

Fall 2015

PATTERN DISCOVERY IN DNA USING STOCHASTIC AUTOMATA

Shweta Shweta

Follow this and additional works at: http://scholarworks.sjsu.edu/etd_projects

Recommended Citation

Shweta, Shweta, "PATTERN DISCOVERY IN DNA USING STOCHASTIC AUTOMATA" (2015). *Master's Projects*. Paper 460.

This Master's Project is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Projects by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

PATTERN DISCOVERY IN DNA USING STOCHASTIC AUTOMATA

A Writing Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Shweta Shweta

December 2015

© 2015

Shweta Shweta

ALL RIGHTS RESERVED

The Writing Project Committee Approves the Project Titled

PATTERN DISCOVERY IN DNA USING STOCHASTIC AUTOMATA

By

Shweta Shweta

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

December 2015

Dr. T. Y. Lin

Department of Computer Science

Prof. Chris Tseng

Department of Computer Science

Dr. Howard Ho

IBM Technologies Ltd.

ABSTRACT

PATTERN DISCOVERY IN DNA USING STOCHASTIC AUTOMATA

By Shweta Shweta

We consider the problem of identifying similarities between different species of DNA. To do this we infer a stochastic finite automata from a given training data and compare it with a test data. The training and test data consist of DNA sequence of different species. Our method first identifies sentences in DNA. To identify sentences we read DNA sequence one character at a time, 3 characters form a codon and codons form proteins (also known as amino acid chains). Each amino acid in proteins belongs to a group. In total we have 5 groups' polar, non-polar, acidic, basic and stop codons. A protein always starts with a start codon ATG that belongs to the group polar and ends with one of the stop codons that belongs to the group stop codon. After identifying sentences our method converts it into a symbolic representation of strings where each number represents the group to which an amino acid belongs to. We then generate a PTA tree and merge equivalent states to produce a Stochastic Finite Automata for a DNA.

In addition to producing SFA, we apply secondary storage to handle huge DNA sequences. We also explain some concepts that are necessary to understand our paper.

ACKNOWLEDGMENTS

For this project I am indebted to Dr. T. Y. Lin for his constant guidance and supervision as well as providing essential information regarding the project and helping me in completing this project. I would also like to express my gratitude towards my parents for their encouragements and co-operation in finishing the project.

In addition, I would like to express my thanks to the committee members Prof. Chris Tseng and Dr. Howard Ho for taking time out of their busy schedule and attending my defense. I am grateful for their suggestions and encouragements.

My thanks and appreciation also goes to Martin Luther King Library, San Jose for providing materials that helped me in my research. I would also like to thank Department of Computer Science, SJSU for their relentless supervision.

I would also like to extend my gratitude towards my peers for making suggestions whenever I hit a road block and providing constant encouragement in finishing this project.

Table of Contents

Introduction	1
1. Definition and Notations	3
1.1. Finite Automata.....	3
1.2. Stochastic Finite Automata	3
1.3. Prefix Tree Acceptor	5
1.4. Storage and Buffer Management.....	6
2. Translating DNA to Strings	8
2.1. DNA Structure.....	8
2.2. DNA Translation and Transcription.....	9
2.3. Identifying Strings in DNA	12
2.4. Example – Hemoglobin.....	13
2.5. Some information about Proteins.....	13
3. Algorithms Explained.....	14
3.1. Blue Fringe Algorithm	14
3.2. Algorithm Alergia	14
3.3. Run time Complexity	19
3.4. Secondary Storage.....	19
4. Extension to Alergia: Buffer Usage.....	21
4.1. Calculating Buffer Size and number of Buffers to be used.	21
4.2. How to perform Alergia using Buffers	21
4.3. Sentences in DNA	22
4.4. Steps: How to Compare DNA's from Test Data with Stochastic Finite Automata of Learning Document.	23
5. Experiment Results and Verification.....	25
5.1. Data Source	25
5.2. Test Set up for Experiments	25
5.3. Result Analysis for Human, Chimpanzee, Cat, Cow, Mouse, Fruit Fly and Chicken 25	
a) Chromosome Count and size of Genome Sequence	26
b) Number of nodes created and merged in PTA tree	26
5.4. Result Analysis for Cat with Cat, Cattle, Chicken, Fruit fly and Chimpanzee ...	32
a) Number of nodes created and merged in PTA tree for Cat	32
6. Similarity between different species can reduce with time	35
7. Human Genetic Variation	39

8.	Clustering of Races.....	45
8.1.	DBSCAN algorithm for clustering.....	48
8.2.	Clusters obtained for Epsilon = 0.2 and Number of Sample = 1	51
8.3.	Clusters obtained for Epsilon = 1 and Number of Samples = 1.....	52
8.4.	Clusters obtained for Epsilon = 1.5 and minimum Sample = 100	53
9.	Future Work and Enhancements.....	67
9.1.	Use Database	67
9.2.	Number of Strings Accepted	69
9.3.	Number of String Arriving at each node	70
9.4.	Benefit of Using Database.....	70
10.	Conclusion.....	71
11.	References	73

Table of Equations

Equation 1: Probability Constrains for strings in SFA	4
Equation 2: Probability $p(w)$ for the string w to be accepted by A	4
Equation 3: Stochastic Regular Language generated by means of an SFA.....	4
Equation 4: Condition for a node to be deemed as useless	5
Equation 5: Equivalence condition for two SRL's	5
Equation 6: Equivalence of 2 Nodes	18
Equation 7: Hoeffding Bound	18

Table of Figures

Figure 1: DFA vs NFA	3
Figure 2: PTA Tree for $\{110, \lambda, \lambda, \lambda, 0, \lambda, 00, 00, \lambda, \lambda, \lambda, 10110, \lambda, \lambda, 100\}$	6
Figure 3: DNA Structure	8
Figure 4: Chemical structure of DNA; hydrogen bonds shown as dotted lines	9
Figure 5: Amino Acid Groups DNA	10
Figure 6: Alergia compatible checks $q_i \equiv q_j$	17
Figure 7: Alergia different checks similarities of two frequencies	19
Figure 8: Merged Nodes vs Alpha, orange dot represents best result	28
Figure 9: Difference in results	31
Figure 10: Number of nodes merged for CAT	33
Figure 11: Blast comparison of mouse and human	36
Figure 12: Conclusions and applications of the feline genome sequence	37
Figure 13: Similarities between Indian and African, Asian	40
Figure 14: Similarity between Indians and French, Australian	41
Figure 15: French DNA similarity with Australian, Indian and Africa.....	43
Figure 16: Australian DNA's compared against Australian and Italian DNA (1)	46
Figure 17: Australian DNA's compared against Australian and Italian DNA (2)	47
Figure 19: All DNA sequences clustered into 1 cluster	66
Figure 20: PTA tree	67

Appendix of Tables

Table 1: Types of Amino Acids	11
Table 2: Wget command used to obtain data	25
Table 3: Chromosome Count and Size of Genome	26
Table 4: Number of Node Merged for Alpha 0.2 to 0.7	27
Table 5: Number of Node Merged for Alpha 0.8 to 1.3.....	27
Table 6: Number of Node Merged for Alpha 1.4 to 1.9.....	27
Table 7: Number of Node Merged for Alpha 2.0	27
Table 8: Result Summary	30
Table 9: Difference in results	31
Table 10: Number of nodes merged for Cat	32
Table 11: Similarity between Cat and Cattle, Chicken, fruit Fly and Chimpanzee	34
Table 12: Similarity between Indian and African, Asian DNA	40
Table 13: Similarities of Indian with French and Australians.....	41
Table 14: Comparison of French DNA with Indian, African, South American.....	42
Table 15: Cluster obtained for Epsilon = 1 and Number of Samples = 1	52
Table 16: Associate Transition Matrix	69
Table 17: Number of String Accepted by Nodes in PTA.....	70
Table 18: Number of String arriving at each node	70

Introduction

Extracting information from data is a well-established discipline that can be traced back to the sixties. It is also known as Data Science where as stated by Vasant Dhar, PhD, editor-in-Chief of Big Data “Data Science is a study of generalizable extraction of information from data”.

Companies like Google, Facebook, Amazon and Yahoo employs Data Scientists to extract information from data mainly using pattern recognition. Based on the information obtained they make business decisions that lead to a better customer experience and hence increased revenue. An interesting example of using this information is the recommendation algorithm used by Amazon to suggest products based on similar users. Netflix also uses recommendations algorithms to propose movies most likely to suit a user’s taste based on similar movies watched by the user. Many e-commerce and job searching companies like LinkedIn, Spotify, Indeed uses various data science to deliver an enhanced experience to the consumers.

Our research is based on training from input sample and using automata theory and state merging algorithm to produce stochastic finite automata that represents whole DNA of a species. Based on that we will compute the percentage of similarity in other words highlight the differences between two species. In this project, we will create the automata of DNA sequence by appropriately representing the base pair sequences in the form of symbols. We will further create a PTA (Prefix Tree Acceptor) to compare the sequence with various other species.

The reason behind using DNA as data set is that we need better methods to point out the differences between DNA of two species. The default standard for comparing DNA is Blast [2] that uses an empirical approach where it finds a few short matches between two sequences and then starts to look for more similar alignments locally, which means it does not cover the entire search space. It starts with the matching words of size seven to eleven bases in order to match a query with

subject sequence. From there, it starts comparing the nearby sequence areas to find more matches. It only aligns locally, considers few gaps and substitutions. It represents results in the form of x or y which can be represented as x % similarity over y% sequence. It does not even cover whole sequence. So, we need a technique that not only compares the whole sequence, but also points out exact locations of differences between species. This data is optimized especially for state merging algorithms because DNA is made up of repetitive sequences that can be merged if found equivalent.

This paper has been divided into various sections for a better understanding of the concepts involved. Section 1 provides background on Finite automata. It also explains PTA tree and stochastic finite automata. Section 2 focuses on DNA and its structure, it outlines how to identify sentences in DNA. Section 3 comprises the algorithms involved in our project mainly Algeria and Blue-Fringe Algorithm.

Section 4, explains implementation details of extension done to Alergia in order for it to handle huge data. Section 5 presents experimental results and verification of results with Blast Output. Finally, Section 6 the fact that percent similarity between species changes with time and depend on factors like availability of genome and genome evolution. Section 7 describes human genetic variations focusing on “out of Africa: theory. Section 8 describes some future enhancements that can be done to this project. Finally, Section 9 describes conclusion of this project.

1. Definition and Notations

1.1. Finite Automata

A finite automata is represented by a tuple of 5 elements $(Q, \Sigma, \delta, q_0, F)$ where Q represents Set of Finite States, Σ represents alphabets, δ is transition function which further represents mapping of $Q \times \Sigma$ to 2^Q . Initial state is represented as q_0 , and F is a subset of Q that denotes final or accepting states.

Two variations of Automata are DFA and NFA where DFA represents deterministic automata and NFA represents Non-Deterministic Automata.

Deterministic Finite Automata	Non Deterministic Finite Automata
Characterized as a 5 tuple state: $\langle S, A, T, s_0, F \rangle$	Characterized as a 5 tuple state: $\langle S, A, T, s_0, F \rangle$
S is the set of states	S is the set of states
A is the alphabet	A is the alphabet
T is the transition function: $S \times A \rightarrow S$	T is the transition function: $S \times (A \cup \{\epsilon\}) \rightarrow PS$
s_0 is the initial state	s_0 is the initial state
F is the set of accepting states.	F is the set of accepting states.

Figure 1: DFA vs NFA

1.2. Stochastic Finite Automata

In the project the main focus is [7] learning pattern from input and generating a Stochastic Finite Automata. Generating automata from a sample set S , if only S_+ (set of positive samples) are

available, then adding new samples to S, can only present a finite number of changes in hypothesis. Therefore, a compatibility criterion is needed that can help in rejecting language L' whose difference with L lays on L – L'.

For this purpose, [7] Stochastic Finite Automaton is used where (SFA) Stochastic Finite Automata $A = (A, Q, P, q_1)$ consists of an Alphabet A, a finite set of nodes $Q = \{q_1, q_2, \dots, q_n\}$ with q_1 the initial node and a set of Probability matrices $p_{if}(a)$ given the probability of a transition from node I to node q_j led by the symbol a in the alphabet. If a p_{if} denotes the probability that the string ends at node Q_i , the following constraints applies:

$$p_{if} + \sum_{q_j \in Q} \sum_{a \in A} p_{ij}(a) = 1 \quad (1)$$

Equation 1: Probability Constrains for strings in SFA

$$\begin{aligned} p(w) &= \sum_{q_j \in Q} p_{1j}(w) p_{if} \\ p_{ij}(w) &= \sum_{q_k \in Q} \sum_{a \in A} p_{ik}(wa^{-1}) p_{kj}(a) \end{aligned} \quad (2)$$

Equation 2: Probability p (w) for the string w to be accepted by A

Stochastic Regular Language generated by automaton A is defined as:

$$L = \{ w \in A^* : p(w) \neq 0 \} \quad (3)$$

Equation 3: Stochastic Regular Language generated by means of an SFA

A node q_i is useless if there are no strings $x, y \in A^*$ such that the below equation is satisfied.

$$\sum_j p_{1j}(x)p_{ij}(y)p_{if} \neq 0 \quad (4)$$

Equation 4: Condition for a node to be deemed as useless

Finally, two SRL's are equivalent if they provide identical probability distribution over A^* .

$$L_1 \equiv L_2 \Leftrightarrow p_1(w) = p_2(w) \forall w \in A^* \quad (5)$$

Equation 5: Equivalence condition for two SRL's

1.3. Prefix Tree Acceptor

In our project Prefix Tree Acceptor is used for the representation of DFA in the project. A PTA can be compared with a Trie data structure i.e. it represents DFA in a tree like form by taking all the prefixes in a given sample as states and constructing the smallest DFA [6] ($\forall q \in Q, |\{q' : \delta(q', a) = q\}| \leq 1$).

Figure 1[6] represents a PTA for Input Set $S = \{110, \lambda, \lambda, \lambda, 0, \lambda, 00, 00, \lambda, \lambda, \lambda, 10110, \lambda, \lambda, 100\}$

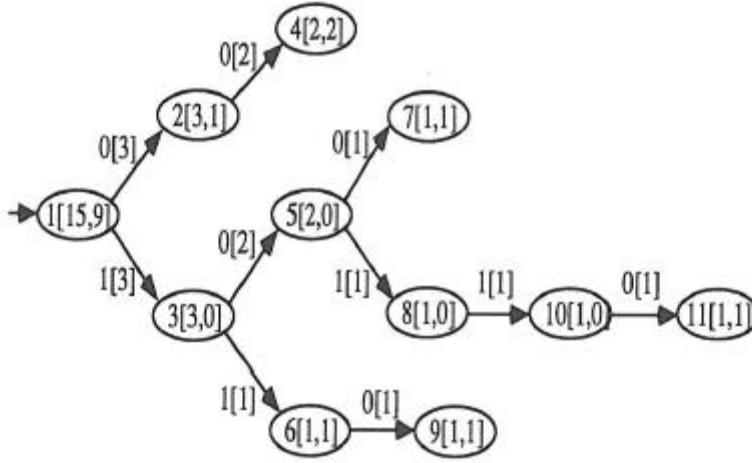


Figure 2: PTA Tree for $\{110, \lambda, \lambda, \lambda, 0, \lambda, 00, 00, \lambda, \lambda, \lambda, 10110, \lambda, \lambda, 100\}$

In the PTA tree represented above we created our own data type to represent a node. A node stores following information.

- Number of string reaching at each node – represented by 15 in node 1 where 15 is sum of all the strings that passed from node 0. In our example those 15 strings are 110, λ , λ , λ , 0, λ , 00, 00, λ , λ , λ , 10110, λ , λ , 100.
- Number of String ending at each node – represented by 9 in node 0 where 9 represents number of node that ended at node 0. In the example given above it is all the empty strings represented by λ .
- Number of string outgoing from a node over alphabet – represented by 0[3] on arrow that connects node 0 to node 1, where 0[3] implies that given alphabet 0 node 0 has 3 strings that transits from node 0 to node 1.

1.4. Storage and Buffer Management

The data in our project reside in secondary storage mainly hard disk of the system. However, to perform useful operations on data, that data needs to be in main memory. In this report we have shown a comparison of human against six other species in section 5.3. The overall size of this data

is equal to 11.789 GB. The system on which this program is being executed has a main memory of 8GB.

In order to process all data and generate output we will use buffers in our program. Buffer management is a concept in databases where a manager is responsible for partitioning memory into buffers. Thus, all components that need information from the disk will interact with buffers or buffer manager, either directly or indirectly through execution engine.

Further explanation on how our program is using buffers is provided in Section 4.

2. Translating DNA to Strings

2.1. DNA Structure

DNA [8] is a chemical that chromosomes and genes are made of. DNA itself is made of four simple chemical units shortened as A, G, C, and T. The chromosome is simply a long piece of DNA that cells can easily replicate. A gene is a stretch of DNA on a chromosome that has guidelines on how to make a protein. A protein is a molecular machine that does a specific job. A protein is made up of amino acids (as represented in Figure 7) that are stuck together. The order and number of 20 different amino acids determine what a protein can do. For example Amalyse helps in digesting food.

Figure 4 and 5 represents the DNA structure [9]

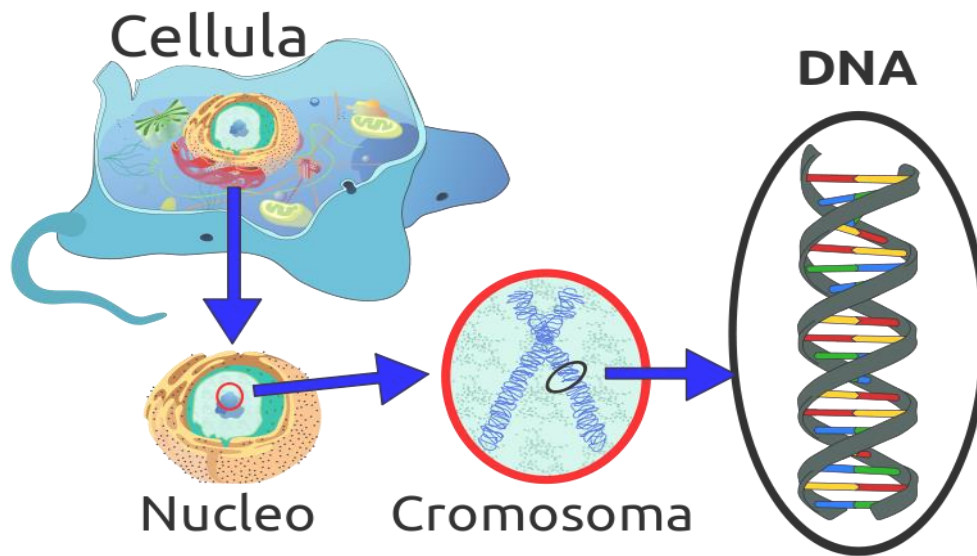


Figure 3: DNA Structure

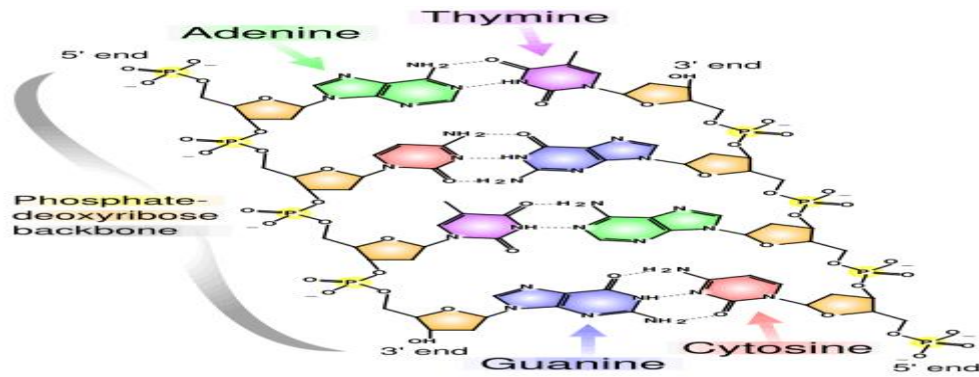


Figure 4: Chemical structure of DNA; hydrogen bonds shown as dotted lines

2.2.DNA Translation and Transcription

A gene in a DNA code for a single protein molecule known as polypeptide bond which is also used for protein synthesis. It comprises of 2 steps Transcription and Translation [13]. Transcription occurs when a sequence of one gene is replicated in an RNA molecule whereas Translation acts as a code for the formation of the amino acid chain (a polypeptide) [14].

The first step in synthesizing protein is translation of DNA to RNA, which is done by replacing U (Uracil) with T (Thymine). The second step is identifying amino acids. A set of 20 amino acids exists today as shown in figure 6.A chain of amino acids also known as polypeptide string then makes up a protein.

		second base in codon					
		T	C	A	G		
first base in codon	T	TTT Phe	TCT Ser	TAT Tyr	TGT Cys	T	third base in codon
		TTC Phe	TCC Ser	TAC Tyr	TGC Cys	C	
		TTA Leu	TCA Ser	TAA stop	TGA stop	A	
		TTG Leu	TCG Ser	TAG stop	TGG Trp	G	
	C	CTT Leu	CCT Pro	CAT His	CGT Arg	T	
		CTC Leu	CCC Pro	CAC His	CGC Arg	C	
		CTA Leu	CCA Pro	CAA Gln	CGA Arg	A	
		CTG Leu	CCG Pro	CAG Gln	CGG Arg	G	
	A	ATT Ile	ACT Thr	AAT Asn	AGT Ser	T	
		ATC Ile	ACC Thr	AAC Asn	AGC Ser	C	
		ATA Ile	ACA Thr	AAA Lys	AGA Arg	A	
		ATG Met	ACG Thr	AAG Lys	AGG Arg	G	
	G	GTT Val	GCT Ala	GAT Asp	GGT Gly	T	
		GTC Val	GCC Ala	GAC Asp	GGC Gly	C	
		GTA Val	GCA Ala	GAA Glu	GGA Gly	A	
		GTG Val	GCG Ala	GAG Glu	GGG Gly	G	

Figure 5: Amino Acid Groups DNA

Full forms of Amino acids are shown in the table below

Amino Acid	one letter code	three letter code
L-alanine	A	Ala
L-arginine	R	Arg
L-asparagine	N	Asn
L-aspartic acid	D	Asp
L-cysteine	C	Cys
L-glutamine	Q	Gln
L-glutamic acid	E	Glu
glycine	G	Gly
L-histidine	H	His
L-isoleucine.	I	Ile
L-leucine	L	Leu
L-lysine	K	Lys
L-methionine	M	Met
L-phenylalanine	F	Phe
L-proline	P	Pro
L-serine	S	Ser
L-threonine	T	Thr
L-tryptophan	W	Trp
L-tyrosine	Y	Tyr
L-valine	V	Val

Table 1: Types of Amino Acids

Figure 6 represents 20 amino acids that are present in DNA. These Amino acids can be categorized into 4 major sets mainly polar, non-polar, acidic and basic. There is one additional group that comprises of stop codons which marks the end of a protein.

0. Non Polar: Glycine, Alanine, Valine, Leucine, Isoleucine, Proline, Methionine, Phenylalanine, Tryptophan
1. Polar (neutral): Serine, Threonine, Cysteine, Asparagine, Glutamine, Tyrosine
2. Acidic (polar acidic): Aspartic acid, Glutamic acid
3. Basic (polar basic): Lysine, Arginine, Histidine
4. Stop codons : TAA, TGA, TAG

Among 20 amino acids only 8 acids are essential acids: Isoleucine, leucine, lysine, methionine, phenylalanine, threonine, tryptophan and Valine. Remaining 12 are not essential [16]. Various functions in our body like maintaining healthy hair, overseeing insulin are done by essential proteins.

2.3. Identifying Strings in DNA

A DNA is read 3 characters at a time, these three letters are known as Codons and provides instructions for putting together a protein. To convert DNA into strings, we will follow below mentioned steps:

1. Read DNA 3 letters at a time, these 3 letters are known as Codons.
2. A string begins when a *start codon* is encountered and ends when a *stop codon* is encountered. A start codon is represented as ATG (methionine) whereas a stop codon is represented as TAA or TAG or TGA.
3. If letter read is 'N' or 'n' than consider it as an empty string and ignore it where 'N' represents a DNA sequence that has not been discovered yet.

Following above mentioned 3 steps we can find strings in DNA, Strings are nothing but proteins that start with *start codon* and ends with *stop codon*.

2.4.Example – Hemoglobin

Hemoglobin comprises 444 letter of DNA and starts with the following string:

ATGGTGCATCTGACTCCTGAGGAG.....

Reading it 3 characters (codon) at a time we get:

ATG GTG CAT CTG ACT CCT GAG GAG....

Parsing protein to its group, we get following representation:

0 0 3 0 1 0 2

Each of these codons relays information to the cells as to which amino group should be added next.

For example, ATG tells the cell to add methionine amino acid. Parsing DNA string to its amino acid group we get:

Met-Val-Leu-Thr-Pro-Glu-Glu...

2.5.Some information about Proteins

Protein ranges from biggest protein named titin, which comprise 34,350 amino acids to insulin, which contains less than 100 amino acids. In the program, all proteins irrespective of their sizes has been considered. For example, if a protein contains only start and stop codon with no codons in between, they are also consider as a string. Such a protein can be represented as “*ATG TAG*” with no codons in between.

3. Algorithms Explained

3.1. Blue Fringe Algorithm

Blue Fringe [10] is a state merging algorithm that merges two equivalent states.

Input - PTA tree

Output - Tree with merged equivalent nodes.

- Start with the root node, color it red. Color its children blue, color all other nodes white.
- Blue nodes are parents of isolated trees, all non- red nodes are blue in color.
- Compute the score for merging a red/blue pair.
- Promote a blue node to red if it is unmerge able otherwise merge a blue node with red node.

3.2. Algorithm Alergia

To identify stochastic regular languages of DNA and distribution of probabilities of strings in the language Alergia [5] algorithm is used. The algorithm builds a PTA tree from sample set, the tree is traversed in lexicographic order. It then merges states that are equivalent in nature. In other words, we need to find partitions Π' of nodes in T such that $\Pi' (T)$ coincides with $M (L)$, the Stochastic Finite Automata (It is a generally known fact that: given a large enough sample S , there exists a partition Π' of nodes in T such that $\Pi' (T)$ coincides with $M (L)$ [7]).

In order to find partitions, traversal idea will be used from Blue Fringe Algorithm, however, there will be one more addition to the algorithm, i.e. adding lexicographic ordering while traversing the tree in order to maintain similar order for nodes in automaton.

Now proceed as follows to find partitions

- Merge node q_i and q_j (only if (7) holds), varying sub index j from 2 to t and then for every j changing i from $j - 1$, where $t = |T|$, T = nodes in PTA.

- In this way only $\frac{1}{2} t (t - 1)$ comparisons are made to obtain the Stochastic Finite Automata.

Algorithm Alergia

Input:

S : sample set of strings

α : 1-confidence level

output:

stochastic DFA

begin

A = stochastic Prefix Tree Acceptor from S

do (for j = successor (first node (A) to last node (A))

do (for i = firstNode(A) to j)

if compatible (i, j)

merge A(i, j)

determinize (A)

exit (i - 1) loop

end if

end for

end for

return A

end Algorithm

Figure 7: Alergia [5] Algorithm

To check compatibility of two nodes (q_i, q_j), Alergia [Figure 7] calls *algorithm Compatible* [Figure 8]. *Algorithm compatible* is recursive in nature, in addition to checking roots of two nodes (q_i, q_j), it also checks all the children of q_i and q_j for compatibility. If root and all the children of q_i and q_j are compatible, it merges q_i and q_j , otherwise it promotes them to red states.

Algorithm compatible

Input:

i, j : nodes

output:

boolean

begin

if different ($n_i, f_i(\#)$, $n_j, f_j(\#)$)

return false

end if

do ($\forall_a \in A$)

if different($n_i, f_i(a)$, $n_j, f_j(a)$)

return false

end if

if not compatible ($\delta(i,a)$, $\delta(j,a)$)

return false

end if

end do

return true

end algorithm

Figure 6: Alergia compatible checks $q_i \equiv q_j$

Two nodes are considered equivalent if they have equal transition probabilities for every symbol $a \in A$ and destination nodes are also equivalent. Since the number of calls to recursion are limited to number of alphabets in the language L , recursion is finite in nature.

$$q_i \equiv q_j \Rightarrow A \begin{cases} p_i(a) = p_j(a) \\ \delta_a(i) \equiv \delta_a(j). \end{cases}$$

Equation 6: Equivalence of 2 Nodes

(6)

However, we need to provide a range in which two nodes are considered compatible as training set is prone to statistical fluctuations. For that purpose, Hoeffding bound is used.

$$\left| p - \frac{f}{n} \right| < \sqrt{\frac{1}{2n} \log \frac{2}{\alpha}}$$

Equation 7: Hoeffding Bound

(7)

In the above equation p is the probability, f is frequency, and n is the number of samples and α is a confidence range. The Hoeffding bound check is done for both outgoing arcs $f_i(a)$ and termination frequencies $f_i(\#)$. Thus $A + 1$ comparisons are done at every node where $|A|$ is the size of the alphabet. This check is represented via *algorithm different* in Figure 9.

Algorithm different

input:

n, n' : number of strings arriving at each node

f, f' : number of strings ending/following a given arc

output:

Boolean

Begin

$$\text{Return } \left| \frac{f}{n} - \frac{f'}{n'} > \sqrt{\frac{1}{2}} \log_{\alpha}^2 \left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{n'}} \right) \right|$$

End algorithm

Figure 7: Alergia different checks similarities of two frequencies

3.3.Run time Complexity

Time needed by the algorithm to output n hypothesis:

Worst Case = $O(s^3)$

Average Case = $O(s)$, where s is size of the sample.

Even though in worst case complexity is $O(s^3)$ all the experiments show a linear complexity.

3.4.Secondary Storage

A human DNA comprises of about 3 billion bases that are arranged along the chromosomes. One million bases of DNA, also known as mega base are roughly equivalent to 1 MB. So, the total amount of space required to store a human DNA is 3GB. In order to avoid “Stack Overflow” errors or “Out Of Memory”, we will have to provide secondary storage to our program. Secondary

storage, also known as auxiliary memory is a storage that is not under direct control of a computer's central processing unit.

4. Extension to Alergia: Buffer Usage

4.1.Calculating Buffer Size and number of Buffers to be used.

To calculate the size of buffer we will perform a check. If size of file is less than 512MB then there is no need of going through the overhead of managing and creating buffers as our program can handle performing all the operations on file of size 512MB. However, if size of file is greater than 512MB then we need to compute the number of buffers needed to execute the program smoothly.

For this purpose, we have to consider two things

- a) Size of buffer should not be too small otherwise it will lead to a lot of buffers.
- b) Size of the buffer cannot be too large otherwise it will results in too few buffers that effect efficiency of the program.

For our program, we have buffer Size to be 512 MB. Number of slices are calculated as follows:

$$\text{Number of Buffers} = \frac{\text{Size of File}}{512}$$

4.2.How to perform Alergia using Buffers

First step is to convert DNA sequence into a symbolic representation of proteins. In order to do that we will follow below mentioned steps:

Generate Symbol File

1. Calculate Number of buffers using section 4.1.
2. While (Number of Buffers!= 0)
3. Read data from file and store in a buffer1 of size 512 MB
4. While (buffer has next)
5. Identify DNA Sentences in buffer as shown in section 4.3.

6. Convert DNA sentences to their symbol representation and store in buffer2
7. End while
8. Write buffer2 to a file name symbolFileRepresentation.txt in append mode
9. Flush buffer 2 and buffer 1
10. Number of Buffers = Number of buffers - 1
11. End while

Generate PTA from symbol file

After creating symbol file, each sentence in symbol file is passed to the PTA tree generator that generates a PTA tree.

1. While (symbol file has sentences)
2. Pass sentences to the tree generator
3. End

Apply Alergia to produce SFA

Once PTA is generated Alergia is applied to it in order to generate a SFA. It is possible to do this in memory for a system with specifications provided in section 5.2.

4.3.Sentences in DNA

- Here are some of DNA sequences which are represented in the form of their corresponding amino acid group, ATG represents start codon and TGA, TAA, and TAG represents stop codons. Each amino acid can be represented in terms of its group number as specified in section 2.3.

Sequence 1: ATG GCG AGA CAG CCA TGA

Representation: 0 0 3 1 0 4

Sequence 2: ATG CTC AGG GAT TAA

Representation: 0 0 3 2 4

Sequence 3: ATG GCC CAC TGA

Representation: 0 0 3 4

Sequence 4: ATG AAA TGC TGG TAG

Representation: 0 3 1 0 4

Sequence 5: ATG CGT ACG CAT ACC TAA

Representation: 0 3 1 3 1 4

Sequence 6: ATG GCC TGC ACG TAA

Representation: 0 0 1 1 4

Sequence 7: ATG TGG GAT TAG

Representation: 0 0 2 4

Sequence 8: ATG TTT AAG TGA

Representation: 0 0 3 4

- Above sequence can then be passed to our program to generate a PTA tree. Equivalent states are then merged by Alergia[5] to generate a stochastic finite automata. This automata is an estimate of initial one.

4.4.Steps: How to Compare DNA's from Test Data with Stochastic Finite Automata of Learning Document.

1. Follow Instructions in Section 4.1 to generate Stochastic Finite Automata for a Learning Document. For example – Human DNA.
2. Read DNA files in a testing folder one by one. For each file, read it in 512 MB chunks, parse it to DNA sentences. Once 512 MB is processed, write (in append mode) the parsed

DNA strings into a file, clean the buffer and start again. We will call the final file as symbolFile.txt.

3. For each symbol File in the training set, pass strings in symbol file to the Stochastic Finite Automata generated in the Step1. Maintain count of Number of Strings Accepted and total Number of String Passed in the Tree.
4. Percentage similarity of 2 species is the $\text{Number of String Accepted} / \text{Total Number of Strings} * 100$.
5. After processing 1 symbol File, clean the buffer and start again with the next file.
6. Display the result when finished.

5. Experiment Results and Verification

5.1.Data Source

The dataset is obtained from <http://hgdownload.soe.ucsc.edu/downloads.html>. Command used to download Data is wget. Example – To download whole Genome for Dog type following command in command prompt.

```
Wget --timestamping  
'ftp://hgdownload.cse.ucsc.edu/goldenPath/canFam3/bigZips/canFam3.fa.gz'
```

Table 2: Wget command used to obtain data

5.2.Test Set up for Experiments

We ran this program on a system with below described configurations.

System Information

- Processor – Intel® Core™ i7-2630QM CPU @ 2.00 GHZ 2.00 GHZ
- RAM – 8.00 GB (7.89 GB usable)
- System Type – 64- bit operating system, x64- based processor
- Operating System – Windows 10 Pro

5.3.Result Analysis for Human, Chimpanzee, Cat, Cow, Mouse, Fruit Fly and Chicken

In this project we compared human DNA with Cat, Cow, Mouse, Fruit Fly and chicken species. Below table represents the number of chromosomes in each species. For example, the number of chromosomes in human is 23, due to double helix structure of DNA it is represented as

46 i.e. a multiple of 2. Last row of the table represents size required to represent DNA in terms of computer memory. For example, human DNA is approximately 3 GB in size.

One thing to note here is that entire genome sequence of all the species has not been identified yet. The results are based on genome sequences identifies till date.

a) Chromosome Count and size of Genome Sequence

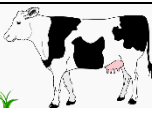
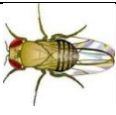
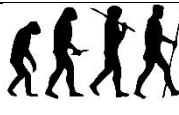
	Cat	Cow	Chicken	Fruit Fly	Chimpanzee	Human
Scientific Name	<i>Felis silvestris catus</i>	<i>Bos primigenius</i>	<i>Gallus gallus domesticus</i>	<i>Drosophila melanogaster</i>	Chimpanzee	<i>Homo sapiens</i>
Picture Representation						
Chromosome Count(2*n)	38	60	78	8	48	46
Size(GB)	2.33	2.53	0.99	0.139	2.87	2.93

Table 3: Chromosome Count and Size of Genome

b) Number of nodes created and merged in PTA tree

Initially, when PTA tree is generated; it creates nodes based on strings fed to it. The number of nodes created by PTA tree is independent of confidence range alpha used in our program. In the table below, we can see that over all the alphas, number of nodes created in PTA for a human is 9703.

After applying Alergia to PTA tree we obtain SFA with merged nodes. The number of nodes merged depends on alpha value. Lesser the value of alpha lesser the nodes that are merged. Higher the value of alpha higher the nodes are merged. However, after a certain value of alpha the SFA converges i.e. the algorithm stops performing merges.

As seen in the table below initial number of nodes in PTA tree for a human is 9703. When alpha is 0.1 it did not perform any merges, when alpha is 0.2 it merged 210 nodes, when alpha is 0.3 it

merged 4254 nodes which is more than 50%. Finally, when alpha reaches 1.9 the number of merges stop and becomes constant with 2027 merged nodes.

Alpha	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Number of Nodes Created in PTA	9703	9703	9703	9703	9703	9703	9703
Number of Nodes after Merge	9703	9493	5449	5175	2996	3217	2875

Table 4: Number of Node Merged for Alpha 0.2 to 0.7

Alpha	0.8	0.9	1.0	1.1	1.2	1.3
Number of Nodes Created in PTA	9703	9703	9703	9703	9703	9703
Number of Nodes after Merge	3217	2230	2467	2723	2720	2088

Table 5: Number of Node Merged for Alpha 0.8 to 1.3

Alpha	1.4	1.5	1.6	1.7	1.8	1.9
Number of Nodes Created in PTA	9703	9703	9703	9703	9703	9703
Number of Nodes after Merge	2074	2075	2071	2030	2028	2027

Table 6: Number of Node Merged for Alpha 1.4 to 1.9

Alpha	2.0
Number of Nodes Created in PTA	9703
Number of Nodes after Merge	2027

Table 7: Number of Node Merged for Alpha 2.0

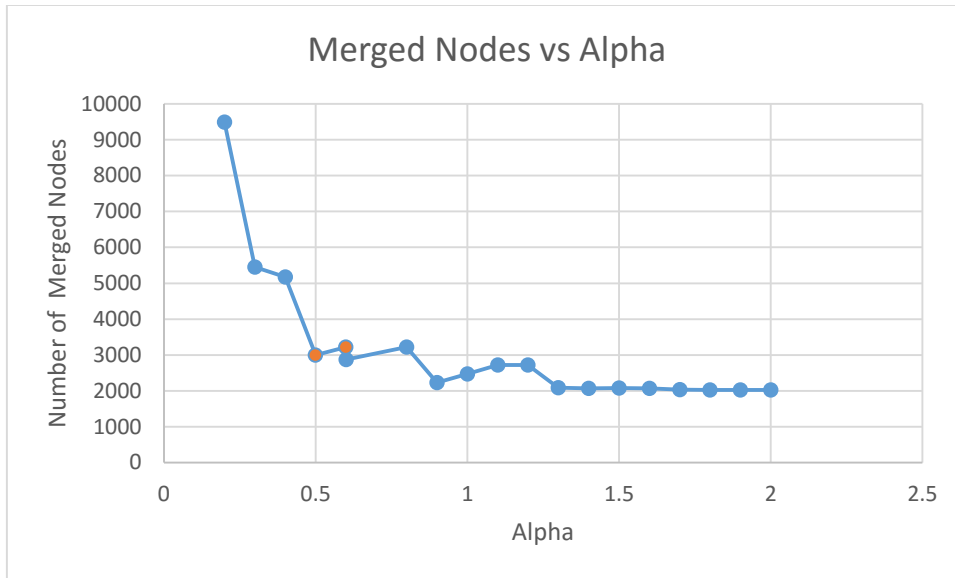


Figure 8: Merged Nodes vs Alpha, orange dot represents best result

By looking at the graph above, we can say that resultant SDFA obtained after applying Alergia is performing more merges as alpha increases from 0.1 to 1.0. After 1.0 it is becoming more stable and performing less and lesser merges with increasing value of alpha.

Number of Nodes merged affects percentage similarity of species: - The percentage similarity between different species is dependent on the number of nodes merged in a PTA tree. When SFA is strict, i.e. it has more number of nodes merged; it tends to reject more strings as compared to SFA which is lenient in nature i.e. SFA that rejects less number of strings. The results for this assumption is shown in Table 8.

In the table below, we trained PTA tree on human DNA over various alphas and compared with Cat, Cow, Chicken, Fruit Fly and Chimpanzee DNA.

Observation

- As alpha increases the percentage similarity decreases between various species.

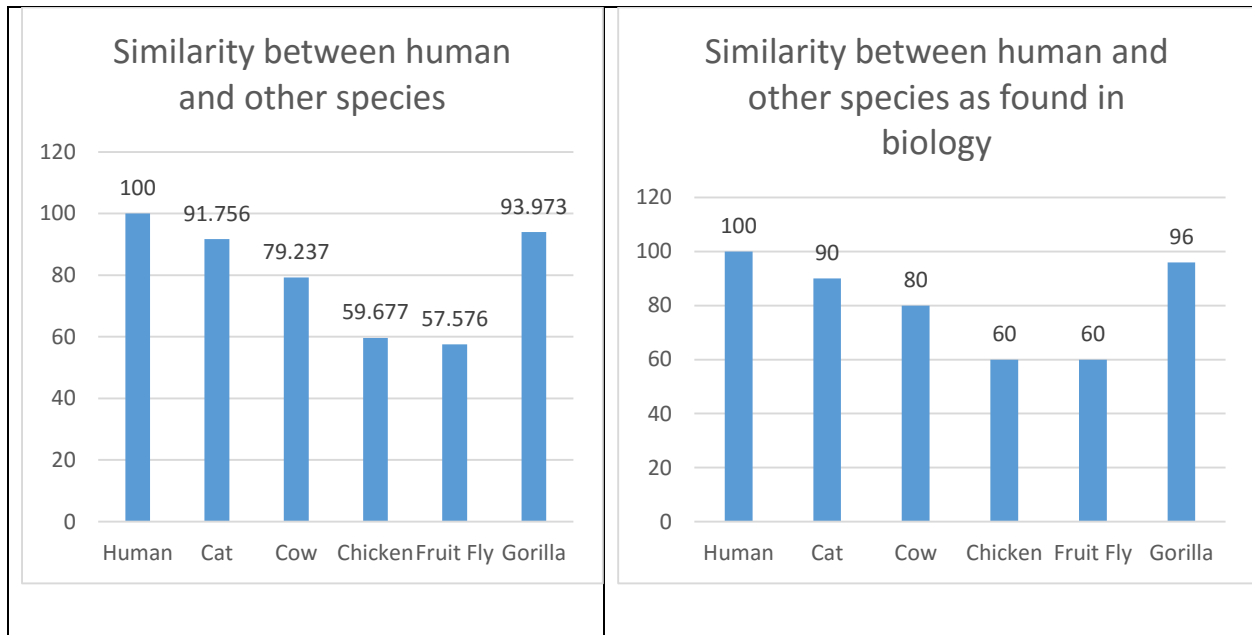
- As alpha becomes greater than equal to 1.7 the results are not showing any fluctuations, i.e. the SFA has stopped merging nodes and has converged.
- The values in blue rows represents percentage similarity found between human and Chimpanzee [18], Cattle [19], Cat [20], Fruit Fly [21] and Chicken [22] till date in various genome researches.
- Values in red shows the values obtained in our program that are close to values in blue, i.e. the genetic similarity found between species till date.
- In the table below the human similarity shows 100 %, which is due to the fact that we are testing same human on which we trained our SFA. Since both DNA's are same we are getting a 100 percent result however, no two human have same DNA, which will be shown in a later section of this report.

Alpha	Human	Cat	Cow	Chicken	Fruit Fly	Chimpanzee
0.1	100	100	100	100	100	100
0.2	100	99.821	99.718	100	100	99.726
0.3	100	91.756 90	91.384	87.903	96.97	93.973 96
0.4	100	92.832	90.113	86.29	100	92.329
0.5	100	79.298	79.237 80	74.194	93.939	85.479
0.6	100	75.269	76.695	69.355	57.576 60	78.356
0.7	100	72.76	68.644	61.29	51.515	75.342
0.8	100	68.996	66.949	63.71	63.636	70.685
0.9	100	70.609	65.395	55.645	51.515	65.479
1	100	67.563	64.689	59.677 60	39.394	63.562
1.1	100	61.111	61.299	61.29	36.364	60.274
1.2	100	62.903	60.903	56.452	36.364	60
1.3	100	63.978	59.463	55.645	27.272	60
1.4	100	64.158	59.322	55.645	27.273	59.726
1.5	100	62.366	59.746	54.032	24.242	59.452
1.6	100	63.262	59.605	53.226	24.242	59.178
1.7	100	61.29	58.757	54.032	24.242	60

1.8	100	61.29	58.757	54.032	24.242	60
1.9	100	61.29	58.757	54.032	24.242	60
2	100	61.29	58.757	54.032	24.242	60

Table 8: Result Summary

If we look at the graphs below, we can visualize the results in a better manner. The graphs show almost equivalent results.



In order to get an alpha value at which the results predicted by our program are more similar to results obtained in genome sequencing, we computed the difference in values predicted which are shown in the table below. By looking at the table, we can see that minimum difference is obtained when value of alpha is 0.5 or 0.6. At these values, the difference is only 5-6%. A graph is presented after the table that can help in better visualization of results.

Note: - This comparison is based on similarities found till date on various species. It can change as explained in section 6.

Alpha	Sum of Percentage Similarity found by our program	Sum of percentage similarity known in Biology	Percent Difference
0.2	599.265	486	23.3

0.3	561.986	486	15.63
0.4	561.564	486	15.54
0.5	512.147	486	5.38
0.6	457.251	486	5.91
0.6	429.551	486	11.61
0.8	433.976	486	10.7
0.9	408.643	486	15.91
1	394.885	486	18.74
1.1	380.338	486	21.74
1.2	376.622	486	22.5
1.3	366.358	486	24.61
1.4	366.124	486	24.66
1.5	359.838	486	25.95
1.6	359.513	486	26.02
1.7	358.321	486	26.27
1.8	358.321	486	26.27
1.9	358.321	486	26.27
2	358.321	486	26.27

Table 9: Difference in results

Orange color dots represent minimum error found over alpha 0.5 and 0.6.



Figure 9: Difference in results

5.4.Result Analysis for Cat with Cat, Cattle, Chicken, Fruit fly and Chimpanzee

We also compared Cat with other species to check whether it also shows the same behavior as shown in section 5.3. i.e. SFA stops merging after certain values of alpha or not.

As we can see in the table below Alergia is merging more nodes in the resultant SFA as alpha increase and after a certain point in this case at alpha 1.7 it stops merging. The behavior is same as seen above.

a) Number of nodes created and merged in PTA tree for Cat

Alpha	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Number of Nodes Created in PTA	8728	8728	8728	8728	8728	8728	8728
Number of Nodes after Merge	8725	8717	5516	5175	2395	2581	2449

Table 10: Number of nodes merged for Cat

Alpha	0.8	0.9	1.0	1.1	1.2	1.3
Number of Nodes Created in PTA	8728	8728	8728	8728	8728	8728
Number of Nodes after Merge	2167	2039	2299	2497	2614	2012

Alpha	1.4	1.5	1.6	1.7	1.8	1.9
Number of Nodes Created in PTA	8728	8728	8728	8728	8728	8728
Number of Nodes after Merge	2004	2002	1998	1955	1955	1955

Alpha	2.0
Number of Nodes Created in PTA	9703
Number of Nodes after Merge	1955

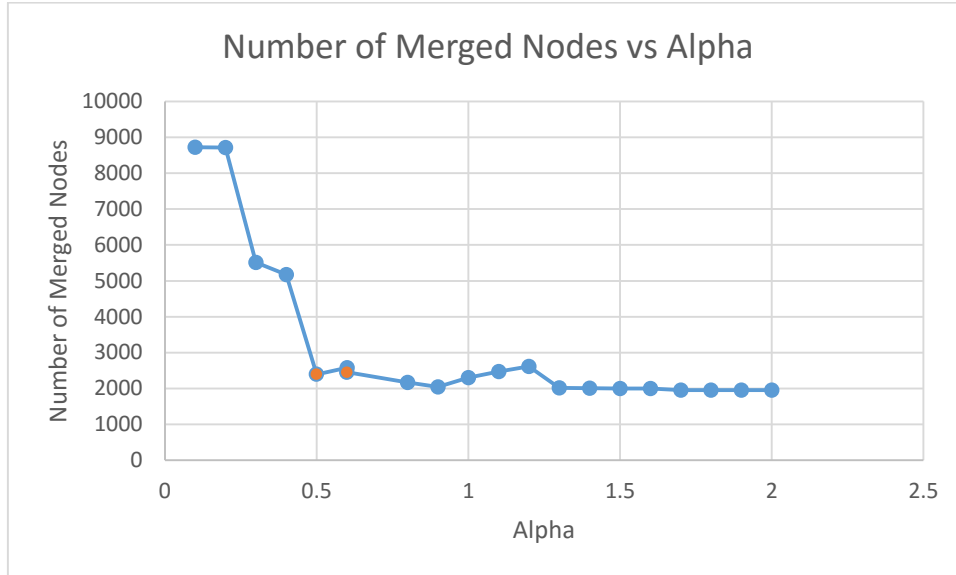


Figure 10: Number of nodes merged for CAT

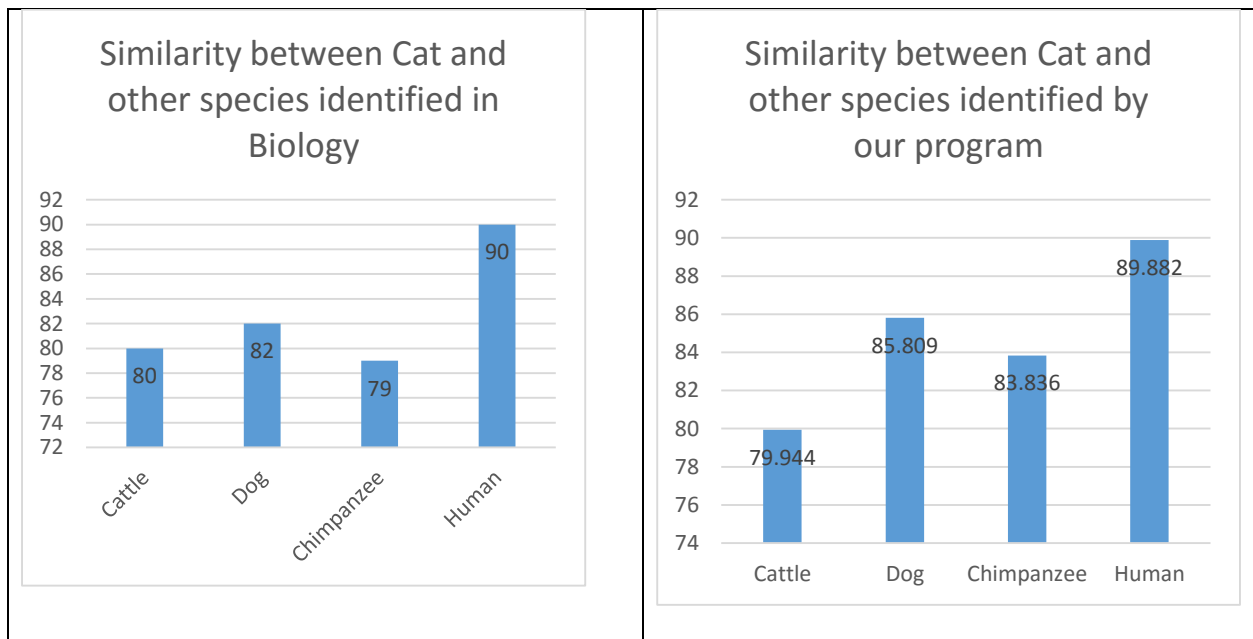
Below is a table that represents result obtained when program was trained on cat and tested against Cat, Cattle, Chicken, Fruit Fly, Chimpanzee and Human

Alpha	Cat	Cattle	Dog	Chimpanzee	Human
0.1	100	100	100	100	100
0.2	100	100	100	100	100
0.3	100	88.842	90.429	89.863	88.324
0.4	100	88.701	88.722	89.041	87.858
0.5	100	79.944 80	87.122	83.836 79	82.462
0.6	100	68.785	85.809 82	67.671	89.882 90
0.7	100	66.243	65.347	67.671	66.948
0.8	100	62.853	57.756	64.932	62.395
0.9	100	59.322	60.726	63.014	62.226
1	100	62.006	60.066	63.288	62.563
1.1	100	56.073	54.455	57.534	58.347
1.2	100	59.04	54.785	58.082	60.034
1.3	100	57.203	53.465	52.055	55.143
1.4	100	57.486	53.465	53.151	54.975
1.5	100	56.78	50.825	53.973	55.818
1.6	100	56.356	50.825	53.973	56.324
1.7	100	57.062	51.155	53.973	56.83

1.8	100	57.062	51.155	53.973	56.83
1.9	100	57.062	51.155	53.973	56.83
2	100	57.062	51.155	53.973	56.83

Table 11: Similarity between Cat and Cattle, Chicken, fruit Fly and Chimpanzee

As seen in the table above the difference in values known in biology (represented as blue) and identified by our program (represented as red) are quite similar. Again, most accurate values are found at alpha 0.5 and 0.6.



6. Similarity between different species can reduce with time

While comparing species in section 5 we noticed that when SFA becomes stable or in other words, it stops merging nodes the genetic similarity predicted by our program reaches a constant value.

As shown in the table below at alpha greater than equal to 1.7 the similarity between human and cat is shown as 61.29 % and it does not change after that. So we researched on finding if genetic similarity known till date are facts or are they are bound to change with time.

Alpha	Human	Cat	Cow	Chicken	Fruit Fly	Chimpanzee
1.7	100	61.29	58.757	54.032	24.242	60

We found that percentage similarity identified till date by Biologists can vary with time. Various factors that can lead to change in percentage similarity over time:-

- Complete genome sequence of many animals is still not sequenced. The percentage provided above is based on sequences found till date.
- The default standard for comparing DNA is Blast [2] that uses an empirical approach where it finds a few short matches between two sequences and then starts to look for more similar alignments locally, which means it does not cover the entire search space. It starts with the matching words of size seven to eleven bases in order to match a query with subject sequence. From there, it starts comparing the nearby sequence areas to find more matches. It only aligns locally, considers few gaps and substitutions. It represents results in the form of x or y which can be represented as x % similarity over y% sequence. It does not even cover whole sequence. Hence the results known till date are not based on complete DNA matching it is based on subsequence comparisons.

The figure below represents result found when a DNA sequence from mouse genome represented as Query 1 was passed to blast. Query 1 is compared against subject human. The result shows that Query 1 was found at position 3252 to position 3270 in human DNA. Based on it the result obtained is 97 o 67 i.e. 97% of mouse DNA matched with 67% of human DNA. The result is based on a small sequence of mouse DNA.

```

Query 1      GTACCTTGATTTCGTATTC 19
             |||
Sbjct 3252 GTACCTTGATTTCGTATTC 3270

```



```

Query 56      GACTCTACTACCTTTACCC 74
             |||
Sbjct 3475 GACTCTACTACCTTTACCC 3457

```

Figure 11: Blast comparison of mouse and human

- An example of such case is **Chimpanzee** [18]. Initially, it was believed that human and chimpanzee are 98.5–99.4 % similar. At that time whole human genome sequence was available, but whole chimpanzee genome was not. Thus, much of the past work was based on only a part of the total DNA. In the fall of 2005, in a special issue of *Nature* [17] devoted to chimpanzees, researchers described the draft sequence of the chimpanzee genome. Researchers that genetic similarity reduced to 4%. [18]. It might go down further in future, our program can actually help in predicting this difference. According to our program only 60% of human, chimpanzee DNA match when SFA is in stable state. It can help in reducing the difference from 96% to 60% by generating lists of rejected strings.

Here is an example for String rejected for Chimpanzee over Alpha 0.5

033311112121121424121111221211212124233141212111424241144131114211112111111313
121241141224111132221413241414111321211121141141421112221141111241113111413122
414143141413124414141314124

Where the string shown above is a protein, each number in the string represents its corresponding Amino Acid Group.

- In article “Initial Sequence and comparative analysis of Cat Genome” [21], cat genome was compared against 6 mammals and similarity percentages were predicted. Those percentages are used in our project. However, in conclusion section they have mentioned that there are some weaknesses of the approach being used to compare cat and other species which is *light coverage*.

In spite of the benefits derived from the comparative genomics-based genome annotation presented here, there are some notable weaknesses due to a light coverage. Among them are the following: (1) The assembled cat genome retains only 65% of the euchromatin genome sequence, leaving some 660,000 gaps between the contigs; (2) fewer than 58% of the genes have >50% of their gene feature sequence captured (based on cat-dog gene homologs); and (3) estimating the number, extent, and location of segmental duplications (which comprise 5% of the human genome) is difficult with low coverage since segmental duplication discovery depends on highly redundant genome coverage for accuracy (International Human Genome Sequencing Consortium 2001; Mouse Genome Sequencing Consortium 2002).

Figure 12: Conclusions and applications of the feline genome sequence

The result obtained is over 65% of cat genome, the rest of the genome is gaps which are represented by “N” in DNA sequence. Hence, in future if the rest of DNA sequence is identified the similarity will go down even further.

- **Genome Evolution** – Another factor that can lead to changes in DNA similarity is Genome Evolution. Genome Evolution [22] is a phenomena that leads to changes in

the size or structure of DNA over time. Due to evolution, we need techniques that can match whole DNA sequence and save time in matching small samples in order to predict similarity.

Whenever a new DNA sequence is found, our program can train on it and predict the changes in percentage similarity. We can also train on old DNA and test new DNA to generate the strings that were rejected. The rejected strings can help in better understanding of mutations occurring over time.

7. Human Genetic Variation

Human Genetic variation [23] is the variation in DNA found among humans. This variation leads to polymorphism, however, not all of the genes in DNA polymorphic, many of them are fixed. Due to this even though 2 humans have highly similar DNA approx. 95%, no two humans have the same DNA. This variation in DNA can help in identifying a bunch of information. For example

- Trace origin of races
- Group races in clusters
- Identify diseases common in a race.

In this section we will look at a popular theory and try to see if our program supports it or not.

Out of Africa Theory [24] - Archaic human evolved into modern human in Africa. The theory states that early human races lived in Africa. A quotation by Charles Darwin is shown below:-

In each great region of the world the living [mammals](#) are closely related to the extinct species of the same region. It is, therefore, probable that Africa was formerly inhabited by extinct apes closely allied to the [gorilla](#) and [chimpanzee](#); and as these two species are now man's nearest allies, it is somewhat more probable that our early progenitors lived on the African continent than elsewhere. But it is useless to speculate on this subject, for an ape nearly as large as a man, namely the [Dryopithecus](#) of Lartet, which was closely allied to the anthropomorphous [Hylobates](#), existed in Europe during the [Upper Miocene](#) period; and since so remote a period the earth has certainly undergone many great revolutions, and there has been ample time for migration on the largest scale.

— Charles Darwin, *Descent of Man*^[27]

In this theory mtDB- Human Mitochondrial DNA was compared to identify the origin of humans. The result obtained was that most of the humans have originated in Africa. They later migrated and developed individual differences. In order to check this theory, we compared Indian Human Mitochondrial DNA with African Human Mitochondrial DNA.

We found out that Indians are 97.917% similar to African DNA. We also compared the Indian DNA with Australian and French DNA. We found out that Indians are 95.718% similar to French DNA

Still Indians are more similar to African DNA than French DNA, which supports “**Out of Africa Theory**”.

S.no	Alpha	Indian	Aborginal Malaya(Asia)	Bamileke(Africa)
1	0.1	100	52.632	97.917
2	0.2	100	52.632	97.917
3	0.3	100	52.632	97.917
4	0.4	100	60.526	97.917
5	0.5	100	39.474	95.833
6	0.6	100	47.368	97.917
7	0.7	100	47.368	95.833
8	0.8	100	44.737	97.917
9	0.9	100	21.053	95.833
10	1	100	15.789	95.833

Table 12: Similarity between Indian and African, Asian DNA

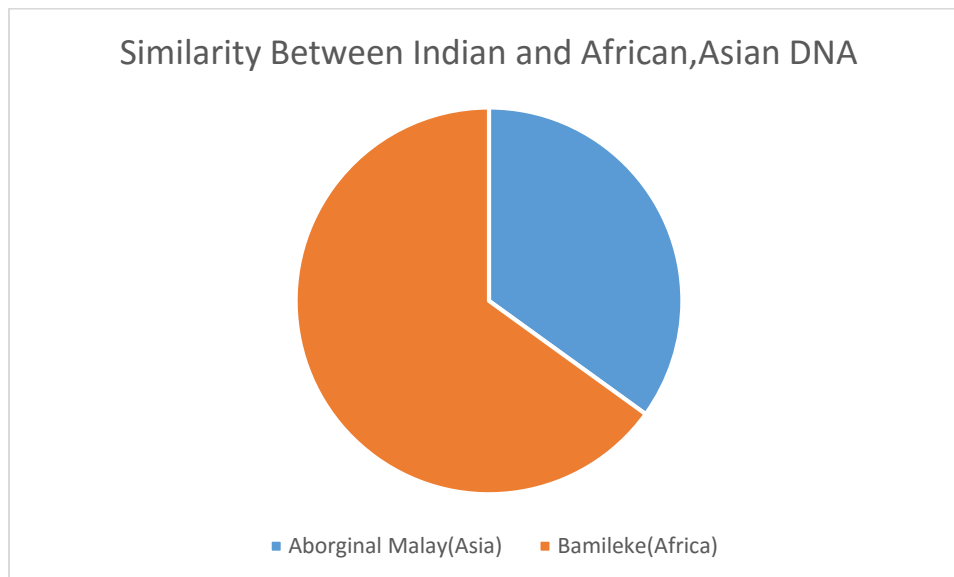


Figure 13: Similarities between Indian and African, Asian

We also tried to compare Indian DNA with Australian DNA and French DNA and received following results:

S.no	Alpha	Indian	Australia	French
1	0.1	100	69.231	95.918
2	0.2	100	69.231	95.618
3	0.3	100	69.231	94.918
4	0.4	100	71.795	94.718
5	0.5	100	53.846	95.918
6	0.6	100	46.154	95.918
7	0.7	100	48.718	95.918
8	0.8	100	41.026	95.918
9	0.9	100	33.333	95.918
10	1	100	25.641	95.918

Table 13: Similarities of Indian with French and Australians

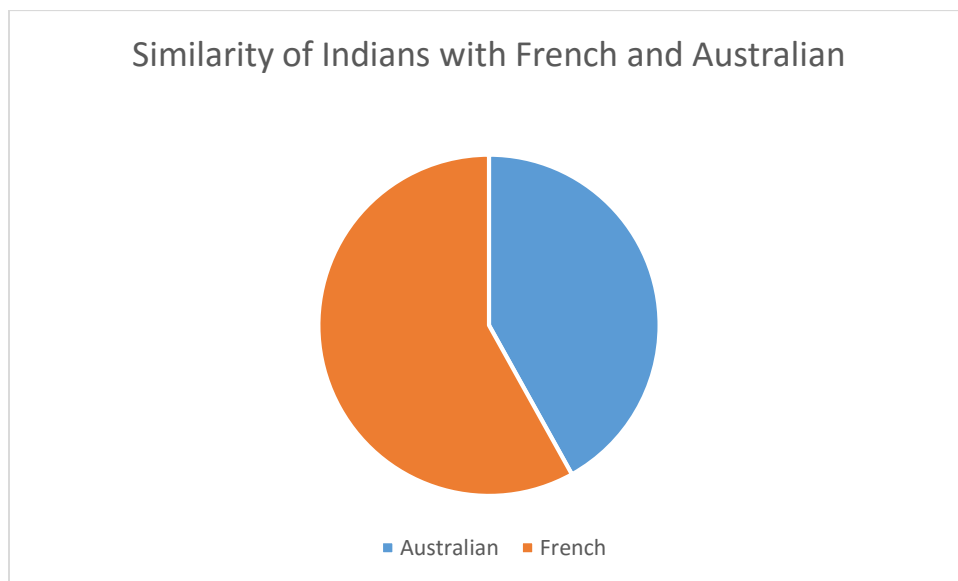


Figure 14: Similarity between Indians and French, Australian

We also trained on French DNA and compared against Indian, South American, and African DNA. We again found same results French are more similar to African DNA than any other race. Comparison of French DNA with Indian, African, and Australian is shown in the table below. At alpha 0.5 French are 97.91% similar to African, 46.154% similar to Australian and 46.154% similar to Indian.

S.no	Alpha	African	Australian	French	Indian
1	0.1	100	97.436	100	97.436
2	0.2	100	97.436	100	97.436
3	0.3	100	79.487	100	76.923
4	0.4	100	69.231	100	66.667
5	0.5	97.917	46.154	100	46.154
6	0.6	97.917	30.769	100	30.769
7	0.7	97.917	35.897	100	38.462
8	0.8	100	53.846	100	53.846
9	0.9	97.917	48.718	100	48.718
10	1	97.917	30.769	100	30.769

Table 14: Comparison of French DNA with Indian, African, South American

By looking at the table above, we can see that French are more similar to African than any other DNA. We tested for Indian and French against various species and found that in both cases, they are more similar to African DNA than any other DNA which supports “out of Africa” theory.

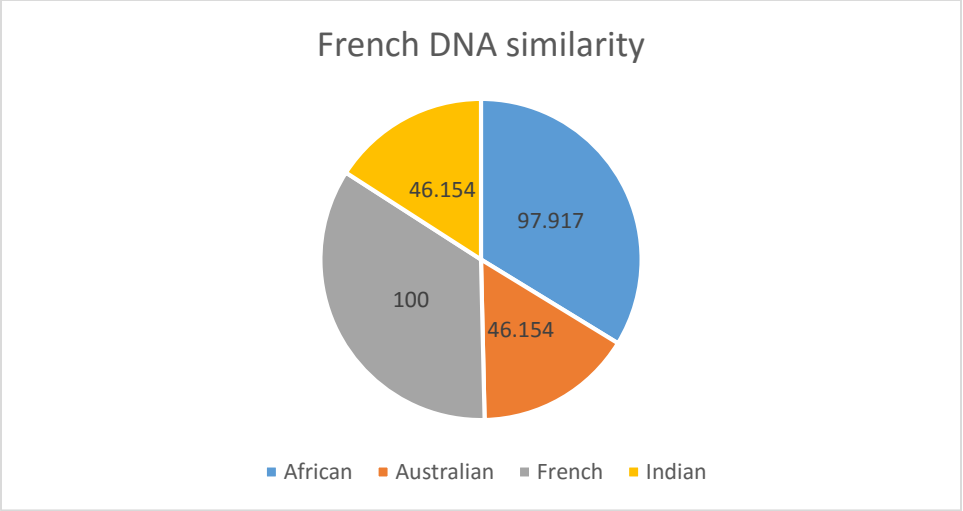


Figure 15: French DNA similarity with Australian, Indian and Africa

8. Clustering of Races

In this section we used DBSCAN (Density-based spatial clustering of applications with noise) clustering algorithm to group different human DNA into clusters. For this purpose we used data from mtDB – Human Mitochondrial Genome database [25] as this same data was used to prove “out of Africa” theory.

This data is grouped into 10 major geographic regions based on population origin of the human donor which are:-

1. Africa
2. America (North)
3. Australia – Covers Australia, Aborigine DNA's
4. Europe
5. Melanesia
6. Micronesia
7. Middle East
8. Polynesia
9. South Asia
10. Great Apes

Large sets from the same population are available as batches of individual files. For example Africa region covers DNA from following races.

Algeria, Bamileke, Biaka Pygmy, Canary Island, Effik, Egypt, Ethiopia, Ewondo, Fulpe, Hausa, Ibo, Khwe, Kikuyu, Lisongo, Mandeka, Morocco

Total number of Mitochondrial Sequences – 1942

We generated SFA for each species in the database provided and compared against all other species to generate a percentage similarity.

Below picture represents a sample obtained when SFA comparison was done among all the human DNA's obtained from Australia Region.

```
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (1):10.0
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (10):1.0416666666666665
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (11):1.5789473684210527
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (12):1.0416666666666665
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (13):0.7692307692307693
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (14):0.7894736842105263
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (15):0.6521739130434783
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (16):0.8108108108108109
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (17):0.7692307692307693
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (2):0.6521739130434783
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (3):9.473684210526317
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (4):1.0638297872340425
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (5):1.0416666666666665
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (6):1.0416666666666665
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (7):1.0204081632653061
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (8):1.0416666666666665
Ingman_Australian_AY289051 (1),Ingman_Australian_AY289051 (9):6.0606060606060606
Ingman_Australian_AY289051 (1),Italian(AF346988):0.7692307692307693
Ingman_Australian_AY289051 (1),Pellekaan_Australian_DQ404442 (1):0.7692307692307693
Ingman_Australian_AY289051 (1),Pellekaan_Australian_DQ404442 (2):0.7692307692307693
Ingman_Australian_AY289051 (1),Pellekaan_Australian_DQ404442 (3):0.7692307692307693
Ingman_Australian_AY289051 (1),Pellekaan_Australian_DQ404442 (4):1.0416666666666665
Ingman_Australian_AY289051 (1),Pellekaan_Australian_DQ404442 (5):0.7692307692307693
Ingman_Australian_AY289051 (1),Pellekaan_Australian_DQ404442 (6):0.5263157894736843
Ingman_Australian_AY289051 (1),Pellekaan_Australian_DQ404442 (7):1.0204081632653061
Ingman_Australian_AY289051 (1),Pellekaan_Australian_DQ404442 (8):1.0416666666666665
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (1):0.5405405405405406
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (10):10.0
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (11):0.7894736842105263
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (12):8.958333333333332
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (13):0.7692307692307693
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (14):0.7894736842105263
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (15):3.695652173913044
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (16):5.405405405405405
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (17):0.7692307692307693
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (2):3.695652173913044
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (3):0.5263157894736843
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (4):9.574468085106384
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (5):9.583333333333332
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (6):9.583333333333332
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (7):9.387755102040817
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (8):9.583333333333332
Ingman_Australian_AY289051 (10),Ingman_Australian_AY289051 (9):0.9090909090909092
Ingman_Australian_AY289051 (10),Italian(AF346988):0.7692307692307693
```

Figure 16: Australian DNA's compared against Australian and Italian DNA (1)

Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (11):10.0
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (12):1.0416666666666665
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (13):8.461538461538462
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (14):8.421052631578949
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (15):5.434782608695652
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (16):4.864864864864865
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (17):8.717948717948719
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (2):5.6521739130434785
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (3):1.5789473684210527
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (4):1.0638297872340425
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (5):1.0416666666666665
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (6):1.0416666666666665
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (7):1.0204081632653061
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (8):1.0416666666666665
 Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051 (9):3.939393939393939
 Ingman_Australian_AY289051 (11),Italian(AF346988):8.461538461538462
 Ingman_Australian_AY289051 (11),Pellekaan_Australian_DQ404442 (1):7.948717948717949
 Ingman_Australian_AY289051 (11),Pellekaan_Australian_DQ404442 (2):8.205128205128206
 Ingman_Australian_AY289051 (11),Pellekaan_Australian_DQ404442 (3):8.205128205128206
 Ingman_Australian_AY289051 (11),Pellekaan_Australian_DQ404442 (4):1.0416666666666665
 Ingman_Australian_AY289051 (11),Pellekaan_Australian_DQ404442 (5):7.692307692307692
 Ingman_Australian_AY289051 (11),Pellekaan_Australian_DQ404442 (6):8.157894736842106
 Ingman_Australian_AY289051 (11),Pellekaan_Australian_DQ404442 (7):1.0204081632653061
 Ingman_Australian_AY289051 (11),Pellekaan_Australian_DQ404442 (8):1.0416666666666665
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (1):0.5405405405405406
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (10):8.958333333333332
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (11):0.7894736842105263
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (12):10.0
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (13):1.794871794871795
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (14):1.842105263157895
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (15):4.565217391304348
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (16):4.594594594594595
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (17):1.794871794871795
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (2):4.3478260869565215
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (3):0.5263157894736843
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (4):8.936170212765957
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (5):8.958333333333332
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (6):8.958333333333332
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (7):8.775510204081632
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (8):8.958333333333332
 Ingman_Australian_AY289051 (12),Ingman_Australian_AY289051 (9):0.9090909090909092
 Ingman_Australian_AY289051 (12),Italian(AF346988):1.794871794871795
 Ingman_Australian_AY289051 (12),Pellekaan_Australian_DQ404442 (1):1.794871794871795
 Ingman_Australian_AY289051 (12),Pellekaan_Australian_DQ404442 (2):1.794871794871795

Figure 17: Australian DNA's compared against Australian and Italian DNA (2)

In above picture the numerical values presented are distances between the species on which SFA was generated and the species tested.

8.1.DBSCAN algorithm for clustering

Number of DNA's present in Dataset	1942
Total number of combinations generated when each species was trained and compared against all other species	3765539
Time taken to generate output	1 hr 17 mn

As shown in figure 16, 17 each species was trained and compared against each other. The resultant data was fed to DBSCAN algorithm to check clusters formed.

DBSCAN [26] requires two parameters: ϵ (eps) and the minimum number of points required to form a dense region (minPts). It starts with an arbitrary starting point that has not been visited. This point's ϵ -neighborhood is retrieved, and if it contains sufficiently many points, a cluster is started. Otherwise, the point is labeled as noise. Note that this point might later be found in a sufficiently sized ϵ -environment of a different point and hence be made part of a cluster.

If a point is found to be a dense part of a cluster, its ϵ -neighborhood is also part of that cluster. Hence, all points that are found within the ϵ -neighborhood are added, as is their own ϵ -neighborhood when they are also dense. This process continues until the density-connected cluster is completely found. Then, a new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.

```

''' Simple clustering script based on dbScan '''

import numpy as np
from sklearn.cluster import DBSCAN
import random

__author__ = 'ssaroha'
cluster = []
nameToIndex = {}
indexToName = {}

with open('CountriesName.txt') as f:
    countries = f.read().splitlines()
    print (countries)
    print (countries[0])
    print (len(countries))
    ''' List of sample ethnicities '''
    cluster = countries
    indexToName = dict(enumerate(countries))
    i = 0
    for name in countries:
        nameToIndex[name]=i
        i = i + 1

    #nameToIndex = dict((value, key) for key, value in indexToName.iteritems())
    print (nameToIndex)
    print (len(nameToIndex))
    print (indexToName)
    print (len(indexToName))

data = np.zeros((len(nameToIndex),len(nameToIndex)));

with open('Results.txt') as f:
    reader = f.read().splitlines()
    for row in reader:
        row = row.replace(':', ',')
        values = row.split(",")
        if len(values)== 3 and values[0] in nameToIndex and values[1] in nameToIndex:
            i = nameToIndex[values[0]]
            j = nameToIndex[values[1]]
            score = float(values[2])
            if data[i,j] == 0.0:
                data[i,j] = score
data = np.zeros((len(cluster),len(cluster)));

''' Computer the distance between each ethnicity

```

```

    for example : indian , french : 7
                  indian , italian : 4
                  .
                  .
                  .
                  american, japanese : 7

'''
''' Printing the array generated randomly '''
print(">> Array generated for all ethnicities in a 5 x 5 matrix :")
# print data
''' Initialize DBSCAN
    Consider epsilon as the radius of the circle
    which encircles the points on the graph to form a cluster '''
clust = DBSCAN(eps=1,min_samples=10,metric="precomputed");

''' Fit the matrix to dbScan '''
clust.fit(data)

''' If you have done the above steps correctly,
    then the below code should work

    Below we are just printing the identified clusters '''

ethnicities = cluster

preds = clust.labels_
clabels = np.unique(preds)
core_samples_mask = np.zeros_like(preds, dtype=bool)
core_samples_mask[clust.core_sample_indices_] = True

# Number of clusters in labels, ignoring noise if present.
n_clusters_ = len(set(preds)) - (1 if -1 in preds else 0)
print('Estimated number of clusters: %d' % n_clusters_)

for i in range(clabels.shape[0]):
    if clabels[i] < 0:
        continue
    cmem_ids = np.where(preds==clabels[i])[0]
    cmembers = []

```

```

count = 0
for cmem_id in cmem_ids:
    cmembers.append(ethnicities[cmem_id])
    count += 1
clusteritems = ",".join(cmembers)
# print ("Clustered: " + clusteritems)
print ("Clustered :", count)
# Plot result
import matplotlib.pyplot as plt

# Black removed and is used for noise instead.
unique_labels = set(preds)
colors = plt.cm.Spectral(np.linspace(0, 1, len(unique_labels)))
for k, col in zip(unique_labels, colors):
    if k == -1:
        # Black used for noise.
        col = 'k'

    class_member_mask = (preds == k)

    xy = data[class_member_mask & core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=col,
             markeredgecolor='k', markersize=12)

    xy = data[class_member_mask & ~core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=col,
             markeredgecolor='k', markersize=6)

plt.title('Estimated number of clusters: %d' % n_clusters_)
plt.show()

```

The figure above represents DBSCAN algorithm used for our project. In the sample we are generating clusters on random values. Following the idea above us generated clusters on our output.

8.2.Clusters obtained for Epsilon = 0.2 and Number of Sample = 1

9. Estimated number of clusters: 1913
10. Clustered : 4
11. Clustered : 1
12. Clustered : 1
13. Clustered : 1
14. Clustered : 1
15. Clustered : 1
16. Clustered : 1
17. Clustered : 1
18. Clustered : 1
19. Clustered : 1
20. Clustered : 1
21. Clustered : 1
22. Clustered : 1

23. Clustered : 1
24. Clustered : 1
25. Clustered : 1
26. Clustered : 1
27. Clustered : 1
28. Clustered : 1
29. Clustered : 1
30. Clustered : 1
31. Clustered : 1
32. Clustered : 1
33. Clustered : 1
34. Clustered : 1
35. Clustered : 1
36. Clustered : 1
37. Clustered : 1
38. Clustered : 1
39. Clustered : 1
40. Clustered : 1
41. Clustered : 1
42. Clustered : 1
43. Clustered : 1
44. Clustered : 1
45. Clustered : 1
46. Clustered : 1
47. Clustered : 1
48. Clustered : 1
49. Clustered : 1
50. Clustered : 1

8.3.Clusters obtained for Epsilon = 1 and Number of Samples = 1

All 3765539 data points were passed to the algorithm to generate clusters. In total 36 clusters were generated for Epsilon = 1 and Number of Samples = 1

Below table represents the count of DNA's grouped in Each Cluster

Table 15: Cluster obtained for Epsilon = 1 and Number of Samples = 1

Estimated number of clusters: 36
Clustered : 1907
Clustered : 1
Clustered : 1
Clustered : 1
Clustered : 1
Clustered : 1

5779,AY195780,AY195782,AY195783,AY195784,AY195785,AY195786,AY195787,AY195788,AY195789,AY195790,AY195791,AY195792,AY275527,AY275528,AY275529,AY275530,AY275531,AY275532,AY275533,AY275534,AY275535,AY275536,AY275537,AY289076,AY289077,AY289078,AY289079,AY289080,AY289081,AY289082,AY289083,AY289084,AY289085,AY289086,AY289087,AY289088,AY289089,AY289090,AY289091,AY289092,AY495090,AY495091,AY495092,AY495093,AY495094,AY495095,AY495096,AY495097,AY495098,AY495099,AY495100,AY495101,AY495102,AY495103,AY495104,AY495105,AY495106,AY495107,AY495108,AY495109,AY495110,AY495111,AY495112,AY495113,AY495114,AY495115,AY495116,AY495117,AY495118,AY495119,AY495120,AY495121,AY495122,AY495123,AY495124,AY495125,AY495126,AY495127,AY495128,AY495129,AY495130,AY495131,AY495132,AY495133,AY495134,AY495135,AY495136,AY495137,AY495138,AY495139,AY495140,AY495141,AY495142,AY495143,AY495144,AY495145,AY495146,AY495147,AY495148,AY495149,AY495150,AY495151,AY495152,AY495153,AY495154,AY495155,AY495156,AY495157,AY495158,AY495159,AY495160,AY495161,AY495162,AY495163,AY495164,AY495165,AY495166,AY495167,AY495168,AY495169,AY495170,AY495171,AY495172,AY495173,AY495174,AY495175,AY495176,AY495177,AY495178,AY495179,AY495180,AY495181,AY495182,AY495183,AY495184,AY495185,AY495186,AY495187,AY495188,AY495189,AY495190,AY495191,AY495192,AY495193,AY495194,AY495195,AY495196,AY495197,AY495198,AY495199,AY495200,AY495201,AY495202,AY495203,AY495204,AY495205,AY495206,AY495207,AY495208,AY495209,AY495210,AY495211,AY495212,AY495213,AY495214,AY495215,AY495216,AY495217,AY495218,AY495219,AY495220,AY495221,AY495222,AY495223,AY495224,AY495225,AY495226,AY495227,AY495228,AY495229,AY495230,AY495231,AY495232,AY495233,AY495234,AY495235,AY495236,AY495237,AY495238,AY495239,AY495240,AY495241,AY495242,AY495243,AY495244,AY495245,AY495246,AY495247,AY495248,AY495249,AY495250,AY495251,AY495252,AY495253,AY495254,AY495255,AY495256,AY495257,AY495258,AY495259,AY495260,AY495261,AY495262,AY495263,AY495264,AY495265,AY495266,AY495267,AY495268,AY495269,AY495270,AY495271,AY495272,AY495273,AY495274,AY495275,AY495276,AY495277,AY495278,AY495279,AY495280,AY495281,AY495282,AY495283,AY495284,AY495285,AY495286,AY495287,AY495288,AY495289,AY495290,AY495291,AY495292,AY495293,AY495294,AY495295,AY495296,AY495297,AY495298,AY495299,AY495300,AY495301,AY495302,AY495303,AY495304,AY495305,AY495306,AY495307,AY495308,AY495309,AY495310,AY495311,AY495312,AY495313,AY495314,AY495315,AY495316,AY495317,AY495318,AY495319,AY495320,AY495321,AY495322,AY495323,AY495324,AY495325,AY495326,AY495327,AY495328,AY495329,AY495330,AY713976,AY713977,AY713978,AY713979,AY713980,AY713981,AY713982,AY713983,AY713984,AY713985,AY713986,AY713987,AY713988,AY713989,AY713990,AY713991,AY713992,AY713993,AY713994,AY713995,AY713996,AY713997,AY713998,AY713999,AY714000,AY714001,AY714002,AY714003,AY714004,AY714005,AY714006,AY714007,AY714008,AY714009,AY714010,AY714011,AY714012,AY714013,AY714014,AY714015,AY714016,AY714017,AY714018,AY714019,AY714020,AY714021,AY714022,AY714023,AY714024,AY714025,AY714026,AY714027,AY714028,AY714029,AY714030,AY714031,AY714032,AY714033,AY714034,AY714035,AY714036,AY714037,AY714038,AY714039,AY714040,AY714041,AY714042,AY714043,AY714044,AY714045,AY714046,AY714047,AY714048,AY714049,AY714050,AY738940,AY738941,AY738942,AY738943,AY738944,AY738945,AY738946,AY738947,AY738948,AY738949,AY738950,AY738951,

AY738952,AY738953,AY738954,AY738955,AY738956,AY738957,AY738958,AY738959,
 AY738960,AY738961,AY738962,AY738963,AY738964,AY738965,AY738966,AY738967,
 AY738968,AY738969,AY738970,AY738971,AY738972,AY738973,AY738974,AY738975,
 AY738976,AY738977,AY738978,AY738979,AY738980,AY738981,AY738982,AY738983,
 AY738984,AY738985,AY738986,AY738987,AY738988,AY738989,AY738990,AY738991,
 AY738992,AY738993,AY738994,AY738995,AY738996,AY738997,AY738998,AY738999,
 AY739000,AY739001,AY882379,AY882380,AY882381,AY882382,AY882383,AY882384,
 AY882385,AY882386,AY882387,AY882388,AY882389,AY882390,AY882391,AY882392,
 AY882393,AY882394,AY882395,AY882396,AY882397,AY882398,AY882399,AY882400,
 AY882401,AY882402,AY882403,AY882404,AY882405,AY882406,AY882407,AY882408,
 AY882409,AY882410,AY882411,AY882412,AY882413,AY882414,AY882415,AY882416,
 AY882417,AY950286,AY950287,AY950288,AY950289,AY950290,AY950291,AY950292,
 AY950293,AY950294,AY950295,AY950296,AY950297,AY950298,AY950299,AY950300,
 AY956412,AY956413,AY956414,AY963572,AY963573,AY963574,AY963575,AY963576,
 AY963577,AY963578,AY963579,AY963580,AY963581,AY963582,AY963583,AY963584,
 AY963585,AY963586,Aborginal Malay
 AY963578,African,Andalusia(AF382011),Aust(AF346963),Aust(AF346964),Aust(AF346965
),Bamileke(AF346967),Berber(AF381989),Berber(AF381990),Biaka(AF346968),Biaka(AF34
 6969),Buriat(AF346970),Buriat(AY519484),Canary(AF381982),Canary(AF382009),Canary(
 AF382010),Chinese(AF346972),Chinese(AF346973),Chukchi(AF346971),Cook(AY289068),
 Cook(AY289069),DQ112686,DQ112687,DQ112688,DQ112689,DQ112690,DQ112691,DQ1
 12692,DQ112693,DQ112694,DQ112695,DQ112696,DQ112697,DQ112698,DQ112699,DQ11
 2700,DQ112701,DQ112702,DQ112703,DQ112704,DQ112705,DQ112706,DQ112707,DQ112
 708,DQ112709,DQ112710,DQ112711,DQ112712,DQ112713,DQ112714,DQ112715,DQ1127
 16,DQ112717,DQ112718,DQ112719,DQ112720,DQ112721,DQ112722,DQ112723,DQ11272
 4,DQ112725,DQ112726,DQ112727,DQ112728,DQ112729,DQ112730,DQ112731,DQ112732
 ,DQ112733,DQ112734,DQ112735,DQ112736,DQ112737,DQ112738,DQ112739,DQ112740,
 DQ112741,DQ112742,DQ112743,DQ112744,DQ112745,DQ112746,DQ112747,DQ112748,
 DQ112749,DQ112756,DQ112757,DQ112758,DQ112759,DQ112765,DQ112777,DQ112778,
 DQ112782,DQ112792,DQ112794,DQ112796,DQ112829,DQ112830,DQ112844,DQ112845,
 DQ112847,DQ112848,DQ112849,DQ112850,DQ112851,DQ112852,DQ112853,DQ112854,
 DQ112855,DQ112856,DQ112857,DQ112883,DQ112884,DQ112885,DQ112886,DQ112887,
 DQ112895,DQ112896,DQ112897,DQ112898,DQ112899,DQ112900,DQ112901,DQ112902,
 DQ112903,DQ112904,DQ112905,DQ112906,DQ112907,DQ112908,DQ112909,DQ112910,
 DQ112911,DQ112912,DQ112913,DQ112914,DQ112915,DQ112916,DQ112917,DQ112918,
 DQ112919,DQ112920,DQ112921,DQ112922,DQ112923,DQ112924,DQ112925,DQ112926,
 DQ112932,DQ112933,DQ112934,DQ112949,DQ112956,DQ112957,DQ112958,DQ112959,
 DQ112960,DQ112961,DQ112962,DQ137398,DQ137399,DQ137400,DQ137401,DQ137402,
 DQ137403,DQ137404,DQ137405,DQ137407,DQ137408,DQ137409,DQ137410,DQ137411,
 DQ246811,DQ246812,DQ246813,DQ246814,DQ246815,DQ246816,DQ246817,DQ246818,
 DQ246819,DQ246820,DQ246821,DQ246822,DQ246823,DQ246824,DQ246825,DQ246826,
 DQ246827,DQ246828,DQ246829,DQ246830,DQ246831,DQ246832,DQ246833,DQ301789,
 DQ301790,DQ301791,DQ301792,DQ301793,DQ301794,DQ301795,DQ301796,DQ301797,
 DQ301798,DQ301799,DQ301800,DQ301801,DQ301802,DQ301803,DQ301804,DQ301805,
 DQ301806,DQ301807,DQ301808,DQ301809,DQ301810,DQ301811,DQ301812,DQ301813,
 DQ301814,DQ301815,DQ301816,DQ301817,DQ301818,DQ372870,DQ372871,DQ372872,

DQ372873,DQ372874,DQ372875,DQ372876,DQ372877,DQ372878,DQ372879,DQ372880,
 DQ372881,DQ372882,DQ372883,DQ372884,DQ372885,DQ372886,DQ523619,DQ523620,
 DQ523621,DQ523622,DQ523623,DQ523624,DQ523625,DQ523626,DQ523627,DQ523628,
 DQ523629,DQ523630,DQ523631,DQ523632,DQ523633,DQ523634,DQ523635,DQ523636,
 DQ523637,DQ523638,DQ523639,DQ523640,DQ523641,DQ523642,DQ523643,DQ523644,
 DQ523645,DQ523646,DQ523647,DQ523648,DQ523649,DQ523650,DQ523651,DQ523652,
 DQ523653,DQ523654,DQ523655,DQ523656,DQ523657,DQ523658,DQ523659,DQ523660,
 DQ523661,DQ523662,DQ523663,DQ523664,DQ523665,DQ523666,DQ523667,DQ523668,
 DQ523669,DQ523670,DQ523671,DQ523672,DQ523673,DQ523674,DQ523675,DQ523676,
 DQ523677,DQ523678,DQ523679,DQ523680,DQ523681,DQ902708,DQ902709,DQ902710,
 DQ902711,DQ981465,DQ981466,DQ981467,DQ981468,DQ981469,DQ981470,DQ981471,
 DQ981472,DQ981474,DQ981475,Dutch(AF346975),EF060313,EF060314,EF060315,EF060
 316,EF060317,EF060318,EF060319,EF060320,EF060321,EF060322,EF060323,EF060324,E
 F060325,EF060326,EF060327,EF060328,EF060329,EF060330,EF060331,EF060332,EF0603
 33,EF060334,EF060335,EF060336,EF060337,EF060338,EF060339,EF060340,EF060341,EF
 060342,EF060343,EF060344,EF060345,EF060346,EF060347,EF060348,EF060349,EF06035
 0,EF060351,EF060352,EF060353,EF060354,EF060355,EF060356,EF060357,EF060358,EF0
 60359,EF060360,EF060361,EF060362,EF064317,EF064318,EF064319,EF064320,EF064321
 ,EF064322,EF064323,EF064324,EF064325,EF064326,EF064327,EF064329,EF064330,EF06
 4331,EF064332,EF064333,EF064334,EF064335,EF064336,EF064337,EF064338,EF064339,
 EF064340,EF064341,EF064342,EF064343,EF064344,Effik(AF346976),Effik(AF346977),En
 glish(AF346978),Evenki(AF346979),Evenki(AY519485),Ewondo(AF346980),Filipino(AF38
 2012),Filipino(AY289070),French(AF346981),Georgian(AF346982),German(AF346983),Gu
 arani(AF346984),Hausa(AF346985),HernamtDNA12 (1),HernamtDNA12
 (10),HernamtDNA12 (11),HernamtDNA12 (12),HernamtDNA12 (13),HernamtDNA12
 (14),HernamtDNA12 (15),HernamtDNA12 (16),HernamtDNA12 (17),HernamtDNA12
 (18),HernamtDNA12 (19),HernamtDNA12 (2),HernamtDNA12 (20),HernamtDNA12
 (21),HernamtDNA12 (22),HernamtDNA12 (23),HernamtDNA12 (24),HernamtDNA12
 (25),HernamtDNA12 (26),HernamtDNA12 (27),HernamtDNA12 (28),HernamtDNA12
 (29),HernamtDNA12 (3),HernamtDNA12 (30),HernamtDNA12 (31),HernamtDNA12
 (32),HernamtDNA12 (33),HernamtDNA12 (34),HernamtDNA12 (35),HernamtDNA12
 (36),HernamtDNA12 (37),HernamtDNA12 (38),HernamtDNA12 (39),HernamtDNA12
 (4),HernamtDNA12 (40),HernamtDNA12 (41),HernamtDNA12 (42),HernamtDNA12
 (43),HernamtDNA12 (44),HernamtDNA12 (45),HernamtDNA12 (46),HernamtDNA12
 (47),HernamtDNA12 (48),HernamtDNA12 (49),HernamtDNA12 (5),HernamtDNA12
 (50),HernamtDNA12 (51),HernamtDNA12 (52),HernamtDNA12 (53),HernamtDNA12
 (54),HernamtDNA12 (55),HernamtDNA12 (56),HernamtDNA12 (57),HernamtDNA12
 (58),HernamtDNA12 (59),HernamtDNA12 (6),HernamtDNA12 (60),HernamtDNA12
 (61),HernamtDNA12 (62),HernamtDNA12 (63),HernamtDNA12 (64),HernamtDNA12
 (65),HernamtDNA12 (66),HernamtDNA12 (67),HernamtDNA12 (68),HernamtDNA12
 (69),HernamtDNA12 (7),HernamtDNA12 (8),HernamtDNA12
 (9),Ibo(AF346986),Ibo(AF346987),India(AF346966),India(AF382013),IndianDQ246811,Ing
 man_Australian_AY289051 (1),Ingman_Australian_AY289051
 (10),Ingman_Australian_AY289051 (11),Ingman_Australian_AY289051
 (12),Ingman_Australian_AY289051 (13),Ingman_Australian_AY289051
 (14),Ingman_Australian_AY289051 (15),Ingman_Australian_AY289051

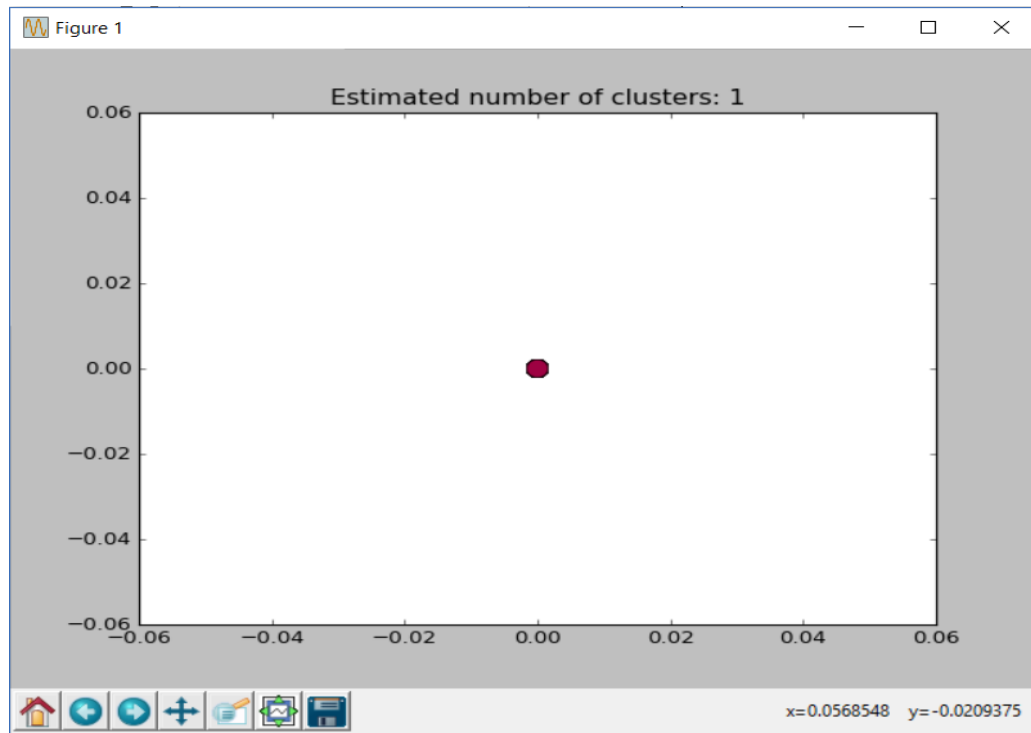
(16),Ingman_Australian_AY289051	(17),Ingman_Australian_AY289051
(2),Ingman_Australian_AY289051	(3),Ingman_Australian_AY289051
(4),Ingman_Australian_AY289051	(5),Ingman_Australian_AY289051
(6),Ingman_Australian_AY289051	(7),Ingman_Australian_AY289051
(8),Ingman_Australian_AY289051	
(9),Italian(AF346988),Japan(AF346989),Japan(AF346990),Kannada(AY289071),Ket(AY519486),Khirgiz(AF346991),Kikuyu(AF346992),Kivisild_Asian_DQ112779	
(1),Kivisild_Asian_DQ112779	(10),Kivisild_Asian_DQ112779
(11),Kivisild_Asian_DQ112779	(12),Kivisild_Asian_DQ112779
(13),Kivisild_Asian_DQ112779	(14),Kivisild_Asian_DQ112779
(15),Kivisild_Asian_DQ112779	(16),Kivisild_Asian_DQ112779
(17),Kivisild_Asian_DQ112779	(18),Kivisild_Asian_DQ112779
(19),Kivisild_Asian_DQ112779	(2),Kivisild_Asian_DQ112779
(20),Kivisild_Asian_DQ112779	(21),Kivisild_Asian_DQ112779
(22),Kivisild_Asian_DQ112779	(23),Kivisild_Asian_DQ112779
(24),Kivisild_Asian_DQ112779	(25),Kivisild_Asian_DQ112779
(26),Kivisild_Asian_DQ112779	(27),Kivisild_Asian_DQ112779
(28),Kivisild_Asian_DQ112779	(29),Kivisild_Asian_DQ112779
(3),Kivisild_Asian_DQ112779	(30),Kivisild_Asian_DQ112779
(31),Kivisild_Asian_DQ112779	(32),Kivisild_Asian_DQ112779
(33),Kivisild_Asian_DQ112779	(34),Kivisild_Asian_DQ112779
(35),Kivisild_Asian_DQ112779	(36),Kivisild_Asian_DQ112779
(37),Kivisild_Asian_DQ112779	(38),Kivisild_Asian_DQ112779
(39),Kivisild_Asian_DQ112779	(4),Kivisild_Asian_DQ112779
(40),Kivisild_Asian_DQ112779	(41),Kivisild_Asian_DQ112779
(5),Kivisild_Asian_DQ112779	(6),Kivisild_Asian_DQ112779
(8),Kivisild_Asian_DQ112779	(7),Kivisild_Asian_DQ112779
(1),Kivisild_Australia_DQ112751	(9),Kivisild_Australia_DQ112751
(3),Kivisild_Australia_DQ112751	(2),Kivisild_Australia_DQ112751
(5),Kivisild_Australia_DQ112751	(4),Kivisild_Australia_DQ112751
(1),Kong_Chinese_AY255138	(6),Kong_Chinese_AY255138
(11),Kong_Chinese_AY255138	(10),Kong_Chinese_AY255138
(13),Kong_Chinese_AY255138	(12),Kong_Chinese_AY255138
(15),Kong_Chinese_AY255138	(14),Kong_Chinese_AY255138
(17),Kong_Chinese_AY255138	(16),Kong_Chinese_AY255138
(19),Kong_Chinese_AY255138	(18),Kong_Chinese_AY255138
(20),Kong_Chinese_AY255138	(2),Kong_Chinese_AY255138
(22),Kong_Chinese_AY255138	(21),Kong_Chinese_AY255138
(24),Kong_Chinese_AY255138	(23),Kong_Chinese_AY255138
(26),Kong_Chinese_AY255138	(25),Kong_Chinese_AY255138
(28),Kong_Chinese_AY255138	(27),Kong_Chinese_AY255138
(3),Kong_Chinese_AY255138	(29),Kong_Chinese_AY255138
(31),Kong_Chinese_AY255138	(30),Kong_Chinese_AY255138
(33),Kong_Chinese_AY255138	(32),Kong_Chinese_AY255138
(35),Kong_Chinese_AY255138	(34),Kong_Chinese_AY255138
(37),Kong_Chinese_AY255138	(36),Kong_Chinese_AY255138
	(38),Kong_Chinese_AY255138

(39),Kong_Chinese_AY255138	(4),Kong_Chinese_AY255138
(40),Kong_Chinese_AY255138	(41),Kong_Chinese_AY255138
(42),Kong_Chinese_AY255138	(43),Kong_Chinese_AY255138
(44),Kong_Chinese_AY255138	(45),Kong_Chinese_AY255138
(46),Kong_Chinese_AY255138	(47),Kong_Chinese_AY255138
(48),Kong_Chinese_AY255138	(5),Kong_Chinese_AY255138
(7),Kong_Chinese_AY255138	(6),Kong_Chinese_AY255138
(9),Koraga(AY289072),Koraga(AY289073),Korea(AF346993),Koryak(AY519487),Leon(AF382004),Leon(AF382005),Leon(AF382006),Leon(AF382007),Lisongo(AF346994),Mandenka(AF346995),Mansi(AY519488),Mansi(AY570524),Maragato(AF382001),Maragato(AF382002),Maragato(AF382003),Mauritania(AF381981),Mauritania(AF381991),Mauritania(AF381992),Mauritania(AF381993),Mauritania(AF381994),Mbenzele(AF346996),Mbenzele(AF346997),Mbuti(AF346998),Mbuti(AF346999),Mkamba(AF347000),Morocco(AF381983),Morocco(AF381984),Morocco(AF381985),Morocco(AF381986),Morocco(AF381987),Morocco(AF381988),Morocco(AF382008),Mullukurun(AY289074),Nasioi(AY289075),Negidal'tsy(AY519489),Nganasan(AY519490),Nganasan(AY519491),Nganasan(AY615359),PNGCoast(AF347002),PNGCoast(AF347003),PNGHigh(AF347004),PNGHigh(AF347005),Pellekaan_Australian_DQ404442	(8),Kong_Chinese_AY255138
(1),Pellekaan_Australian_DQ404442	(2),Pellekaan_Australian_DQ404442
(3),Pellekaan_Australian_DQ404442	(4),Pellekaan_Australian_DQ404442
(5),Pellekaan_Australian_DQ404442	(6),Pellekaan_Australian_DQ404442
(7),Pellekaan_Australian_DQ404442	
(8),Piman(AF347001),Saami(AF347006),Samoa(AF347007),Samoa(AY289093),Samoa(AY289094),San(AF347008),San(AF347009),SibInuit(AF347010),Taiwan(AY289095),Taiwan(AY289096),Taiwan(AY289097),Taiwan(AY289098),Tanaka_Japanese_AP008249	
(1),Tanaka_Japanese_AP008249	(10),Tanaka_Japanese_AP008249
(100),Tanaka_Japanese_AP008249	(101),Tanaka_Japanese_AP008249
(102),Tanaka_Japanese_AP008249	(103),Tanaka_Japanese_AP008249
(104),Tanaka_Japanese_AP008249	(105),Tanaka_Japanese_AP008249
(106),Tanaka_Japanese_AP008249	(107),Tanaka_Japanese_AP008249
(108),Tanaka_Japanese_AP008249	(109),Tanaka_Japanese_AP008249
(11),Tanaka_Japanese_AP008249	(110),Tanaka_Japanese_AP008249
(111),Tanaka_Japanese_AP008249	(112),Tanaka_Japanese_AP008249
(113),Tanaka_Japanese_AP008249	(114),Tanaka_Japanese_AP008249
(115),Tanaka_Japanese_AP008249	(116),Tanaka_Japanese_AP008249
(117),Tanaka_Japanese_AP008249	(118),Tanaka_Japanese_AP008249
(119),Tanaka_Japanese_AP008249	(12),Tanaka_Japanese_AP008249
(120),Tanaka_Japanese_AP008249	(121),Tanaka_Japanese_AP008249
(122),Tanaka_Japanese_AP008249	(123),Tanaka_Japanese_AP008249
(124),Tanaka_Japanese_AP008249	(125),Tanaka_Japanese_AP008249
(126),Tanaka_Japanese_AP008249	(127),Tanaka_Japanese_AP008249
(128),Tanaka_Japanese_AP008249	(129),Tanaka_Japanese_AP008249
(13),Tanaka_Japanese_AP008249	(130),Tanaka_Japanese_AP008249
(131),Tanaka_Japanese_AP008249	(132),Tanaka_Japanese_AP008249
(133),Tanaka_Japanese_AP008249	(134),Tanaka_Japanese_AP008249
(135),Tanaka_Japanese_AP008249	(136),Tanaka_Japanese_AP008249
(137),Tanaka_Japanese_AP008249	(138),Tanaka_Japanese_AP008249

(636),Tanaka_Japanese_AP008249	(637),Tanaka_Japanese_AP008249
(638),Tanaka_Japanese_AP008249	(639),Tanaka_Japanese_AP008249
(64),Tanaka_Japanese_AP008249	(640),Tanaka_Japanese_AP008249
(641),Tanaka_Japanese_AP008249	(642),Tanaka_Japanese_AP008249
(643),Tanaka_Japanese_AP008249	(644),Tanaka_Japanese_AP008249
(645),Tanaka_Japanese_AP008249	(646),Tanaka_Japanese_AP008249
(647),Tanaka_Japanese_AP008249	(648),Tanaka_Japanese_AP008249
(649),Tanaka_Japanese_AP008249	(65),Tanaka_Japanese_AP008249
(650),Tanaka_Japanese_AP008249	(651),Tanaka_Japanese_AP008249
(652),Tanaka_Japanese_AP008249	(653),Tanaka_Japanese_AP008249
(654),Tanaka_Japanese_AP008249	(655),Tanaka_Japanese_AP008249
(656),Tanaka_Japanese_AP008249	(657),Tanaka_Japanese_AP008249
(658),Tanaka_Japanese_AP008249	(659),Tanaka_Japanese_AP008249
(66),Tanaka_Japanese_AP008249	(660),Tanaka_Japanese_AP008249
(661),Tanaka_Japanese_AP008249	(662),Tanaka_Japanese_AP008249
(663),Tanaka_Japanese_AP008249	(664),Tanaka_Japanese_AP008249
(665),Tanaka_Japanese_AP008249	(666),Tanaka_Japanese_AP008249
(667),Tanaka_Japanese_AP008249	(668),Tanaka_Japanese_AP008249
(669),Tanaka_Japanese_AP008249	(67),Tanaka_Japanese_AP008249
(670),Tanaka_Japanese_AP008249	(671),Tanaka_Japanese_AP008249
(672),Tanaka_Japanese_AP008249	(68),Tanaka_Japanese_AP008249
(69),Tanaka_Japanese_AP008249	(7),Tanaka_Japanese_AP008249
(70),Tanaka_Japanese_AP008249	(71),Tanaka_Japanese_AP008249
(72),Tanaka_Japanese_AP008249	(73),Tanaka_Japanese_AP008249
(74),Tanaka_Japanese_AP008249	(75),Tanaka_Japanese_AP008249
(76),Tanaka_Japanese_AP008249	(77),Tanaka_Japanese_AP008249
(78),Tanaka_Japanese_AP008249	(79),Tanaka_Japanese_AP008249
(8),Tanaka_Japanese_AP008249	(80),Tanaka_Japanese_AP008249
(81),Tanaka_Japanese_AP008249	(82),Tanaka_Japanese_AP008249
(83),Tanaka_Japanese_AP008249	(84),Tanaka_Japanese_AP008249
(85),Tanaka_Japanese_AP008249	(86),Tanaka_Japanese_AP008249
(87),Tanaka_Japanese_AP008249	(88),Tanaka_Japanese_AP008249
(89),Tanaka_Japanese_AP008249	(9),Tanaka_Japanese_AP008249
(90),Tanaka_Japanese_AP008249	(91),Tanaka_Japanese_AP008249
(92),Tanaka_Japanese_AP008249	(93),Tanaka_Japanese_AP008249
(94),Tanaka_Japanese_AP008249	(95),Tanaka_Japanese_AP008249
(96),Tanaka_Japanese_AP008249	(97),Tanaka_Japanese_AP008249
(98),Tanaka_Japanese_AP008249	
(99),Tatar(AF346974),Thai(AY289099),Thai(AY289100),Thai(AY289101),Tofalar(AY519492),Tofalar(AY519493),Tofalar(AY615360),Tonga(AY289102),Tubalar(AY519494),Tuvan(AY519495),Tuvan(AY570525),Tuvli(AY570526),Ul'chi(AY519496),Ul'chi(AY519497),Ul'chi(AY615361),Uzbek(AF347011),Yoruba(AF347014),Yoruba(AF347015),mtDNA104,mtDNA108,mtDNA136,mtDNA140,mtDNA142,mtDNA149,mtDNA153,mtDNA155,mtDNA156,mtDNA158,mtDNA159,mtDNA160,mtDNA162,mtDNA163,mtDNA164,mtDNA165,mtDNA166,mtDNA172,mtDNA173,mtDNA175,mtDNA180,mtDNA189,mtDNA192,mtDNA193,mtDNA194,mtDNA195,mtDNA196,mtDNA199,mtDNA207,mtDNA208,mtDNA211,mtDNA215,mt	

DNA216,mtDNA217,mtDNA222,mtDNA223,mtDNA233,mtDNA242,mtDNA293,mtDNA309,mtDNA328,mtDNA379,mtDNA380,mtDNA381,mtDNA382,mtDNA385,mtDNA386,mtDNA387,mtDNA388,mtDNA389,mtDNA401,mtDNA413,mtDNA421,mtDNA434,mtDNA514,mtDNA560

Figure 18: All DNA sequences clustered into 1 cluster



At epsilon 1.5 and number of samples = 100, all the data has been put in one cluster. The reason being DNA of humans is highly similar in nature.

9. Future Work and Enhancements

9.1. Use Database

In future, instead of maintaining tree structure, the database can be used. For example – Information stored in a PTA tree of figure 17 displayed below can be represented using following tables.

- Associate Transition Matrix – Stores transition from one node to another when a symbol is passed to it
- Number Reached – Stores number of strings arriving at each node
- Number Accepted – Number of String ending at a Node.
- Number Following – Number of Strings following an arc.

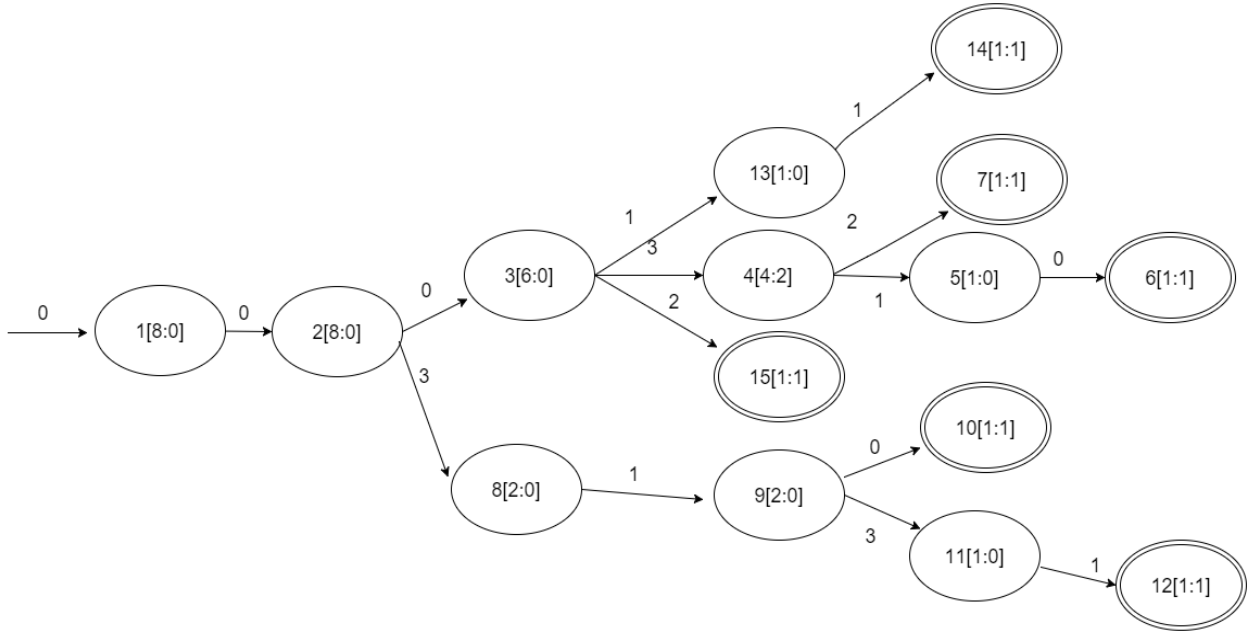


Figure 19: PTA tree

7.1.Associated Transition Matrix Representation of PTA Tree

PTA tree in Figure 17 can be represented as follows.

Column 0,1,2,3 represents alphabets in DNA automaton. Data can be read as follows:

$$\delta(1, 0) = 2$$

Node ID	0	1	2	3
1	2	-	-	-
2	3	-	-	8
3	-	13	15	4
4	-	5	7	-
5	6	-	-	-
6	-	-	-	-
7	-	-	-	-
8	-	9	-	-
9	10	-	-	11
10	-	-	-	-
11	-	12	-	-

12	-	-	-	-
13	-	14	-	-
14	-	-	-	-
15	-	-	-	-

Table 16: Associate Transition Matrix

9.2.Number of Strings Accepted

Table below represents Number of Strings accepted at each Node for figure 17 For example number of strings ending at node 1 in figure 17 are 0.

Node ID	Number of String Accepted
1	0
2	0
3	0
4	2
5	0
6	1
7	1
8	0
9	0
10	1
11	0
12	1
13	0

14	1
15	1

Table 17: Number of String Accepted by Nodes in PTA

9.3.Number of String Arriving at each node

Table below represents the number of strings arriving at each node. For example Node 1 has 8 strings that passed through it.

Node ID	Number of String arriving at each node
1	8
2	8
3	6
4	4
5	1
6	1
7	1
8	2
9	2
10	1
11	1
12	1
13	1
14	1
15	1

Table 18: Number of String arriving at each node

9.4.Benefit of Using Database

1. Eliminates Data Redundancy i.e. instead of each node storing information about its children over an alphabet, a table as represented in Table 4 can be maintained . Therefore, whenever there is a need to know, if transition exists from one node to another over an alphabet, a simple database lookup can be performed.
2. Data Consistency – By making Node ID as primary and using it as a foreign key for other tables, it can be ensured that whenever a node is changed, its changes are reflected in other tables as well.

10. Conclusion

It is concluded that by merging equivalent states of DNA, we can generate a Stochastic Finite Automata that when passed with the DNA of other species predicts the similarities between them. The whole process takes linear time on average. In particular a finite automata of the human species was learned and then equivalent states were merged using Alergia[5] algorithm to produce a merged state Stochastic Finite Automata. The results were compared against similarities known in biology till date. Most accurate results were found at alpha 0.5 and 0.6.

The results indicate that state merging algorithm are very effective for capturing patterns, especially for DNA data and can help bio-informatics researchers in their research. It is more than capable of generating non matching sequences with high precision. Some applications that can be created based on the results are:-

- Training with healthy DNA and passing diseased DNA to check if a match occurs.
- Using the system to generate reports of rejected strings and study them to understand dissimilarities of different species.
- Alternatively, generate reports of similar sequences between DNA's to understand the position of similarities.
- Understanding the proteins detected by the system.

Avenue of future work that can enhance software's performance and applications.

- Use of database to store results, in that way we only compute the Stochastic Finite Automatas of DNA species once and store the results in database.
- If a change in data occurs then only re-compute Stochastic Finite Automata of new data.

- Adding database opens the opportunity of hosting a web application that can display our results.

11. References

- [1] Carrasco, Rafael C. "Grammatical Inference: An Introductory Survey." *Grammatical Inference and Applications: Second International Colloquium, ICGI-94, Alicante, Spain, September 21-23, 1994: Proceedings*. Berlin: Springer-Verlag, 1994.
- [2] E. M. Gold: Language Identification in the Limit. *Inf. And Control*, Vol.10, pp.447-474, 1967.
- [3] Carrasco, Rafael C. "Grammatical Inference: What is the search space of the regular inference" *Grammatical Inference and Applications: Second International Colloquium, ICGI-94, Alicante, Spain, September 21-23, 1994: Proceedings*. Berlin: Springer-Verlag, 1994. Pp 25- pp27
- [4] A. Aho and J. Ullman. *The Theory of Parsing, Translation and Compiling*, Vol.1: Parsing, Series in Automatic Computation, Prentice-Hall, Englewood, Cliffs, 1972.
- [5] Carrasco, R. C., & Oncina, J. (1994). Learning stochastic regular grammars by means of a state merging method."Alergia" In *Grammatical Inference and Applications* (pp. 139-152). Springer Berlin Heidelberg.
- [6] Carrasco, Rafael C., Oncina Jose "Grammatical Inference: Learning Stochastic Regular Grammars by Means of a State Merging Method" *Grammatical Inference and Applications: Second International Colloquium, ICGI-94, Alicante, Spain, September 21-23, 1994: Proceedings*. Berlin: Springer-Verlag, 1994. pp 142 - 147
- [7] Starr, Dr.Barry. "Understanding Genetics." *Understanding Genetics*. Stanford University, 16 Dec. 2008. Web. 13 Nov. 2015. <<http://genetics.thetech.org/ask/ask293>>.
- [8] Wikipedia contributors. "DNA." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 13 Nov. 2015. Web. 13 Nov. 2015.

- [9] Lang, Kevin J., Barak A. Pearlmutter, and Rodney A. Price. "Results of the Abbadingo One DFA Learning Competition and a New Evidence-driven State Merging Algorithm." *Grammatical Inference Lecture Notes in Computer Science*: pp 9- pp12.1998
- [10] Lamar, R Bowen. "Chromosome Numbers in Different Species." *Chromosome Numbers in Different Species*. Web. 16 Nov. 2015.
- [11] "Basic Local Alignment Search Tool." *BLAST*: Web. 20 Nov. 2015.
- [12] Martin, J. C. & Hawk, J. F.: DNA sequence analysis by optical pattern recognition. In: The International Society for Optical Engineering, 938, pp 38- pp 45(1988)
- [13] Jir Poner , Filip Lanka.: Computational Studies of RNA and DNA. Challenges and Advances in Computational Chemistry and Physics (2006)
- [14] Justin Davis.: Finite State Machine Datapath Design, Optimization, and Implementation (Synthesis Lectures on Digital Circuits and Systems) (2008)
- [15] Anderson C., Brunak S.: Representation of Protein Sequence Information by Amino Acid Subalphabets. In: American Association for Artificial Intelligence, Volume 1, 97-104 (2004)
- [16] Pontius, J. U., J. C. Mullikin, D. R. Smith, K. Lindblad-Toh, S. Gnerre, M. Clamp, J. Chang, R. Stephens, B. Neelam, N. Volfovsky, A. A. Schaffer, R. Agarwala, K. Narfstrom, W. J. Murphy, U. Giger, A. L. Roca, A. Antunes, M. Menotti-Raymond, N. Yuhki, J. Pecon-Slaterry, W. E. Johnson, G. Bourque, G. Tesler, and S. J. O'brien. "Initial Sequence and Comparative Analysis of the Cat Genome." *Genome Research* 17.11 (2007): 1675-689. Web.
- [17] For example, D.E. Wildman et al., "Implications of Natural Selection in Shaping 99.4% Nonsynonymous DNA Identity between Humans and Chimpanzees: Enlarging Genus Homo," *Proc. Natl. Acad. Sci.* 100 no. 12 (2003): 7181–7188.

- [18] Discussed in D.A. DeWitt, "Greater than 98% Chimp/Human DNA Similarity? Not Any More," *TJ* 17 no. 1 (2003): 8–10.
- [19] The Bovine Genome Sequencing and Analysis Consortium*, Christine G. Elsik¹, Ross L. Tellam², Kim C. Worley Department of Biology, 406 Reiss, Georgetown University, 37th and O Streets, NW, Washington, DC 20057, USA. E-mail: ce75@georgetown.edu
- [20] "Initial Sequence and Comparative Analysis of the Cat Genome." *Genome Research* 17.11 (2007): 1675-689. Pontius, J. U., J. C. Mullikin, D. R. Smith, K. Lindblad-Toh, S. Gnerre, M. Clamp, J. Chang, R. Stephens, B. Neelam, N. Volfovsky, A. A. Schaffer, R. Agarwala, K. Narfstrom, W. J. Murphy, U. Giger, A. L. Roca, A. Antunes, M. Menotti-Raymond, N. Yuhki, J. Pecon-Slatery, W. E. Johnson, G. Bourque, G. Tesler, and S. J. O'Brien.
- [21] Background on Comparative Genomic Analysis. Nobrega, Marcelo A., and Len A. Pennacchio. "Comparative Genomic Analysis as a Tool for Biological Discovery." *The Journal of Physiology* (2003): pp31-pp39.
- [22] "2004 Release: Researchers Compare Chicken, Human Genomes." *2004 Release: Researchers Compare Chicken, Human Genomes*. Web. 18 Dec. 2015.
- [22] Genome evolution. (2015, December 16). In *Wikipedia, The Free Encyclopedia*. Retrieved 20:17, December 18, 2015, from https://en.wikipedia.org/w/index.php?title=Genome_evolution&oldid=695470952
- [23] Human genetic variation. (2015, December 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 20:24, December 18, 2015, from https://en.wikipedia.org/w/index.php?title=Human_genetic_variation&oldid=693605785

- [24] Recent African origin of modern humans. (2015, December 18). In *Wikipedia, The Free Encyclopedia*. Retrieved 20:32, December 18, 2015, from https://en.wikipedia.org/w/index.php?title=Recent_African_origin_of_modern_humans&oldid=695738311
- [25] Ingman, M. & Gyllensten, U. mtDB: Human Mitochondrial Genome Database, a resource for population genetics and medical sciences. *Nucleic Acids Res* 34, D749-D751 (2006).
- [26] Wikipedia contributors. "DBSCAN." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 12 Nov. 2015. Web. 19 Dec. 2015.