play an important role in the process of translation of nucleotide sequences into proteins. Selection of a wrong reading frame can alter the protein product. Such events that alter the reading frame occur extremely rarely during translation; Frame-shift is one such event. Frame-shift is quite common in viruses and also occurs in bacteria, yeast and other organisms [8, 17]. It's a type of genetic mutation caused generally by indels, i.e. insertion and deletion of nucleotides. Coding sequences lack stop codons but myriad of stop codons materialize off-frame. Off-frame stops i.e. stop codons in +1 and −1 shifted reading frames, are termed as hidden stop codons or hidden stops (HSCs) [24]. Frame-shifts lead to the waste of energy, resources and activity of the biosynthetic machinery. In addition, some peptides synthesized after frame-shifts are probably cytotoxic which serve as possible cause for innumerable number of diseases and disorders such as muscular dystrophies, lysosomal storage disorders, and cancer. Frame-shift mutations might be beneficial sometime such as a frame-shift mutation was responsible for the creation of Nylonaser [1, 4, and 7].

2.1 HIDDEN MARKOV MODEL (HMM)

A **hidden Markov model** (**HMM**) is a statistical Markov model in which the system being modelled is assumed to be a Markov process with unobserved (*hidden*) states. A HMM could be considered the simplest dynamic Bayesian network. In simpler Markov models (like a Markov chain), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a Hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. Note that the adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; even if the model parameters are known exactly, the model is still 'hidden' [5, 22].

HMM could be considered a generalization of a mixture model where the hidden variables (or latent variables), which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other [5, 22].

2.1.1 FORWARD HMM

The **forward algorithm**, in the context of a hidden Markov model, is used to calculate a 'belief state': the probability of a state at a certain time, given the history of evidence. The process is also known as *filtering*. The forward algorithm is closely related, but distinct from, the Viterbi algorithm. The forward algorithm takes advantage of the conditional independence rules of the hidden Markov model (HMM) to perform the calculation recursively [5, 25].

2.2 PROBLEM DEFINITIONAND MOTIVATION

The topic of my project is to identify and analyze frame-shift mutations and their disease specific consequences. For this purpose several mitochondrial vertebrate genomic sequences are collected, which are having 13 protein coding sequences, namely: ND1, ND2, COX1, COX2, ATP8, COX3, ND3, ND4L, ND4, ND5, ND6 and CYTB.

Based on these protein coding sequences respective transition matrices are being developed. After generating transition matrices I implemented HMM forward algorithm which computes the joint probability, on the sequence entered by the user neglecting the stop codons, because coding regions lack stop codons [18, 23]. There can be various probable states 0, 1 and 2 which are based upon the position of characters in nucleotide sequences for normal translation(0), +1 frameshift (1) and -1frameshift (2), and symbols are our very own nucleotides i.e. A, T, G and C, with equal probabilities.

Therefore, FrameOut (FO) is a tool to predict the mutational events occurring in genomic sequences through frame-shift events. Data is being framed through HMM. The mutation probability calculation of user input sequence is done by implementing Hidden Markov Model (which is equivalent to stochastic regular grammars), particularly HMM Forward Algorithm, in which we calculate the probability based on certain training set.

FrameOut DB (FODB) is a collection of all available Frameshift events and their association with various diseases specifically human diseases such as Corhn’s disease, Rett-Syndrome, and Sandhoff disease, etc [11,13 and14].

2.3 TOOLS & TECHNIQUES

2.3.1 Webpage: HTML, CSS & JS

2.3.1.1 Hypertext Mark-up Language (HTML): HTML is the main mark-up language for creating web pages and other information that can be displayed on the web browser. HTML is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example: <img>. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also known *opening tags* and *closing tags*). In between these tags web designers can add text, further tags, comments and other types of text-based content. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but rather it uses the tags to interpret the content of the page [25].

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and could be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It could embed scripts written in languages such as JavaScript which affect the behaviour of HTML web pages [25].

2.3.1.2 Cascading Style Sheets (CSS): CSS is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a mark-up language. It's most common application is to style web pages written in HTML and XHTML, but the language could also be applied to any kind of XML document, including plain XML, SVG and XUL [26].

CSS is designed primarily to enable the separation of document content (written in HTML or a similar mark-up language) from document presentation, including elements such as the layout, colors, and fonts. This separation could improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design) [25].

2.3.1.3 JavaScript (**JS**): JS is an interpreted computer programming language. As a part of web browsers, implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It has also become common in server-side programming, game development and the creation of desktop applications [25].

JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have different semantics. The key design principles within JavaScript are taken from the self and Scheme programming languages. It's a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles [25].

2.3.2 Application/Tool: PHP & MySQL

2.3.2.1 Hypertext Pre-processor (PHP): PHP is a server-side scripting language designed for web development but can also be used as a general-purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Lerdorfin 1995, the reference implementation of PHP is now produced by The PHP Group. While PHP originally stood for *Personal Home Page*, it now stands for *PHP: Hypertext Pre-processor*, a recursive acronym [6].

PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page: PHP commands could be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone graphical applications [6].

2.3.2.2 MySQL: MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users can use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively being developed by Oracle, and is freely available for use [25].

The official MySQL Workbench is a free integrated environment developed by MySQL AB, which enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL front end, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator). MySQL Workbench is available in two editions, the regular free and open source *Community Edition* which can be downloaded from the MySQL website, and the proprietary *Standard Edition* which extends and improves the feature set of the Community Edition [25].

2.3.3 WAMP: WAMP refers to a set of free (open source) applications, combined with Microsoft Windows, which are commonly used in Web server environments. The WAMP stack provides developers with the four key elements of a Web server: an operating system, database, Web server and Web scripting software. The combined usage of these programs is known as a server stack. In this stack, Microsoft Windows is the operating system (OS), Apache is the Web server, MySOL handles the database components, while PHP, Python, or PERL represents the dynamic scripting languages [25].

CHAPTER: 3 LITERATURE REVIEW

3.1 Review of the problem:

3.1.1 Frame-shift Events

Frameshift is one of the three classes of recoding [1]. Recoding is the reprogramming of mRNA translation by localized alterations in the standard translational rules. Recoding events occur in competition with standard readout of the transcript, and are site-specific. The efficiency of recoding at the site is usually being influenced by stimulatory signals present on the mRNA (*cis*-elements) and in some cases by protein products or other cellular components (*trans*-elements). The three classes of recoding are (1) frame-shifting (2) Bypassing (Hopping) and (3) codon redefinition.

- Frame-shifting at a particular site could yield two protein products from one coding sequence or one protein product from two overlapping open reading frames (ORFs). The known cases of frame-shifting where the product is utilized involve shifts of one base either +1 or −1, but shifts of two bases have been demonstrated in artificial systems.
- Bypassing (hopping) basically occurs when a block of nucleotides within a coding sequence is not translated. Translation is suspended temporarily, ribosomes traverse the coding gap and protein synthesis yielding a single protein.
- Codon redefinition involves site-specific alteration of codon meaning. The altered meaning of a codon could be the redefinition of an initiation codon or stop codon to specify an amino acid [2].

Frame-shifts are defined as protein translations that start not at the first, but either at the second (+1 frame-shift) or the third (−1 frame-shift) nucleotide of the codon [23]. Presumably, most frame-shifts would yield non-functional proteins. Therefore frame-shifts lead to waste of energy, resources and activity of the biosynthetic machinery. In addition, some peptides synthesized after frame-shifts are probably cytotoxic. Coding sequences lack stop codons, but many stop codons appear off-frame. Off-frame stops i.e. stop codons in +1 and −1 shifted reading frames, are termed **Hidden Stop Codons (HSCs)** or hidden stops [2, 3, 20, and 23].

3.1.2 Types of translational errors:

The process of translation elongation is a complex one, and therefore there are potentially many ways the process can go away. Formally, there are mainly two possible kinds of elongation errors:

3.1.2.1 Missense Errors which results in the substitution of one amino acid for another (termination codon read through is a special one of this type), and

3.1.2.2 Processivity Errors which results in a truncated and usually non-functional protein. Processivity errors are of two types: premature termination of translation, and translational frame-shifting.

- **Premature termination,** may also occur if peptide release factor were to inaccurately recognize a sense codon as a terminator.
- **Translational frame-shifting**, affects processivity because it precludes completion of the nascent peptide chain in the normal reading frame and also, because ribosomes usually encounter a termination codon rather soon in the shifted frame [3, 9, 17].

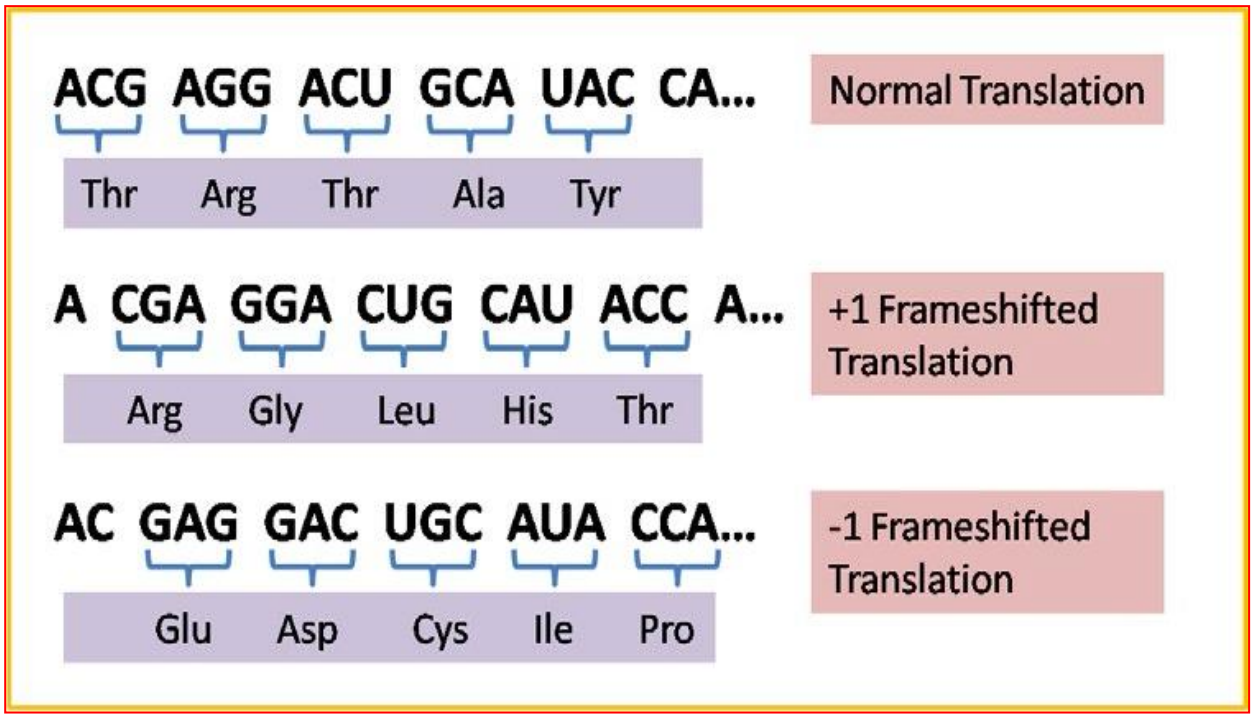


Fig 3.1: Schematic representation of normal translation process along with frame-shift events with +1 and -1 versions [3].

3.1.3 What causes frame-shift errors?

One clear implication of the suppressor analysis is that frame-shifting is strongly stimulated by near-cognate decoding, that is decoding by an isoacceptor that makes a less than optimal wobble interaction with the mRNA. The example of suppression by a structurally normal near cognate tRNA in the*sufB2* strain of *S. typhimurium* clearly shows that near-cognate decoding can stimulate frame errors. Moreover, overproduction of same near-cognate tRNA induces frame-shifting at the same sites suppressed by *sufB2*. Some programmed frame-shifts are also stimulated by near-cognate decoding. The first example comes from the *dnaX* gene of *E.coli*, which encodes alternative forms of a subunit of DNA polymerase III [12]. Frame-shifting results in the expression of a C-terminally truncated form of the protein and occurs on a slippery heptameric sequence A-AAA-AAG, two tRNAs simultaneously slipping −1 from AAA-AAG to AAA-AAA. The unusually high efficiency of this site partly results from the near-cognate recognition of the AAG codon by a tRNA with a modified U in the wobble position which restricts the ability of tRNA to decode AAG. Expressing a tRNA that recognizes AAG in a completely cognate fashion reduced frame-shifting on the site. The weakness of the interaction apparently predisposes the ribosome to frame-shift [17].

3.2 How to resolve the problem?

3.2.1 The MARKOV package: Markovian models

Markovian models are the simplest, easiest to use statistical models available for genomic sequences. Statistical properties associated with a Markovian model make it become a valuable tool to the one who wants to take into account the occurrences of *k*-mers in a sequence. Their most commonly used version, the so-called *classical* Markovian models, could be automatically built from a set of real genomic sequences. Apart from genomics, such models appear in various scientific fields' including-but-not-limited-to speech recognition, population processes, and queuing theory and search engines [22].

3.2.1.1 Some theoretical aspects

3.2.1.1.1 Main definition

Formally, a classical Markovian model applied to a set of random variables $V_{1,}\ldots,V_n$ causes the probabilities associated with the potential values for V_n to depend on the values of $V_1,..V_{n-1}$. We will focus on homogenous Markovian models, where the probabilities for the different values for V_n are conditioned by the values already chosen for a small subset $[V_{i-k-1}, V_{i-1}]$ of variables *from the past*, also called *context* of V_i. The parameter k is called the *order* of the Markovian model. Moreover, the probabilities of the values for V_i in a homogenous Markovian model cannot in any way be conditioned by the index *i* of the variable. Applied to genomic sequences, the random variable V_i stands for the i^{th} base in the sequence. The Markovian model constrains the occurrence probability for a base α in a given context composed of the *k* previously assigned letters, therefore weakly constraining the proportions of each *k+1*-mers [22].

3.2.1.1.2 Hidden Markov Models (HMMs)

Hidden Markovian models address the hierarchical decomposability of most sequences. A hidden Markovian model is a combination of a top-level Markovian model and a set of bottom-level Markovian models, known as hidden states. The generation process associated with an HMM initiates the sequences using a random hidden state. At each step of the generation, the algorithm may switch to other hidden using probabilities from the top-level model, and then emits a symbol using probabilities related to the current urn.Once again, this class of models' expressivity seems to exceed that of the classical Markovian models. However, in our context, it is possible to emulate a hidden model with a classical one just by duplicating the alphabet so that the emitted character also contains the state which it belongs to [5, 22].

3.2.1.2 Implementing a Markovian model

This section describes the syntax and semantics of Markovian description files, as shown in figure:

3.2.1.2.1 Main Structure:

```
TYPE = MARKOV
ORDER = ...
[PHASE = ...]
[SYMBOLS = ...]
[START = ...]
[FREQUENCIES | HMMFREQUENCIES] = ...
[ALIASES = ...]
```

Fig 3.2: Main Structure of a Markovian description files [22].

Clauses nested inside square brackets are mainly optional. The given order for the clauses is mandatory.

3. 2.1.2.2 Markovian generation specific clauses

A Markovian description file allows definition of the Markovian model parameters [22].

3.2.1.2.3 The ORDER clause

ORDER= *k*

$k \in \mathrm{N}$

Required

Sets the *order* of the underlying Markovian model to a positive integer value *k*. The order of a Markovian model is the number of previously emitted symbols taken into account for the emission probabilities of the next possible symbol [22].

3.2.1.2.4 The SYMBOLS clause

SYMBOLS = {WORDS, LETTERS}

Optional, defaults to SYMBOLS = WORDS

Choose the type of symbols to be used for random generation. When WORDS is selected, each pair of symbols must be separated by at least a blank character (space, tabulation or newline). A Markovian

description file written using WORDS will then be easier to read, as explicit names for symbols could be used, but may take a little longer to write [22].

3.2.1.2.5 The FREQUENCIES clause

This clause is used to define the probabilities of emission of the Markovian model [22].

FREQUENCIES= *s1 n1 s2 n2 . . .*

n_i € N

Required

Defines the probabilities of emission for the different symbols. Eachs_i is either a sequence of symbols separated by white spaces or a word, depending on the value of the SYMBOLS parameter. Each s_i is composed of *k+1* symbols, *k*being the order. The first *k* symbols define the context and the last letter a candidate to emission. The relationship between the frequency definition s_i n_i and the probability$p_{c_i w_i}$ of emitting c_i in a context w_i, $s_i=w_i.c_i$ is given by the following formula:

$$p_{c_i w_i} = \frac{n_i}{\sum_{j=w_i.c}^{k} n_j}$$

3.2.1.3 HMM: Forward Algorithm:

The **forward algorithm**, in the context of a hidden Markov model, is used to calculate a 'belief state': the probability of a state at a certain time, given the history of evidence. The process is also called as *filtering*. The forward algorithm is closely related to, but distinct from, the Viterbi algorithm [5, 2, and 25]. For an HMM such as this one:

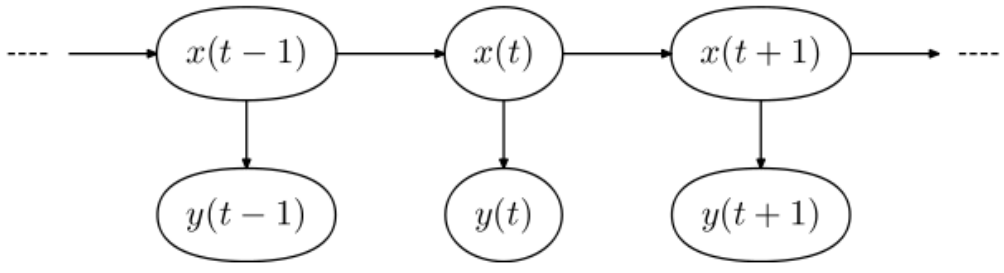


Fig 3.3: Forward Algorithm

This probability is written as $P(x_t|y_{1:t})$. Here *x(t)* is the hidden state which is abbreviated as x_t and $y_{1:t}$ are the observations to *t*. A belief state could be calculated at each time step, but doing this does not, in a strict sense, produce the most likely state *sequence*, but rather the most likely state at each time step, given the previous history.

3.2.1.3.1 Forward HMM: Algorithm:

The goal of the forward algorithm is to compute the joint probability $p(x_t, y_{1:t})$, where for notational convenience we have abbreviated *x(t)* as x_t and *(y(1),y(2),...y(t))* as $y_{1:t}$. Computing $p(x_t, y_{1:t})$ directly would require marginalizing over all possible state sequences $\{x_{1:t-1}\}$, the number of which grows exponentially with *t*. Instead, the forward algorithm takes advantage of the conditional independence rules of the hidden Markov model (HMM) to perform the calculation recursively [5, 22, and 25].

To demonstrate the recursion, let

$$\alpha_t(x_t) = p(x_t, y_{1:t}) = \sum_{x_{t-1}} p(x_t, x_{t-1}, y_{1:t})$$

Using the chain rule to expand $p(x_t, x_{t-1}, y_{1:t})$, we can then write

$$\alpha_t(x_t) = \sum_{x_{t-1}} p(y_t|x_t, x_{t-1}, y_{1:t-1}) p(x_t|x_{t-1}, y_{1:t-1}) p(x_t, x_{t-1}, y_{1:t-1})$$

Because y_t is conditionally independent of everything but x_t, and x_t is conditionally independent of everything but x_{t-1}, this simplifies to

$$\alpha_t(x_t) = p(y_t|x_t) \sum_{x_{t-1}} p(x_t, x_{t-1}) \alpha_{t-1}(x_{t-1})$$

Thus, since $p(y_t|x_t)$ and $p(x_t|x_{t-1})$ are given by the model's emission distributions and transition probabilities, one can quickly calculate $\alpha_t(x_t)$ from $\alpha_{t-1}(x_{t-1})$ and avoid incurring exponential computation time [5,24].

$$Transition\ probability = p(x_t|x_{t-1}) = p(y|x) = \frac{p(xy)}{p(x)} \approx \frac{freq(xy)}{freq(x)}$$

3.3 Diseases due to Frame-shift events:

Several diseases have frame-shift mutations as at least part of the cause. Knowing prevalent mutations could also aid in the diagnosis of the disease. Currently, there are attempts to use frame-shift mutations beneficially in the treatment of diseases, changing the reading frame of the amino acid [20, 25].

3.3.1 Types

3.3.1.1 Cancer

Frame-shift mutations are known to be a factor in colorectal cancer as well as other cancers with microsatellite instability. As stated previously, frame-shift mutations are more likely to occur in a region of repeat sequence. When DNA mismatch repair does not fix the addition or deletion of bases, these mutations are more likely to be pathogenic. This may be in part because the tumor is not told to stop growing [15, 25].

3.3.1.2 Crohn's Disease

Crohn's Disease has an association with the NOD2 gene. A frame-shift mutation within the coding region of the gene can be a factor in Crohn's Disease. The mutation is an insertion of a Cytosine at position 3020. This leads to a premature stop codon, shortening of the protein that is supposed to be transcribed [25].

3.3.1.3 Tay-Sachs Disease

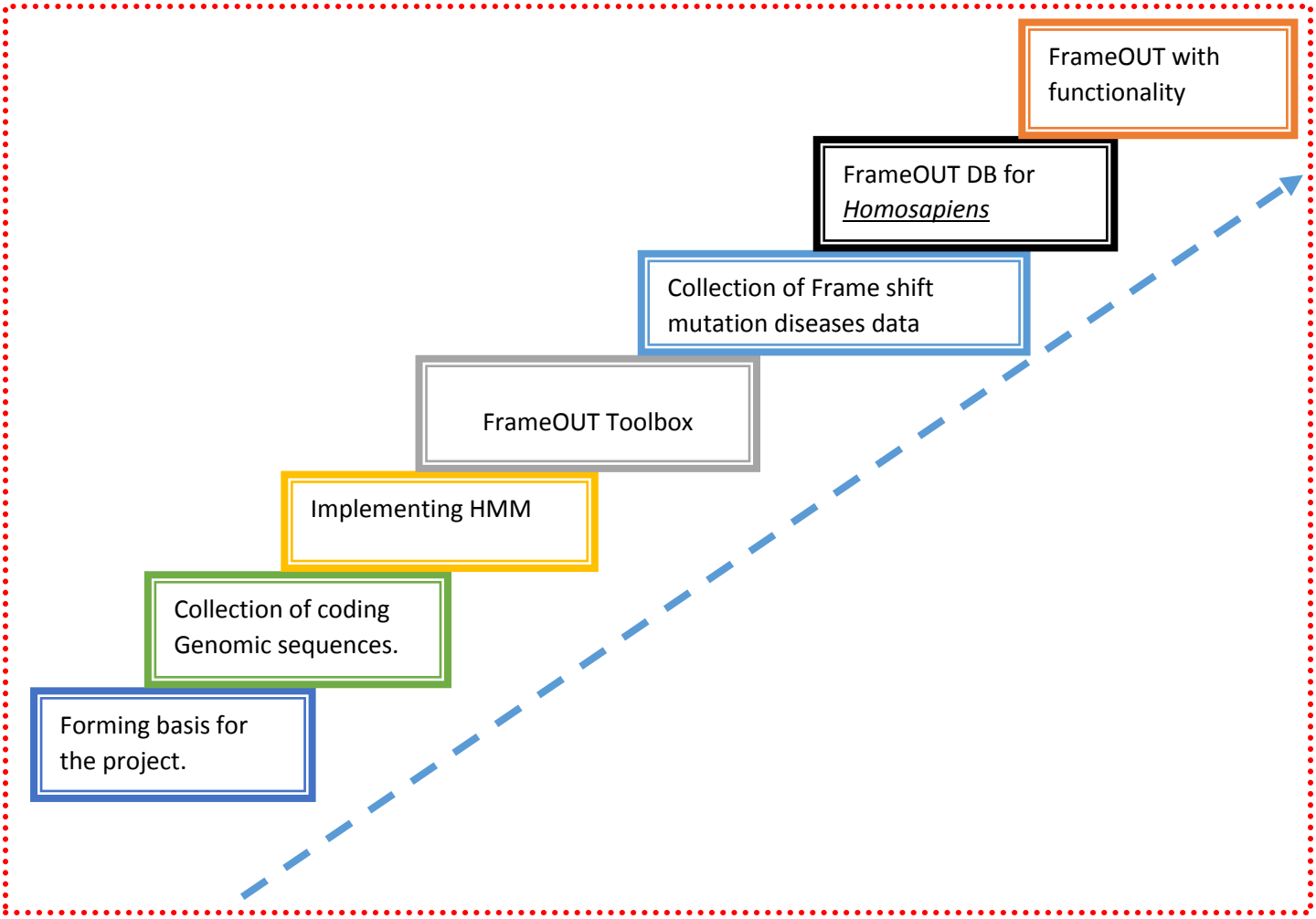
Tay - Sachs disease is a fatal disease affecting the central nervous system. It is most frequently found in infants and small children. Disease progression begins in the womb but later symptoms do not appear until approximately 6 months of age. There is no cure for the disease [25].

3.3.1.4 Other

There are a lot many other harmful diseases caused due to frame-shift mutation e.g. HIV, Smith-Magenis Syndrome, Hypertrophic Cardiomyopathy, Muscular Dystrophy, and a lot more. It is believed that this tool and databse combo will help the scientific community in the analysis of frameshift events and their disease specific consequences [11, 13, 14, and 25].

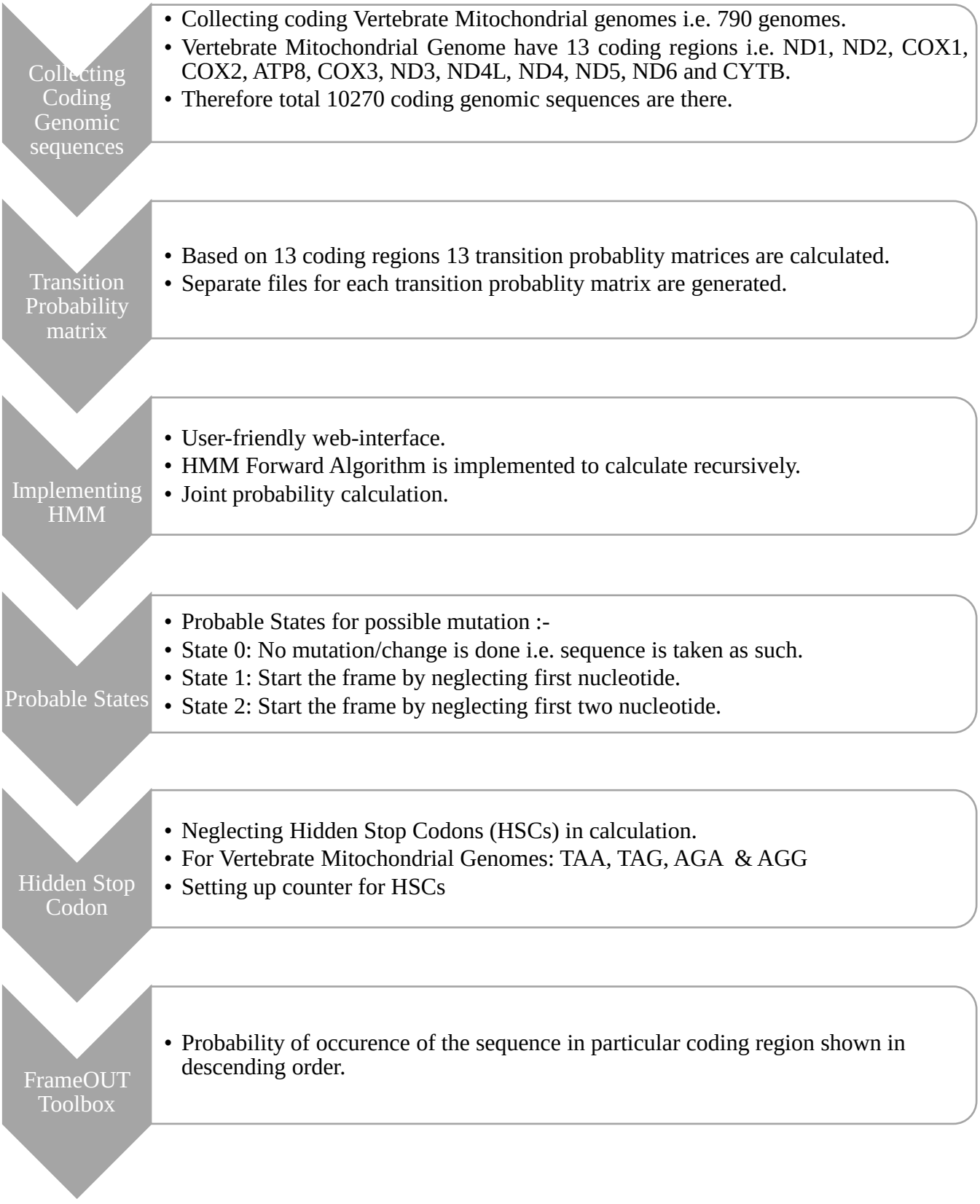
CHAPTER 4: METHODOLOGY

Overall methodology being applied in the project is given below:

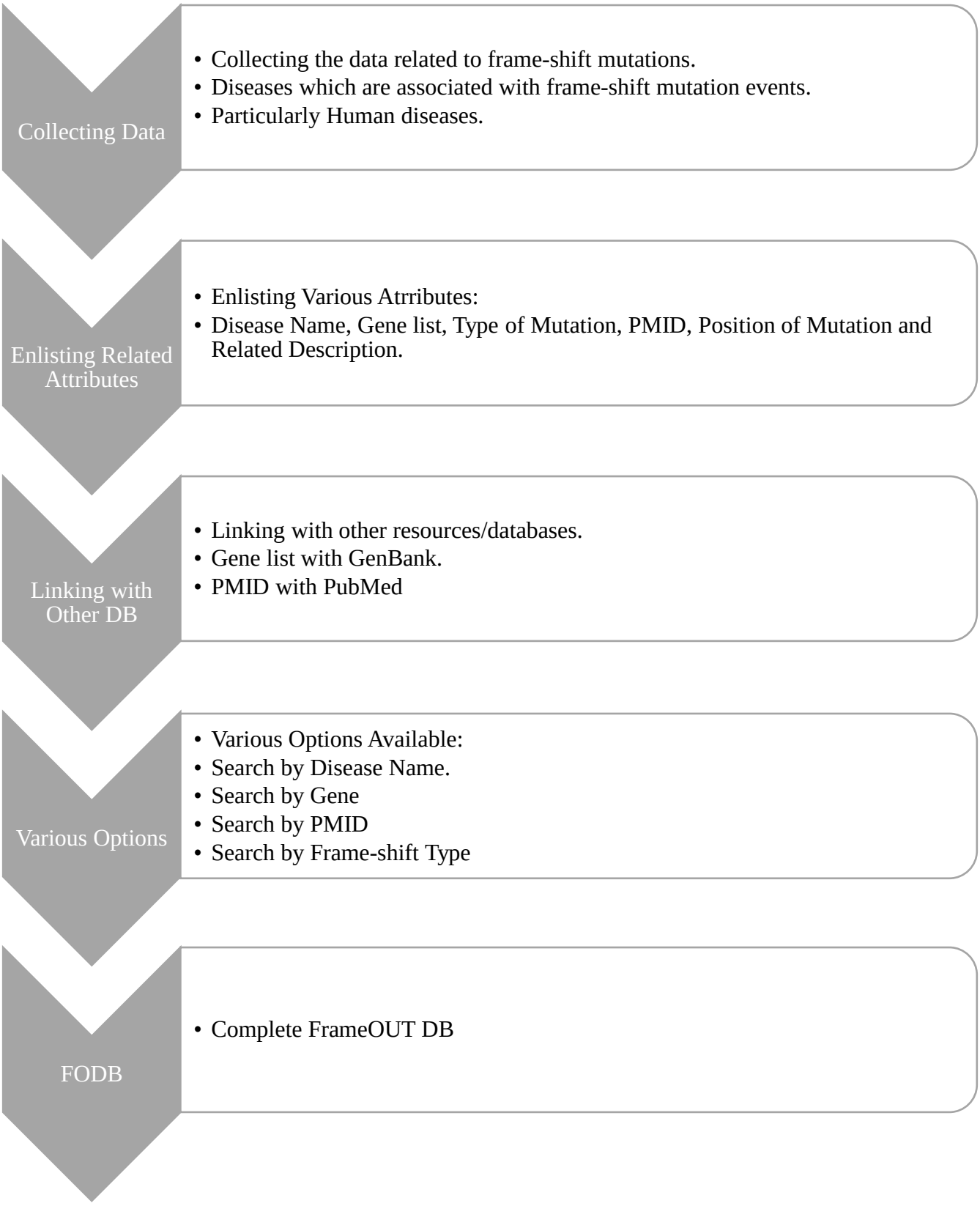


Specific methodology being followed for FrameOUT tool and FrameOUT DB is described in sections 4.1 and 4.2 respectively.

4.1 Methodology for FrameOUT Tool



4.2 Methodology for FrameOUT DB



CHAPTER 5: IMPLEMENTATION AND ANALYSIS

5.1 FrameOUT TOOL

5.1.1 Basic Flowchart

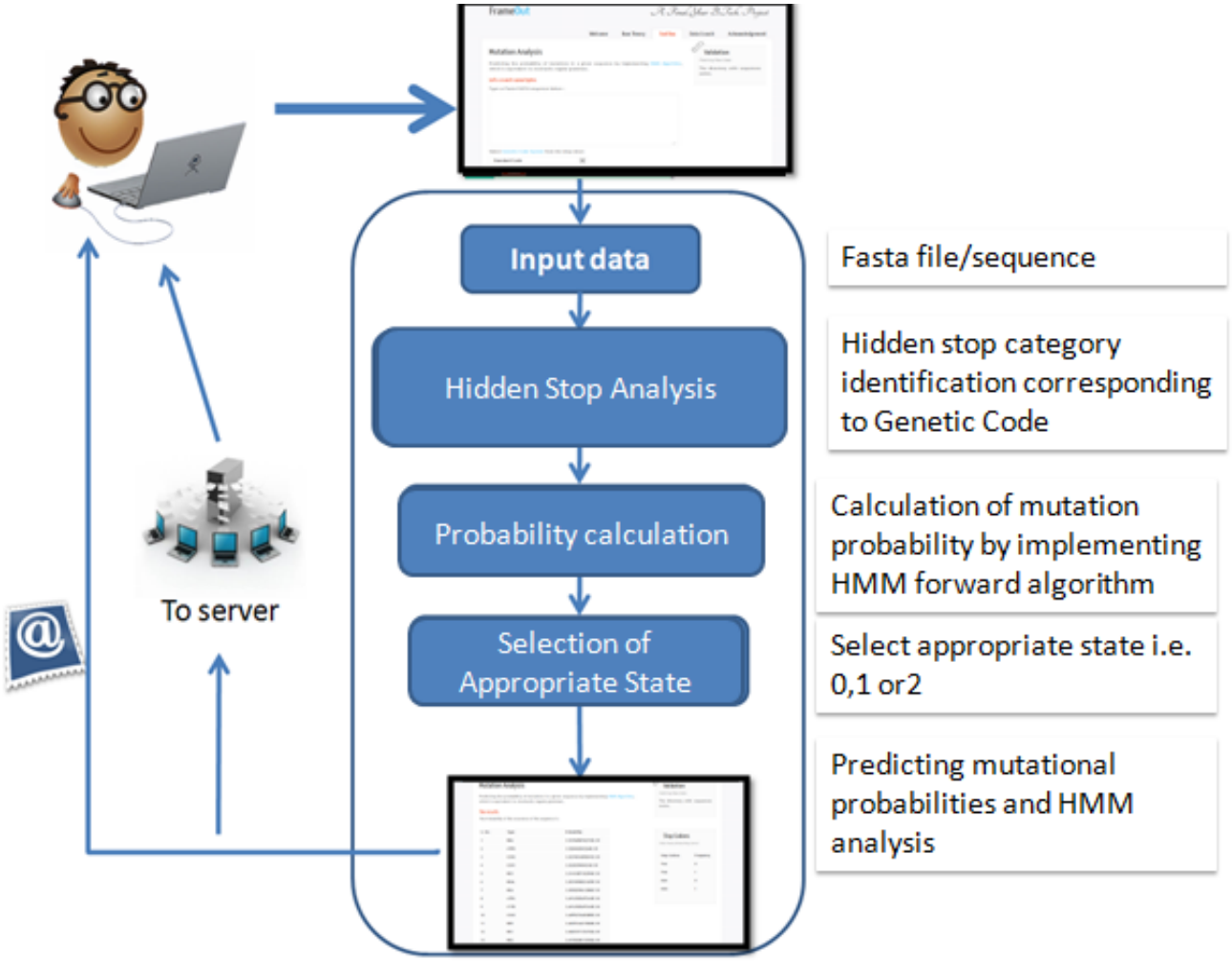


Fig 5.1: *Flow Chart of FrameOUT Tool*

5.1.2 Data Collection

Collecting coding Genomic sequences particularly, Vertebrate mitochondrial genomes; I collected genomic sequence data for 790 genomes. These collected genomes were having 13 protein coding regions, namely: ND1, ND2, COX1, COX2, ATP8, COX3, ND3, ND4L, ND4, ND5, ND6 and CYTB. After collecting the genomic sequence data I segregated these 790 genomes into respective coding sequence files. Therefore separate files listing respective coding regions have corresponding sequences and thereby there are 13 files, each having different 790 sequences. So, total there are 10270 coding genomic sequences.

5.1.3 Transition Probability Calculation

For each separate file transition probability matrix is calculated, therefore 13 different transition probability matrices are generated based on the following formula [5, 22]:

$$Transition\ probability = p(x_t|x_{t-1}) = p(y|x) = \frac{p(xy)}{p(x)} \approx \frac{freq(xy)}{freq(x)}$$

$$= \frac{Frequency\ of\ particular\ codon}{Frequency\ of\ all}$$

For example: -= $\frac{Frequency\ (AAA)}{Frequency(AAA+AAT+AAG+AAC)}$

Table 5.1: Transition Probability Matrix for ND1

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT	Total
A	0.057822	0.066803	0.028136	0.07472	0.079752	0.07522	0.031846	0.118937	0.10145771	0.101073	0.055142	0.071593	0.077195	0.076719	0.050524	0.0898291	1
C	0.091784	0.08985	0.066661	0.077978	0.05887	0.084142	0.030344	0.077563	0.06441337	0.149361	0.067974	0.044816	0.081532	0.087119	0.036541	0.08343339	1
G	0.032106	0.031763	0.035119	0.048846	0.041684	0.032959	0.018576	0.036603	0.05823931	0.018147	0.033395	0.022773	0.058943	0.030298	0.025083	0.03742348	1
T	0.083519	0.081855	0.031905	0.101133	0.078972	0.131991	0.020379	0.082161	0.05379123	0.078556	0.03234	0.046927	0.105164	0.075381	0.023343	0.06147234	1

Table 5.2: Transition Probability Matrix for ND2

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT	Total
A	0.057835	0.066839	0.028115	0.074714	0.079789	0.075221	0.031803	0.118962	0.10143341	0.101147	0.055156	0.071552	0.077217	0.076742	0.050526	0.08982282	1
C	0.091809	0.089856	0.066667	0.077977	0.058883	0.08415	0.030343	0.077582	0.06438979	0.149408	0.067981	0.044798	0.08153	0.087157	0.036543	0.08344647	1
G	0.032099	0.031722	0.035112	0.048826	0.041681	0.032958	0.018568	0.036559	0.05822408	0.018142	0.033374	0.022818	0.058936	0.030294	0.025064	0.03742005	1
T	0.083521	0.081853	0.031879	0.101176	0.078981	0.132009	0.020371	0.082141	0.05381436	0.078566	0.032328	0.046869	0.105193	0.07536	0.023324	0.06142574	1

Table 5.3: Transition Probability Matrix for COX1

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT	Total
A	0.057797	0.066888	0.028116	0.074723	0.07976	0.075199	0.03181	0.118964	0.1013389	0.101033	0.055241	0.071497	0.077226	0.076751	0.050548	0.08977329	1
C	0.091797	0.089855	0.066672	0.077964	0.058914	0.084142	0.030356	0.077508	0.06443286	0.149299	0.068009	0.044807	0.081575	0.087124	0.036526	0.08342598	1
G	0.032088	0.03171	0.035094	0.048819	0.041639	0.032983	0.018631	0.036614	0.05828514	0.018197	0.033379	0.022818	0.058904	0.030316	0.02513	0.03750104	1
T	0.08352	0.081941	0.031858	0.101158	0.079023	0.131956	0.020362	0.082139	0.05384183	0.078561	0.032374	0.046886	0.105088	0.075267	0.023381	0.06146507	1

Table 5.4: Transition Probability Matrix for COX2

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT	Total
A	0.057796	0.066933	0.028162	0.074707	0.079748	0.075265	0.031835	0.118859	0.10150269	0.101	0.055289	0.071577	0.077209	0.076631	0.050566	0.08980006	1
C	0.091795	0.089808	0.066629	0.077919	0.059047	0.084038	0.030361	0.077589	0.064512	0.149145	0.067882	0.044795	0.081476	0.087125	0.036549	0.08345051	1
G	0.032096	0.031748	0.035097	0.048825	0.041655	0.032948	0.018612	0.036698	0.05822535	0.01815	0.033355	0.022841	0.058915	0.030376	0.025163	0.03744078	1
T	0.08354	0.081936	0.031831	0.101177	0.07894	0.131866	0.020419	0.082118	0.05382076	0.078583	0.032419	0.046903	0.105159	0.075347	0.023365	0.06142796	1

Table 5.5: Transition Probability Matrix for ATP8

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT	Total
A	0.057806	0.066889	0.028186	0.07473	0.079689	0.075225	0.031857	0.118965	0.10154878	0.100986	0.055214	0.071589	0.077246	0.076574	0.050561	0.08980691	1
C	0.091766	0.089783	0.066658	0.077936	0.058982	0.084125	0.030377	0.07757	0.06452599	0.149157	0.067929	0.044747	0.081553	0.087222	0.036539	0.08338197	1
G	0.032109	0.031741	0.035025	0.048861	0.041639	0.032999	0.018635	0.036649	0.05814343	0.018191	0.033333	0.022847	0.058952	0.030315	0.025155	0.03746083	1
T	0.083542	0.081986	0.031854	0.101129	0.078923	0.131901	0.020359	0.082104	0.05385222	0.078672	0.032376	0.046888	0.105153	0.075299	0.02339	0.06139004	1

Table 5.6: Transition Probability Matrix for ATP6

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT	Total
A	0.057776	0.066921	0.028171	0.074759	0.079814	0.075291	0.031858	0.118893	0.10153181	0.101048	0.055291	0.071606	0.077226	0.076505	0.050531	0.08976434	1
C	0.091801	0.089773	0.066631	0.077971	0.058962	0.084045	0.030353	0.077515	0.0645295	0.149125	0.067846	0.04475	0.081459	0.087209	0.036536	0.08351483	1
G	0.032085	0.031717	0.035027	0.048852	0.041661	0.033033	0.018609	0.036578	0.05818315	0.018082	0.03338	0.022849	0.058924	0.030335	0.02516	0.03751668	1
T	0.083614	0.08189	0.031874	0.101137	0.078925	0.131804	0.020395	0.082263	0.05387986	0.078594	0.032363	0.046941	0.105128	0.075367	0.023379	0.06144533	1

Table 5.7: Transition Probability Matrix for COX3

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT	Total
A	0.05773	0.06694	0.028165	0.074696	0.07987	0.075299	0.031835	0.118989	0.10148561	0.101091	0.05516	0.071546	0.077261	0.076559	0.050573	0.08978033	1
C	0.091824	0.0898	0.066601	0.077957	0.058982	0.084032	0.03034	0.077481	0.06447855	0.149124	0.068072	0.044786	0.08154	0.087223	0.036547	0.08342494	1
G	0.032104	0.031711	0.035052	0.048898	0.041673	0.033025	0.018579	0.036585	0.058161	0.018045	0.033384	0.022832	0.058926	0.0303	0.025191	0.03751889	1
T	0.083635	0.081887	0.031883	0.101117	0.078948	0.131831	0.020375	0.082155	0.05386704	0.078713	0.032318	0.046937	0.105097	0.075278	0.02338	0.06139946	1

Table 5.8: Transition Probability Matrix for ND3

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT	Total
A	0.057717	0.066975	0.02819	0.074782	0.079843	0.075269	0.031822	0.118941	0.10148865	0.101024	0.055209	0.071558	0.077192	0.076573	0.050582	0.08975983	1
C	0.091812	0.089855	0.066577	0.078007	0.058998	0.084013	0.030353	0.077476	0.06451454	0.148944	0.067982	0.044837	0.081582	0.087191	0.036526	0.08340946	1
G	0.032094	0.031735	0.035024	0.048868	0.04164	0.03297	0.018605	0.03666	0.05810358	0.018146	0.033408	0.022868	0.058946	0.030288	0.025195	0.0375178	1
T	0.083577	0.081871	0.031872	0.101045	0.078982	0.131842	0.020375	0.082211	0.05395474	0.07864	0.032371	0.046951	0.105064	0.075325	0.023427	0.06142134	1

Table 5.9: Transition Probability Matrix for ND4L

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT	Total
A	0.057718	0.067002	0.028173	0.074712	0.079844	0.07534	0.031842	0.118946	0.10150658	0.100933	0.055291	0.071563	0.07721	0.076576	0.05064	0.08969216	1
C	0.091736	0.089797	0.066579	0.078033	0.059054	0.084032	0.030339	0.07745	0.06467239	0.149057	0.067925	0.044753	0.081556	0.087126	0.036506	0.08349802	1
G	0.032089	0.03174	0.035021	0.048916	0.041627	0.032949	0.018632	0.036723	0.05816758	0.018061	0.033404	0.022806	0.058923	0.030376	0.02517	0.03746964	1
T	0.083691	0.081881	0.031848	0.101064	0.078996	0.13176	0.020385	0.082081	0.05392661	0.078532	0.032336	0.047066	0.105025	0.075337	0.023449	0.06144491	1

Table 5.10: Transition Probability Matrix for ND4

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT	Total
A	0.057677	0.06702	0.028197	0.074775	0.079859	0.075287	0.031823	0.119005	0.10146675	0.100893	0.055329	0.071568	0.077296	0.076559	0.050568	0.08971941	1
C	0.091785	0.089739	0.066543	0.077991	0.059055	0.08406	0.03033	0.077472	0.06451135	0.148909	0.068	0.044882	0.081604	0.08714	0.036528	0.08341684	1
G	0.032072	0.031737	0.035003	0.048821	0.041613	0.032934	0.01861	0.036716	0.05822527	0.018166	0.033377	0.022762	0.05887	0.03028	0.0252	0.03755993	1
T	0.083701	0.081955	0.031821	0.101162	0.078965	0.13171	0.02038	0.082181	0.05402467	0.078744	0.03227	0.046869	0.105007	0.075375	0.023427	0.06144875	1

Table 5.11: Transition Probability Matrix for ND5

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT	Total
A	0.05769	0.067036	0.028199	0.074785	0.079967	0.075409	0.031733	0.118934	0.10143253	0.100959	0.055317	0.071469	0.077301	0.076569	0.05059	0.08977995	1
C	0.091792	0.089705	0.06653	0.07809	0.05911	0.083975	0.030354	0.077483	0.06450849	0.148911	0.067974	0.044595	0.081467	0.087267	0.036555	0.08341022	1
G	0.032059	0.031657	0.035001	0.04887	0.041611	0.032946	0.018623	0.036686	0.05817019	0.018097	0.033419	0.022905	0.058947	0.03019	0.025238	0.03754613	1
T	0.083772	0.081872	0.031824	0.101118	0.078998	0.131729	0.020283	0.082159	0.05403351	0.078794	0.032412	0.047004	0.105037	0.075274	0.023398	0.0614302	1

Table 5.12: Transition Probability Matrix for ND6

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT	Total
A	0.057681	0.06702	0.028165	0.074742	0.079894	0.075432	0.031809	0.118938	0.10132167	0.100749	0.055275	0.071394	0.077311	0.076531	0.050628	0.08968101	1
C	0.0917	0.089674	0.066582	0.078074	0.059165	0.083928	0.030331	0.077456	0.06464254	0.148977	0.068058	0.044694	0.081377	0.087211	0.036603	0.08343482	1
G	0.032048	0.031718	0.035056	0.048853	0.041623	0.032859	0.018616	0.036794	0.05809833	0.018192	0.033511	0.022979	0.058966	0.030264	0.025309	0.03759993	1
T	0.083845	0.081898	0.031817	0.101127	0.078906	0.131768	0.020305	0.082177	0.05398229	0.078728	0.032474	0.046925	0.104982	0.075298	0.023453	0.06135216	1

Table 5.13: Transition Probability Matrix for CYTB

	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT	Total
A	0.057718	0.066974	0.028161	0.07478	0.079836	0.075431	0.031805	0.118879	0.10134302	0.1008	0.05523	0.071532	0.077296	0.076437	0.050492	0.08963517	1
C	0.091813	0.089728	0.066557	0.07798	0.058976	0.08401	0.03038	0.077591	0.06444705	0.148857	0.068078	0.04481	0.081438	0.087235	0.036513	0.0834949	1
G	0.032044	0.031656	0.035068	0.048949	0.041673	0.032892	0.01855	0.036746	0.05805267	0.018137	0.033551	0.022795	0.058895	0.030343	0.025391	0.03759749	1
T	0.083761	0.081868	0.031838	0.101103	0.07897	0.131748	0.020324	0.082189	0.05390817	0.078746	0.032584	0.047129	0.10492	0.075411	0.023506	0.06139778	1

5.1.4 Implementing HMM: Forward Algorithm

Implementing Hidden Markov Model (HMM), forward algorithm which basically computes the joint probability$p\ (x_t\ ,\ y_{1:t})$, neglecting the stop codons. The forward algorithm takes advantage of the conditional independence rules of HMM to perform the calculation recursively based on following formula [5, 22]:

$$\alpha_t(x_t) = p(y_t|x_t)\sum_{x_{t-1}} p(x_t, x_{t-1})\alpha_{t-1}(x_{t-1})$$

Thus, since

- SYMBOLS = A, T, G, C
- STATE = 0, 1, 2

$p(y_t|x_t)$= *emission distributions probability*= 0.25 i.e. equal probability for each SYMBOL

$p(x_t|x_{t-1})$ = *transition probabilities* = value from transition probability matrix calculated previously

$\alpha_{t-1}(x_{t-1})$ = *previously calculated probability*

The various probable STATES are 0, 1 and 2.

- State 0: no mutation/change is done i.e. sequence is taken as such,
- State 1: start the frame by neglecting first nucleotide and
- State 2: start the frame by neglecting initial two nucleotides.

Fig 5.2: *FrameOUT Tool Box*

5.2 FrameOUT DB

FODB is a collection of all available frame-shift events and their association with various diseases specifically human diseases such as Corhn's disease, Rett-Syndrome, and Sandhoff disease, etc. This database has various options which may aid to give better results. Also, certain attributes are linked with other databases/resources such as gene list is linked with GenBank and PMID with PubMed.

Fig 5.3: *FrameOUT DB*

CHAPTER 6: FrameOUT CODE

6.1 FrameOUT Tool Code

 # PHP Database Connection

```
<?php
$user_name = "project";
$password = "hello123";
$database = "project";
$server = "localhost";
?>
<!DOCTYPE HTML>
<html>
<head>
<title>FrameOut - A B.Tech. Project</title>
<meta name="description" content="" />
<meta name="keywords" content="" />
<meta http-equiv="content-type" content="text/html; charset=windows-1252" />
<link rel="stylesheet" type="text/css"
href="http://fonts.googleapis.com/css?family=Tangerine&amp;amp;v1" />
<link rel="stylesheet" type="text/css"
href="http://fonts.googleapis.com/css?family=Yanone+Kaffeesatz" />
<link rel="stylesheet" type="text/css" href="style/style.css" />
</head>
<body>
<div id="main">
<div id="header">
<div id="logo">
<h1>Frame<a href="#">Out</a></h1>
<div class="slogan">A Final Year B.Tech. Project</div>
</div>
<div id="menubar">
<ul id="menu">
<!-- put class="current" in the li tag for the selected page - to highlight which page you're on -->
<li><a href="index.php">Welcome</a></li>
<li><a href="rawtheory.php">Raw Theory</a></li>
<li class="current"><a href="toolbox.php">Tool Box</a></li>
<li><a href="datacrunch.php">Data Crunch</a></li>
<li><a href="ack.php">Acknowledgement</a></li>
</ul>
</div>
</div>
<div id="site_content">
<div id="sidebar_container">
<img class="paperclip" src="style/paperclip.png" alt="paperclip" />
```

```
<div class="sidebar">
<!-- insert your sidebar items here -->
<h3>Validation</h3>
<h5>Fetching Raw Data</h5>
<p style="text-align:justify;">
#segregating the genome to coding regions i.e. 13 coding regions
        <?php
                if(is_dir('./sequence')){
                echo 'The directory with sequences exists.';}
                        else if (!is_dir('./sequence')){
                                echo 'The directory with sequences was not found. In order to proceed, you must first generate the sequences by clicking <a href="./cds_new.php">this link</a>. '; }
        ?>
</p>
</div>
</div>
<div id="content">
<!-- insert the page content here -->
#predicting the mutational probability for sequence entered by user
<h1>Mutation Analysis</h1>
<p>FrameOut is a tool to predict the mutational events occurring in genomic sequences through frame-shift events. Data is being framed by implementing<a href="http://en.wikipedia.org/wiki/Forward_algorithm" target="_blank">HMM Algorithm</a>.</p>
<h2>The results</h2>
        The probability of the occurence of the sequence in descending order is - <br/><br/>
<?php
                $str_seq=strtoupper($_POST['typedseq']);
                        if (strpos($str_seq,'GENOME') !== false) {
                                $str_full=explode("\n",strtoupper($str_seq));
                                $str_seq=$str_full[1];
                                //echo $str_seq;
                                }
#validating correct sequence.
                        if (preg_match("/[^ACGT]/", $str_seq)) {
                echo "<b>There was an error in the string you entered. Kindly re-validate.</b>";
                        }
#implementing forward HMM
                                 $gene_code=$_POST['code_list'];
                                 $state=$_POST['state'];
                                 $e=0.25;
                                 $cds= array("", "cox1", "cox2", "cox3", "nd1", "nd2", "nd3", "nd4", "nd4l", "nd5", "nd6", "atp6", "atp8", "cytb");
                                    function formatScientific($someFloat{
                                        $power = ($someFloat % 10) - 1;
                                        echo ($someFloat / pow(10, $power)) . "e" . $power;
                                }
                                    if($gene_code=='Vertebrate Mitochondrial Code') {
```

```
$seqlength=strlen($str_seq);
```

#probability calculation based on probable states

```
if ($state=='0') {
$cnt=0;
$limit=(floor($seqlength/3)-1)*3;
//echo $limit;
}
else if ($state=='1') {
$cnt=1;
$limit=((floor(($seqlength-1)/3)-1)*3)+1;
//echo $limit;
}
else if ($state=='2') {
$cnt=2;
$limit=((floor(($seqlength-2)/3)-1)*3)+2;
//echo $limit;
}
mysql_connect($server, $user_name, $password);
$db_found = mysql_select_db($database);
$freq_final = array();
for ($j=1;$j<14;$j++){
$k=0;
```

#counting hidden stop codons

```
//Set Stop Codon Counter to Zero!
$count_taa = 0;
$count_tag = 0;
$count_aga = 0;
$count_agg = 0;
for($i=$cnt;$i<=$limit;0){
$new_str=substr($str_seq,$i,3);
if ($new_str=='TAA'){
$count_taa=$count_taa+1;
$i=$i+3;
}
else if ($new_str=='TAG'){
$count_tag=$count_tag+1;
$i=$i+3;
}
else if ($new_str=='AGA'){
$count_aga=$count_aga+1;
$i=$i+3;
}
else if ($new_str=='AGG'){
$count_agg=$count_agg+1;
$i=$i+3;
}
```

```
                        else {
                                $value = mysql_query("SELECT * FROM ".$cds[$j]." WHERE Seq = '".$new_str."'")or die(mysql_error());
                                $val_array = mysql_fetch_array($value);
                                if ($k==0)
                                $freq = $e*$val_array['Freq'];
                                else
                                $freq = $e*$preval*$val_array['Freq'];
                                $k=$k+1;
                                $preval=$freq;
                                $i=$i+3;
                                }
                        }
                        $freq_final[$j] = $freq;
                        //echo$cds[$j].' => '.number_format($freq_final[$j],15).'<br/>';
                                }
                        }
                         else {
                                echo "Error processing the selected Genetic Code System. Kindly select <b>Vertebrate Mitochondrial Code</b> in the drop-down. <a href=\"javascript:history.back()\">Go back!</a>";
                                 }
                                //print_r($freq_final);
                                $freq_sorted=array("COX1" => $freq_final[1], "COX2" => $freq_final[2], "COX3" => $freq_final[3], "ND1" => $freq_final[4], "ND2" => $freq_final[5], "ND3" => $freq_final[6], "ND4" => $freq_final[7], "ND4L" => $freq_final[8], "ND5" => $freq_final[9], "ND6" => $freq_final[10], "ATP6" => $freq_final[11], "ATP8" => $freq_final[12], "CYTB" => $freq_final[13]);
                                arsort($freq_sorted);

                                ?>
```

#displaying mutational probability

```
<table width="500px" style="float:left;">
<tbody>
        <tr>
<td width="20%">S. No.</td>
<td width="45%">Type</td>
<td width="35%">Probability</td>
</tr>
<?php
                $cnt=1;
                foreach($freq_sorted as $key => $val) {
                        echo "<tr><td>".$cnt."</td><td>".$key."</td><td>".$val."</td></tr>";
                        $cnt=$cnt+1;
                }
        ?>
</tbody>
```

```
</table>
</div>
```
#displaying STOP codons
```
<div class="sidebar">
<h3>Stop Codons</h3>
<h5>How many times they occur</h5>
<table>
<tbody>
<tr>
<td width="120px">Stop Codons</td>
<td width="30px">Frequency</td>
</tr>
<tr>
<td>TAA</td>
<td><?php echo $count_taa; ?></td>
</tr>
<tr>
<td>TAG</td>
<td><?php echo $count_tag; ?></td>
</tr>
<tr>
<td>AGA</td>
<td><?php echo $count_aga; ?></td>
</tr>
<tr>
<td>AGG</td>
<td><?php echo $count_agg; ?></td>
</tr>
</tbody>
</table>
</div>
</div>
<div id="footer">
<p>Copyright &copy; 2014 | <a href="http://validator.w3.org/check?uri=referer">HTML5</a> | <a href="http://jigsaw.w3.org/css-validator/check/referer">CSS</a> | No rights reserved.</p>
</div></div>
</body>
</html>
```

6.2 FrameOUT DB Code

6.2.1 Search Engine Code

```
<!DOCTYPE HTML>
<html>
<head>
<title>FrameOUT - Frameshift Mutation Analysis</title>
<meta name="description" content="" />
<meta name="keywords" content="" />
<meta http-equiv="content-type" content="text/html; charset=windows-1252" />
<link rel="stylesheet" type="text/css" href="http://fonts.googleapis.com/css?family=Tangerine&amp;amp;v1" />
<link rel="stylesheet" type="text/css" href="http://fonts.googleapis.com/css?family=Yanone+Kaffeesatz" />
<link rel="stylesheet" type="text/css" href="style/style.css" />
</head>
<body>
<div id="main">
<div id="header">
<div id="logo">
<h1>Frame<a href="#">OUT</a></h1>
<div class="slogan">Frameshift Mutation Analysis</div>
</div>
<div id="menubar">
<ul id="menu">
<!-- put class="current" in the li tag for the selected page - to highlight which page you're on -->
<li><a href="index.php">Welcome</a></li>
<li><a href="rawtheory.php">Raw Theory</a></li>
<li><a href="toolbox.php">Tool Box</a></li>
<li class="current"><a href="frameoutdb.php">FrameOUT DB</a></li>
<li><a href="hns.php">Help & Support</a></li>
<li><a href="contact.php">Contact Us</a></li>
```

```
</ul>
</div>
</div>
<div id="site_content">
<div id="sidebar_container">
<img class="paperclip" src="style/paperclip.png" alt="paperclip" />
<div class="sidebar">
<!-- insert your sidebar items here -->
<h3>Keep searchin'</h3>
<p style="text-align:justify;">Search from our vast database of frameshift mutations. What's more, you can search by disease name, PMID, type of frameshift and gene!</p>
</div>
</div>
<div id="content">
<!-- insert the page content here -->
<h1>Out on a hunt!</h1>
<form action="search_disease.php" method="post" name="searchometer">
<div class="form_settings">

#search by disease
<p>Enter the <i>disease name</i> you would like us to show results for :</p>
<p><input type="text" name="disease_name" id="disease_name" value="" /></p>
<p style="padding-top: 15px"><span> </span><input class="submit" type="submit" name="name" value="Search by disease" /></p>
</div>
</form>

#search by gene
<p style="text-align:center;">-OR-</p>
<form action="search_gene.php" method="post" name="searchometer">
<div class="form_settings">
```

Enter the gene you would like us to show results for :
<input type="text" name="gene_name" id="gene_name" value="" />
 <input class="submit" type="submit" name="name" value="Search by gene" />

</form>

#search by PMID

-OR-
<form action="search_pmid.php" method="post" name="searchometer">

Enter the PMID you would like us to show results for :
<input type="text" name="pmid_name" id="pmid_name" value="" />
 <input class="submit" type="submit" name="name" value="Search by PMID" />

</form>

#search by frame-shift type

-OR-
<form action="search_frameshift.php" method="post" name="searchometer">

Select the Type of Frameshift you would like us to show results for :
<select name="frameshift_select">
<option value="+1(insertion)">+1(insertion)</option>
<option value="+1(insertion) & -1(deletion)">+1(insertion) & -1(deletion)</option>
<option value="-1(deletion) & duplication">-1(deletion) & duplication</option>
<option value="-1(deletion)">-1(deletion)</option>
<option value="+1(insertion) & duplication">+1(insertion) & duplication</option>
<option value="-1(deletion), +1(insertion) or duplication">-1(deletion), +1(insertion) or duplication</option>

<option value="duplication">duplication</option>
<option value="indel">indel</option>
<option value="subsitution">substitution</option>
</select>

 <input class="submit" type="submit" name="name" value="Search by Frameshift Type" />

</form>

Copyright © 2014 | HTML5 | CSS | No rights reserved.

</body>
</html>

6.2.2 SQL Query Code

```
--
-- Table structure for table `sheet1`
--
CREATE TABLE IF NOT EXISTS `sheet1` (
  `disease` varchar(89) DEFAULT NULL,
  `gene` varchar(60) DEFAULT NULL,
  `frameshift` varchar(42) DEFAULT NULL,
  `position` varchar(147) DEFAULT NULL,
  `pmid` varchar(11) DEFAULT NULL,
  `description` varchar(598) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

--

-- Dumping data for table `sheet1`

--

INSERT INTO `sheet1` (`disease`, `gene`, `frameshift`, `position`, `pmid`, `description`) VALUES ('Crohn''s disease', 'NOD2', '+1(insertion)', '3020insC', '24137163', 'Mutation in NOD2 gene(in the pericentromeric region of chromosome 16)'),

and similarly for rest of the entries.

CHAPTER 7: RESULT WITH SAMPLE SESSION

7.1 Sample session with FrameOUT tool

User need to input *FASTA* sequence inside the tool box and need to select the respective state from the drop down menu from field "*Select State from List".* But Genetic Code System will be Vertebrate Mitochondrial Code by default. In the result mutational probabilities will be there in descending order and separate counter of Hidden Stop Codons.

7.1.1 State 0:

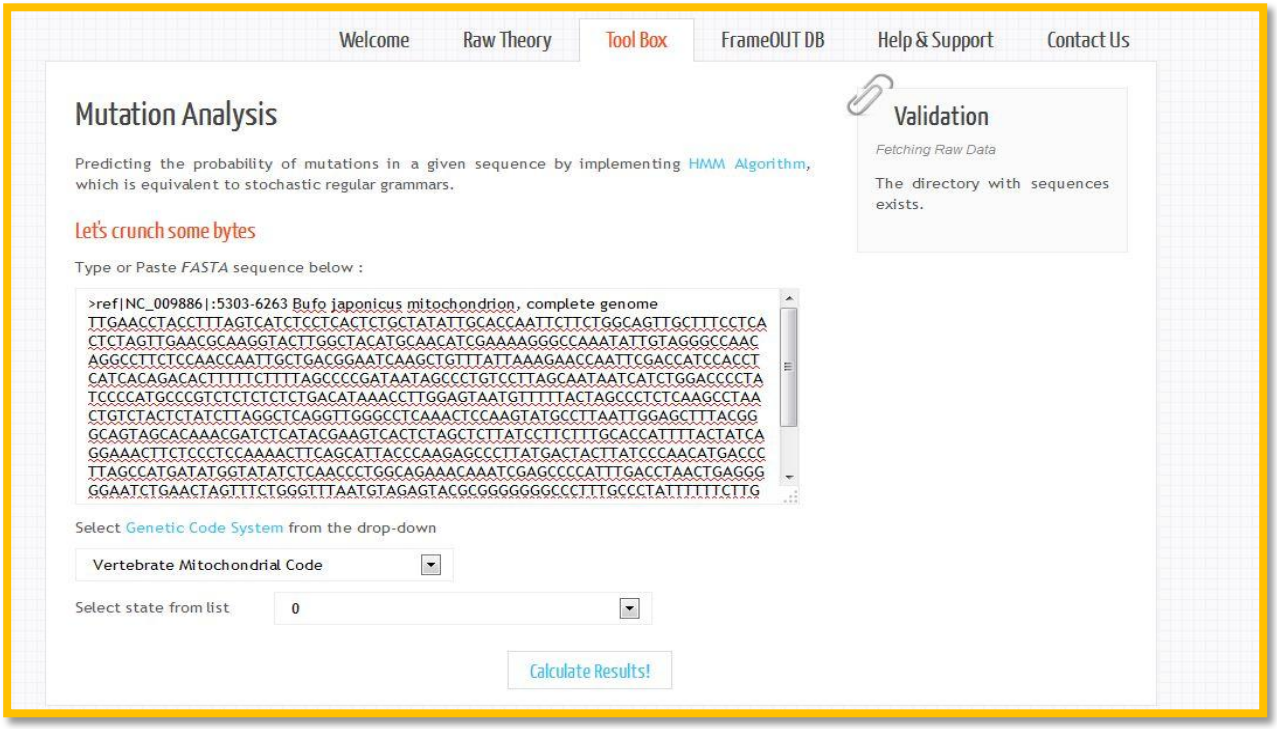


Fig 7.1: *State 0*

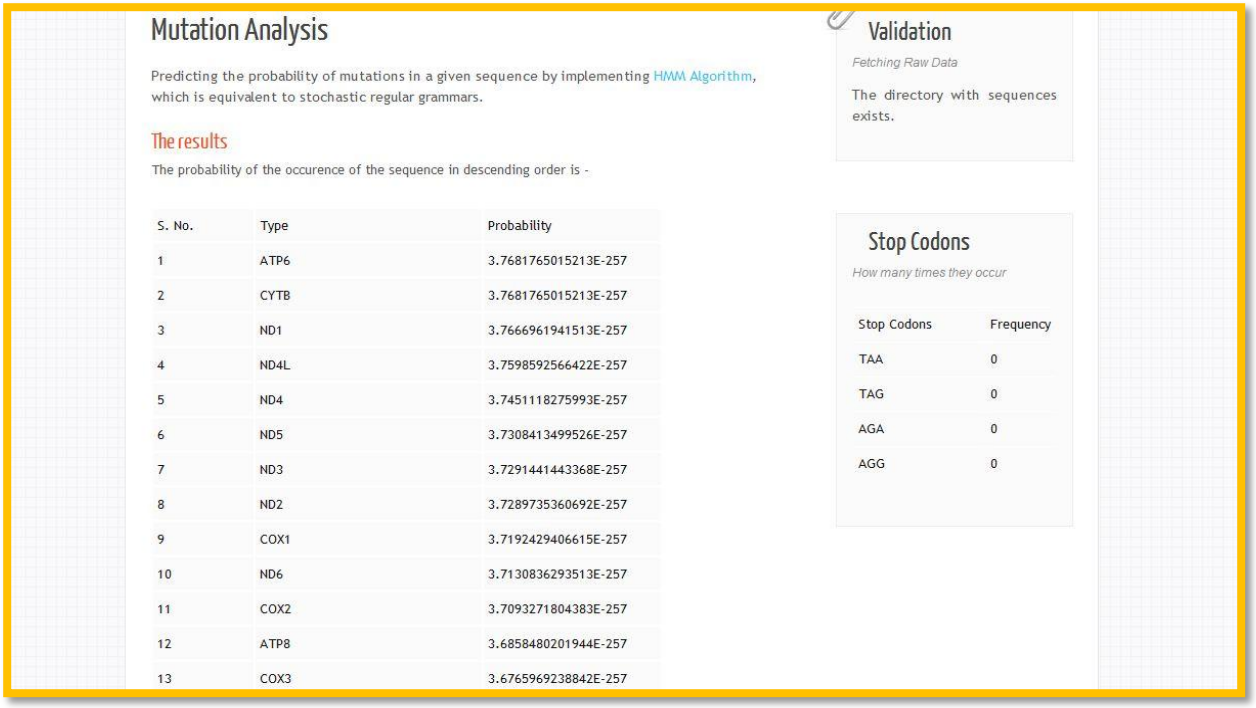


Fig 7.2: *State 0: Result*

7.1.2 State 1:

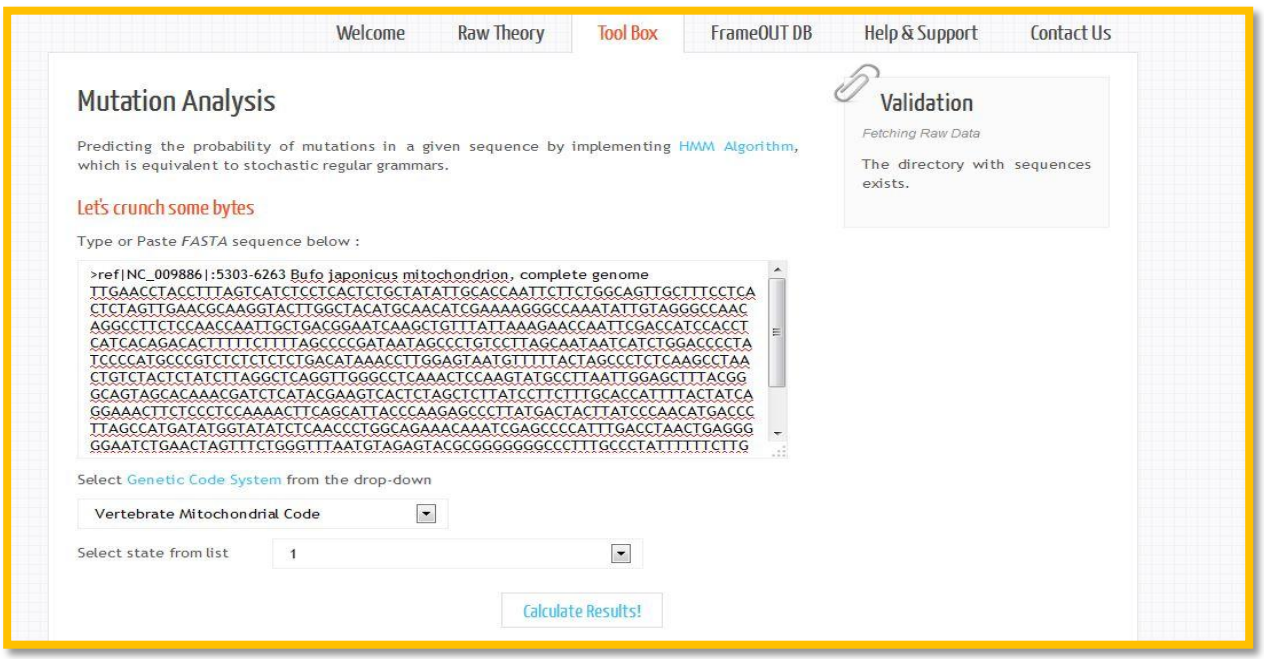


Fig 7.3: *State 1*

Result

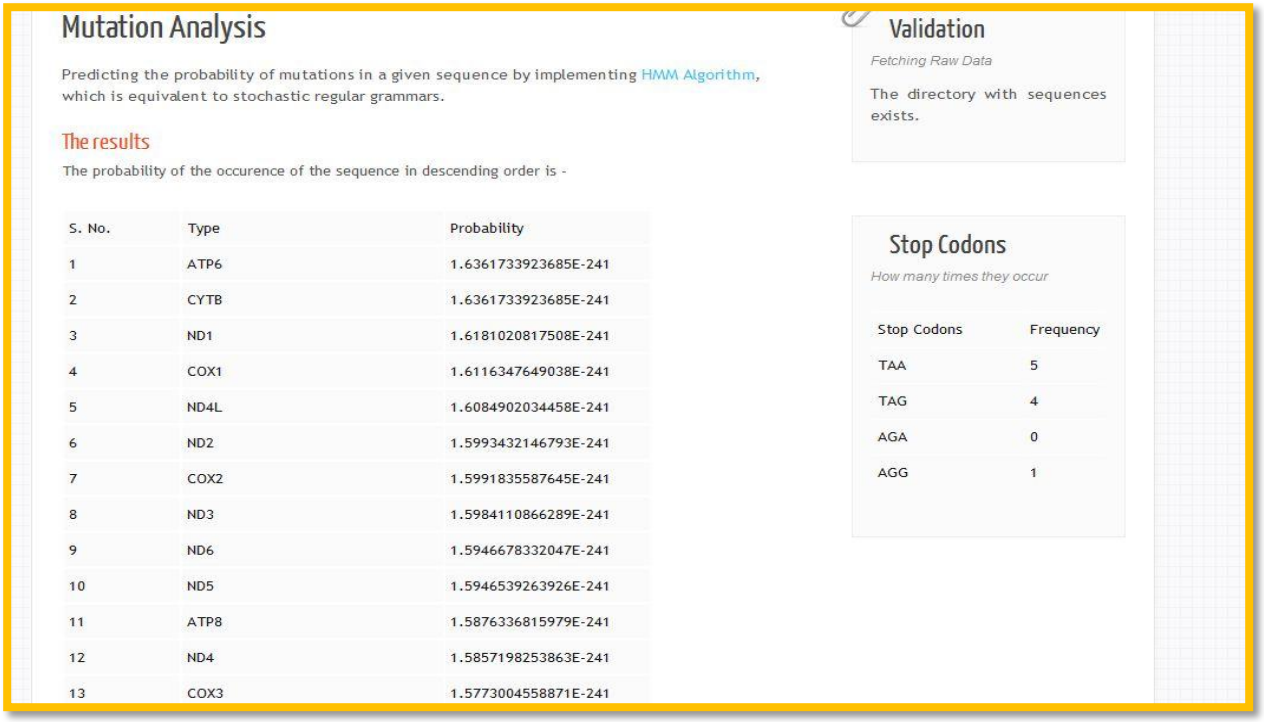


Fig 7.4: *State 1: Result*

7.1.3 State 2:

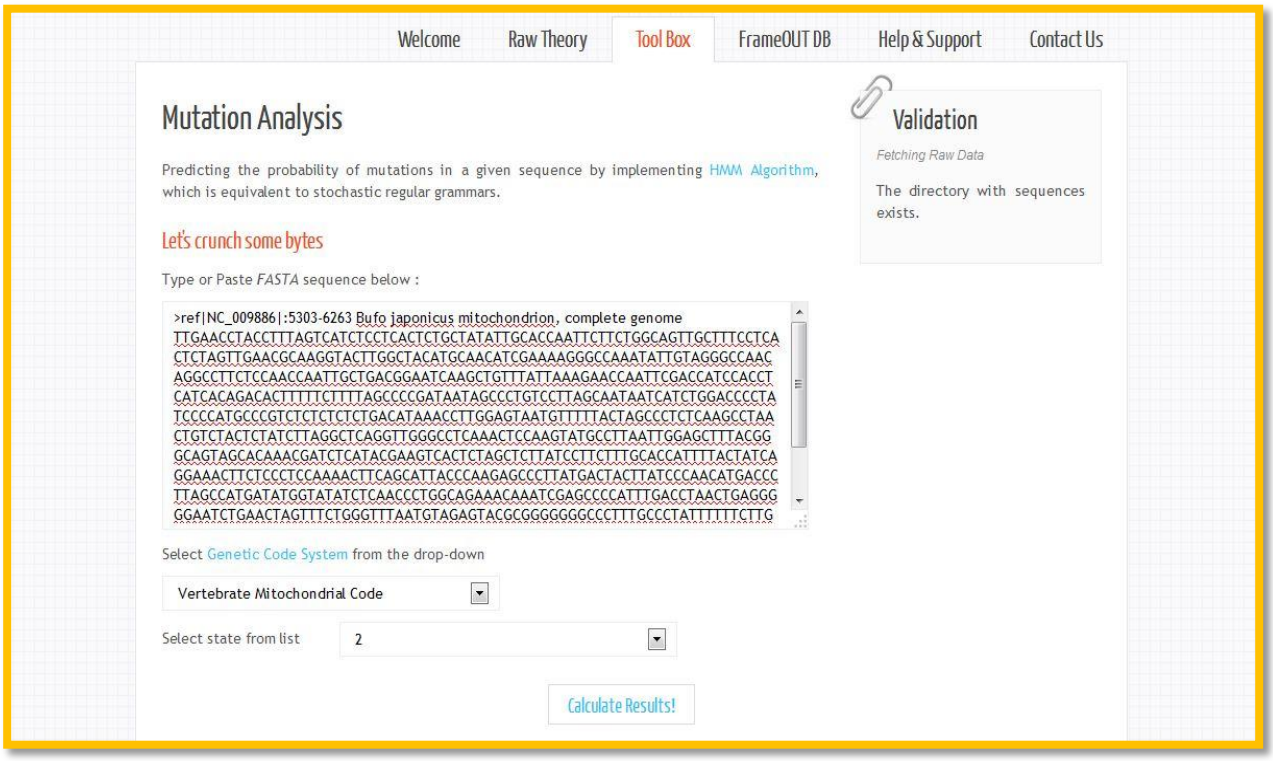


Fig 7.5 *State 2*

Result

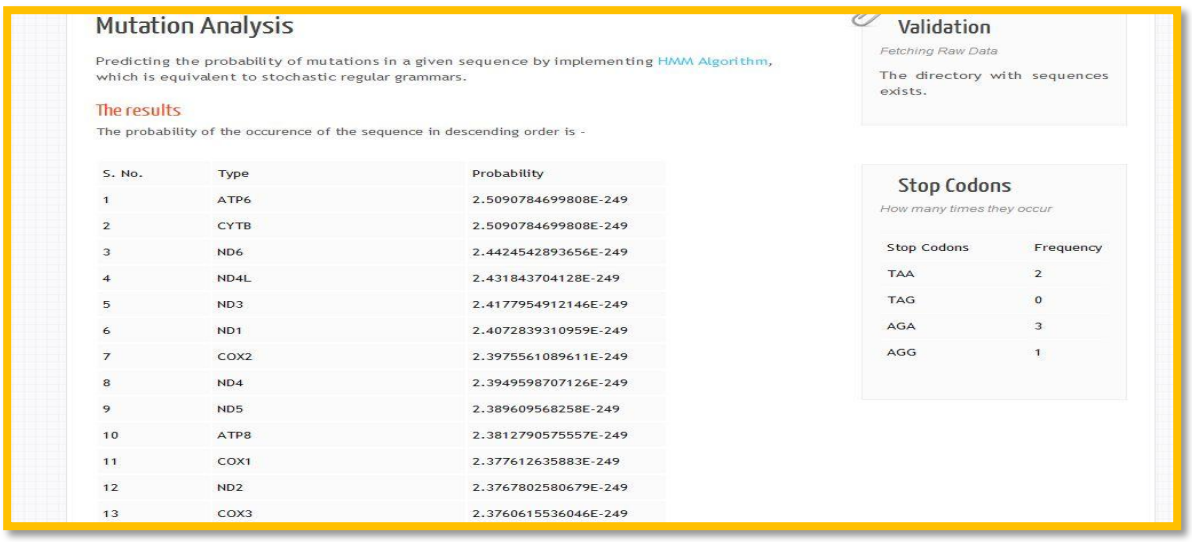


Fig 7.6: *State 2: Result*

7.2 Sample session with FrameOUT DB

User can search inside this database with more specific options. All they need to do is to either enter the name of the disease, gene name, PubMed ID or they can select various options from frame-shift type.

7.2.1 Search By Disease

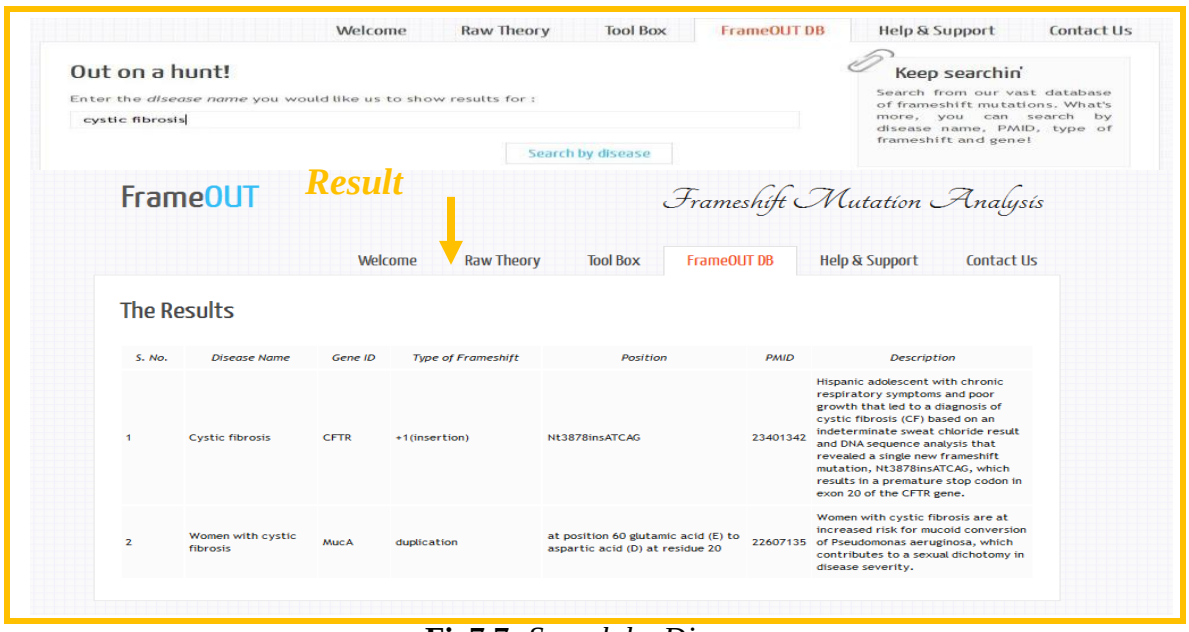


Fig7.7: *Search by Disease*

7.2.2 Search by Gene

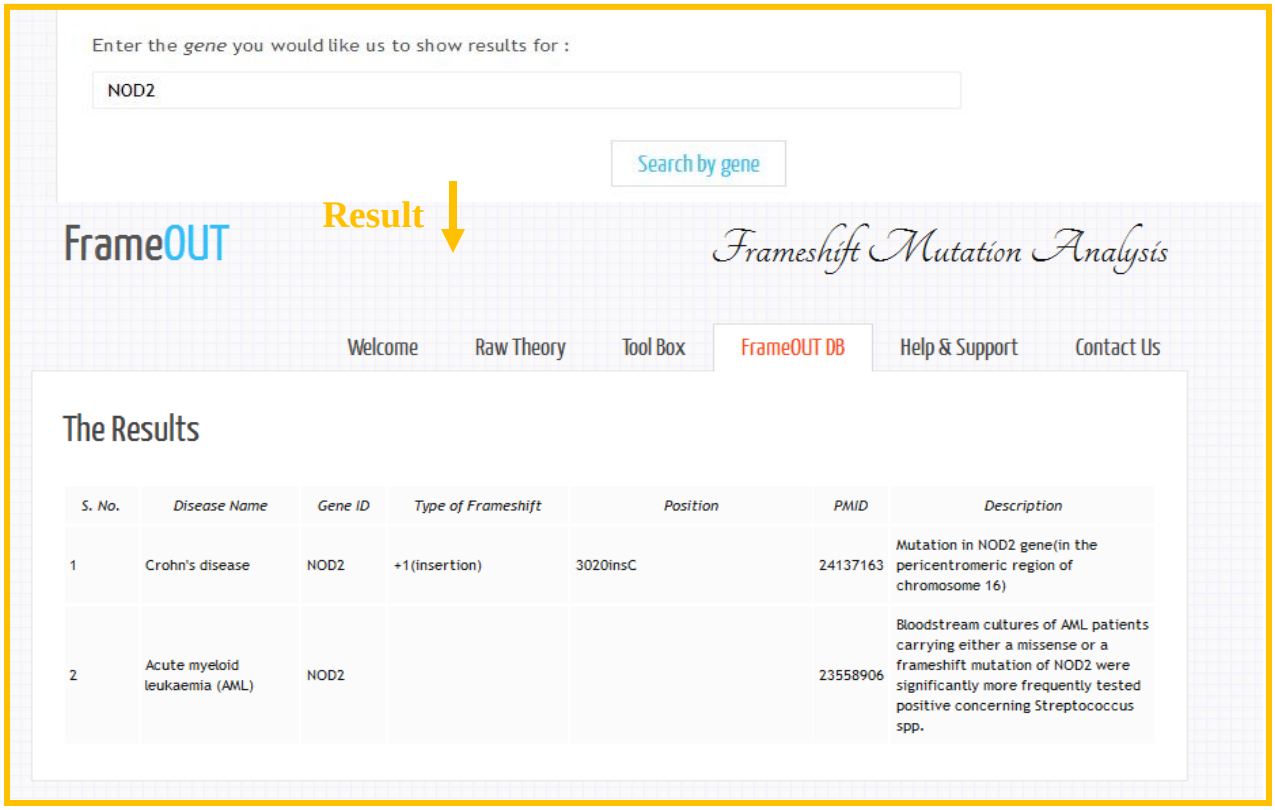
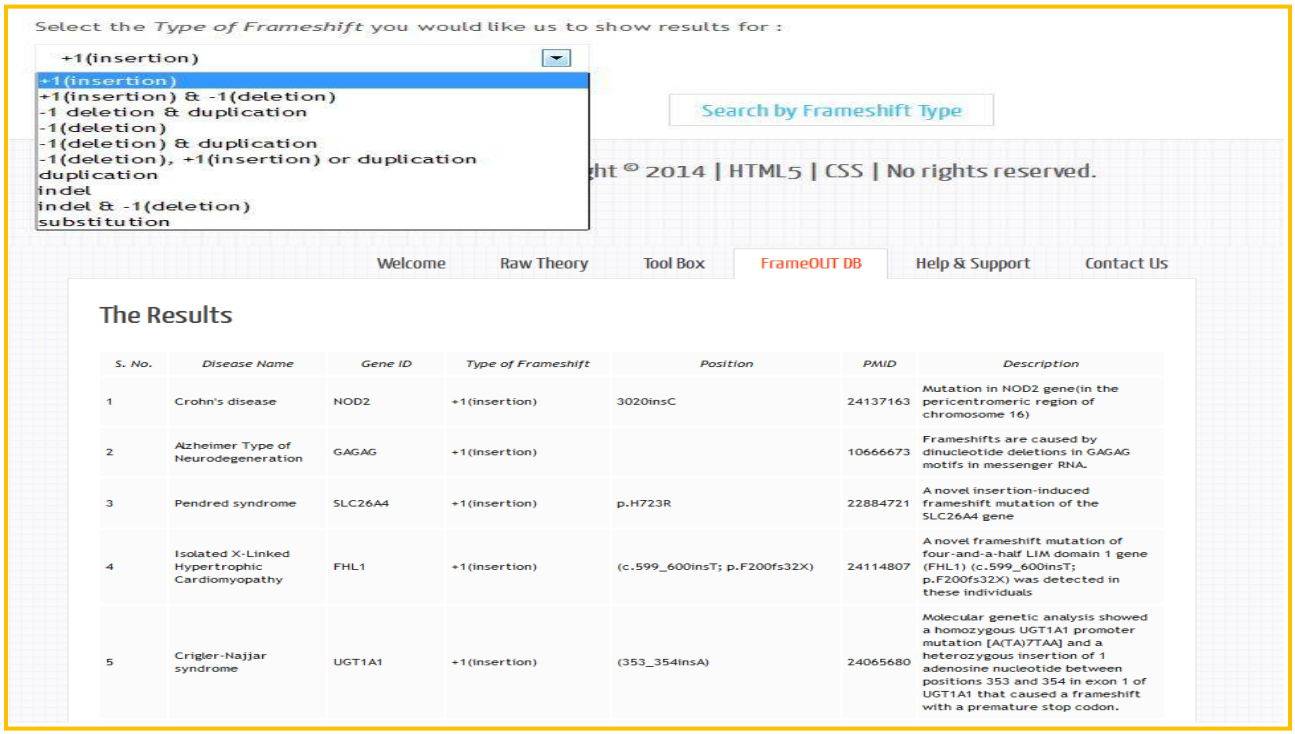


Fig 7.8: *Search by Gene*

7.2.2 Search by PMID

Fig 7.9 *Search by PMID*

7.2.4 Search by Frame-shift Type



S. No.	Disease Name	Gene ID	Type of Frameshift	Position	PMID	Description
1	Crohn's disease	NOD2	+1(insertion)	3020insC	24137163	Mutation in NOD2 gene(in the pericentromeric region of chromosome 16)
2	Alzheimer Type of Neurodegeneration	GAGAG	+1(insertion)		10666673	Frameshifts are caused by dinucleotide deletions in GAGAG motifs in messenger RNA.
3	Pendred syndrome	SLC26A4	+1(insertion)	p.H723R	22884721	A novel insertion-induced frameshift mutation of the SLC26A4 gene
4	Isolated X-Linked Hypertrophic Cardiomyopathy	FHL1	+1(insertion)	(c.599_600insT; p.F200fs32X)	24114807	A novel frameshift mutation of four-and-a-half LIM domain 1 gene (FHL1) (c.599_600insT; p.F200fs32X) was detected in these individuals
5	Crigler-Najjar syndrome	UGT1A1	+1(insertion)	(353_354insA)	24065680	Molecular genetic analysis showed a homozygous UGT1A1 promoter mutation [A(TA)7TAA] and a heterozygous insertion of 1 adenosine nucleotide between positions 353 and 354 in exon 1 of UGT1A1 that caused a frameshift with a premature stop codon.

Fig 7.10: *Search by Frame-shift Type*

CHAPTER 8: CONCLUSION AND FUTURE PROSPECTS

I have developed a new algorithmic tool, FrameOUT, which allows user-friendly exploration, analysis, and visualization of mutational probabilities and hidden stop codons with the mitochondrial vertebrate genome analysis. It is expected that this web based tool would serve as a useful complement for analyzing hidden stop codons in all available genetic code systems, particularly for vertebrate mitochondrial genetic code. HMM's forward algorithm is being implemented to calculate the mutational probability. This forward algorithm takes advantage of the conditional independence rules of HMM to perform the calculation recursively. The algorithm has been implemented with the help of PHP integrated into a user friendly HTML web-page.

FramOUT DB is a collection of diseases caused or are being generated due to frame-shift mutational events such as Corhn's disease, Rett-Syndrome, and Sandhoff disease, etc. Also, users can easily make their search more specific by using various search options enlisted in the database. We plan to update the tool as well as database in near future for more kind of datasets from various other genetic code systems so it will be more useful to the scientific community.

8.1 Availability & Requirement

- **Project name:** FrameOUT- Frameshift Mutation Analysis
- **FrameOUT home page:** http://www.bioinfoindia.org/frameout
- **FrameOUT Tool:** http://bioinfoindia.org/frameout/toolbox.php
- **FrameOUT DB:** http://bioinfoindia.org/frameout/frameoutdb.php
- **Programming Languages:** PHP 5.3.13 / HTML 4.0
- **Web Server:** Apache 2.2 through WampServer
- **Database Server:** MySQL 3.5.1
- **Other requirements:** Web enabled services from standard web browsers

REFERENCES

1. Gupta A, Singh TR (2013) SHIFT: Server for hidden stops analysis in frame-shifted translation. BMC Research Notes 6:68.
2. Singh TR, Pardasani KR (2009) Ambush hypothesis revisited: Evidences for phylogenetic trends. Computational Biology Chem 33: 239-244.
3. Singh TR (2013) Mitochondrial Genomes and Frameshift Mutations: Hidden Stop Codons, their Functional Consequences and Disease Associations. Int J Genomic Med 1: 108. doi: 10.4172/ijgm.1000108
4. Ohno S: Birth of a unique enzyme from an alternative reading frame of the pre-existed, internally repetitious coding sequence. Proc Natl Acad Sci USA 1984, 81:2421–2425.
5. Richard Durbin, Sean R. Eddy, Anders Krogh & Graeme Mitchison *Biological sequence analysis*: Probabilistic models of proteins and nucleic acids
6. Steven Holzner *PHP: The Complete Reference.*
7. J.F., Giddings, M.C., 2001. RECODE: a database of frame-shifting, bypassing and codon redefinition utilized for gene-expression. Nucleic Acids Res. 296, 264–267.
8. Belshaw, R., Pybus, O.G., Rambaut, A., 2007. The evolution of genome compression and genomic novelty in RNA viruses. Genome Res. 17, 1496–1504.
9. Elzanowski, A., Ostell, J., Leipe, D., Soussov, V., 2000. The genetic codes. NCBI.http://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi? mode=c. Farabaugh, P.J., 1996. Programmed translational frame-shifting. Annu. Rev. Genet.30, 507–528.
10. Metzgar, D., Bytof, J., Wills, C., 2000. Selection against frame-shift mutations limits microsatellite expansion in coding DNA. Genome Res. 10, 72–80.
11. Tse H, Cai JJ, Tsoi H-W, Lam EPT, Yuen K-Y: Natural selection retains overrepresented out-offrame stop codons against frame-shift peptides in prokaryotes. BMC Genomics 2010, 11:491.
12. Martina MA, Correa EME, Argaraña CE, Barra JL: *Escherichia coli* Frameshift Mutation Rate Depends on the Chromosomal Context but Not on the GATC Content Near the Mutation Site. PLoS ONE 2012, 7:e33701.
13. Littink KW, van Genderen MM, van Schooneveld MJ, Visser L, Riemslag FC, Keunen JE, Bakker B, Zonneveld MN, den Hollander AI, Cremers FP, van den Born LI: A Homozygous Frameshift Mutation in LRAT Causes Retinitis Punctata Albescens. Ophthalmology 2012, 119:1899–1906.

14. Sagong B, Seok JH, Kwon TJ, Kim UK, Lee SH, Lee KY: A novel insertion induced frame-shift mutation of the SLC26A4 gene in a Korean family with Pendred syndrome. Gene 2012, 508:135–139.

15. Kim SS, Kim MS, Yoo NJ, Lee SH: Frameshift mutations of a chromatin remodeling gene SMARCC2 in gastric and colorectal cancers with microsatellite instability. APMIS 2012. doi:10.1111/j.1600-0463.2012.02953.x. epub ahead of print.

16. Tse H, Cai JJ, Tsoi HW, Lam EP, Yuen KY (2010) Natural selection retains over-represented out-of-frame stop codons against frame-shift peptides in prokaryotes. BMC Genomics 11: 491.

17. Baranov PV, Gesteland RF, Atkins JF (2002) Release factor 2 frame-shifting sites in different bacteria. EMBO Rep 3: 373-377.

18. Russell RD, Beckenbach AT (2008) Recoding of translation in turtle mitochondrial genomes: programmed frame-shift mutations and evidence of a modified genetic code. J Mol Evol 67: 682-695.

19. L Bidou, I Hatin, N Perez, V Allamand, J-J Panthier and J-P Rousset (2004)Premature stop codons involved in muscular dystrophies show a broad spectrum of read through efficiencies in response to gentamicin treatment. *Gene Therapy* (2004) **11,** 619–627.doi:10.1038/sj.gt.3302211

20. Doug A. Brooks, Viv J. Muller, John J. Hopwood (2006) Stop-codon read-through for patients affected by a lysosomal storage disorder. *Trends in Molecular Medicine* Vol.12 No.8.doi.org/10.1016/j.molmed.2006.06.001.

21. Zackary I, Johnson, and Sallie W. Chisholm (2004) Properties of overlapping genes are conserved across microbial genomes. 14:2268-2272.

22. The GRAMMAR package, The *MARKOV package: Markovian Model*, https://www.lri.fr/~genrgens/manual/GRGs-manual-html/node4.html(Accessed: 2013-2014).

23. Louise J Johnson**,** James A Cotton**,** Conrad P Lichtenstein**,**Greg S Elgar**,** Richard A Nichols**,** p David Polly and Steven C Le Comber (2011) Stops making sense: Translational trades-offs and stop codon reassignment. *BMC Evolution Biology* 11:227.

24. Guenter Scheuerbrandt (2004), Exon skipping and reading through stop codons: two research approaches for a therapy of Duchenne Muscular Dystrophy, *DocServer*.

25. [Online]. Available: http:// http://en.wikipedia.org/