Computational Approach for Homology Discovery of Keratin Digestion Genes in Zebrafish

A.M. Dissanayake

Index Number: 13000314

Supervisors : Mrs. M.W.A.C.R. Wijesinghe

Dr. U.K. Premaratne

May 2018

Submitted in partial fulfillment of the requirements of the
B.Sc (Hons) in Computer Science Final Year Project (SCS4124)

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: A.M. Dissanayake

……

Signature of Candidate Date:

This is to certify that this dissertation is based on the work of

Ms. A.M. Dissanayake

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Supervisor Name: Mrs. M.W.A.C.R. Wijesinghe

……

Signature of Supervisor Date:

Abstract

Computational approaches for gene prediction have drawn a significant importance considering the pace at which raw sequences of biological data is getting available in past few decades where biological experiments for drawing the meaningful insights from these raw data have failed to meet this pace. This research study focuses on gene prediction towards the functionality of keratin digestion in scale eating.

For this gene prediction, genomic data of *zebrafish* is used against the known keratin digestion data of *keratin-feeding clothes moths* and *keratin beetles.* Since fishes and insects are highly different organisms, it created the requirement to build a comprehensive pipeline for the gene prediction. Hence we first clustered the Expressed Sequence Tags (ESTs) and then they were passed through a motif discovery process. As the next step, those motifs were matched against the genome of zebrafish by performing a homology search. Exhibiting promising results, we could achieve a match hit with an E - value of 0.058 that starts at the location of 14411 bp in the genome of zebrafish.

To further evaluate the obtained match, a requirement to develop a model that can claim whether a given sequence is a gene or not was raised. As such, in the next phase of the pipeline, a Markov model for CpG island prediction was designed and developed and that model successfully shows an accuracy of 93.5%. Finally, we passed the starting region of the obtained match to this model and most importantly, the model predicted it as a CpG island. This suggests that the obtained match exhibits the properties of a gene which can be considered as the ultimate highest goal that can be achieved in a computational gene prediction research.

Keywords*: Gene Prediction, Homology Discovery, Keratin, Lepidophagy, CpG Island Prediction, Markov Models*

Preface

This research builds a computational pipeline that processes from one step to another for the ultimate keratin digestion gene prediction. This pipeline that is proposed to acquire a homology match for keratin digestion is solely my design. I could not find any other computational gene prediction research attempts on keratin digestion or on analysing keratin digestion in scale eating, to best of my knowledge. Tools, algorithms and methodologies to be followed in each step on the pipeline were analysed and selected from a comprehensive literature survey.

The machine learning approach taken for the prediction of CpG islands is mostly inspired from the similar researches in the domain such as the research done by M. Lan et al. [49] which was done for CpG Islands Identification in Human. Nevertheless, I should state that the model developed for the prediction of CpG island in Zebrafish is my own work.

Acknowledgement

I would like to express my deepest gratitude to my supervisors Mrs. M.W.A.C.R. Wijesinghe and Dr. Upeka Premaratne for their valuable and constructive suggestions, kind co-operation and encouragement which helped me to complete this research successfully and also willingness to give their time generously is very much appreciated. Also, I would like to thank Dr A.R. Weerasinghe for the advice and suggestions given to carry out this one year research project more successfully.

I also bestow my honor to Dr A.M. Premachandra and Dr T.A. Weerasinghe for the guidance they offered me by providing their valuable feedback as examiners.

I also express my sincere gratitude for all my friends who supported and encouraged me on all cause of challenges I faced during this research. All the help given by everyone to make this research a useful project owns my great appreciation. At last but not least I would like to thank my parents for their enormous strength which encouraged me to successfully carry out this research. They are the guiding stars which strengthen me to become the person who I am today.

Table of Contents

List of Figures

List of Tables

List of Acronyms

bp - base pairs

EST – Expressed Sequence Tag

NCBI - National Centre for Biotechnology Information

BLAST - Basic Local Alignment Search Tool

TGICL - TIGR Gene Indices clustering

DNA - Deoxyribonucleic acid

RNA - Ribonucleic Acid

mRNA - messenger RNA

MM - Markov Model

HMM - Hidden Markov Model

HMT - Hidden Markov Tree

Chapter 1 - Introduction

This chapter would lay the foundations for the dissertation where it would discuss the background to the research and the facts that motivated the research. To get a better understanding, some biological concepts that are related to the research would be introduced. Then the research question and objectives is presented which is followed by the scope and limitations of the research.

1.1 Background to the Research

"Bioinformatics is what employs computational methods in order to advance the scientific understanding of living systems"

Bioinformatics is the area which analyzes the information associated with biological data by building interdisciplinary links between biology, computer science, mathematics and statistics. Over the past few years, the availability of various biological datasets resulted from advancements in biotechnology have grown at a phenomenal rate. This has offered the opportunity to draw meaningful insights from these raw sequences of data. Hence designing computational methods are becoming increasingly important in order to extract hidden knowledge that can have a precise impact on various different fields. One such meaningful insight is predicting the genes that are responsible for a particular functionality of an organism which is considered as a key contribution to the field of bioinformatics.

An interesting functionality that can be analyzed using gene prediction methods is the lepidophagous behaviour. Lepidophagous behaviour is a specialized feeding behaviour of fishes who feed on fish scales of other fishes as a special diet. Figure 1.1 shows how a lepidophagous fish preys on fish scales of another fish. It is observed that this scale-eating behaviour is known for several unrelated fish groups [1] such as characoids fish groups, danionin fish groups and Catoprion mento fishes.

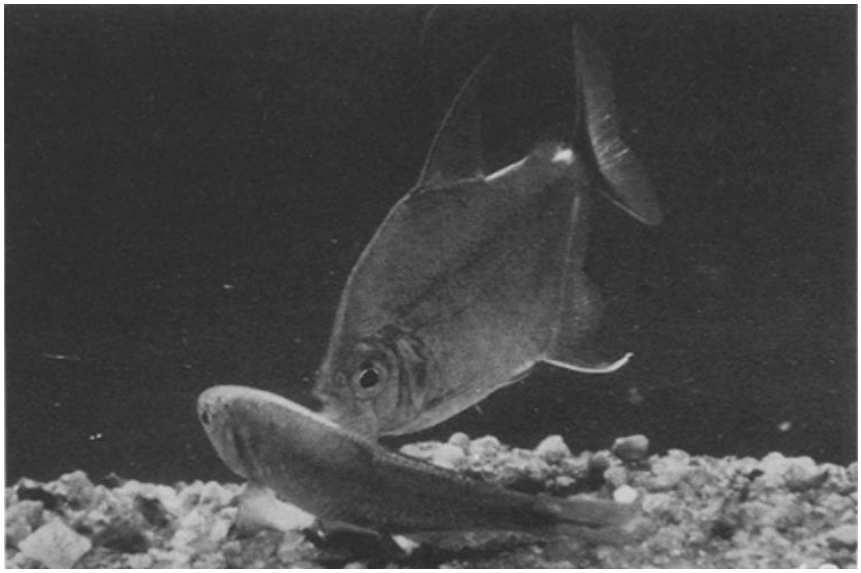


Figure 1.1: *Catoprion mento* feeding on fish scales [1]

This research focuses on the prediction of gene towards lepidophagy behaviour and for that, it uses a tropical freshwater fish named as Zebrafish which is scientifically known as *Danio rerio*.

Zebrafish is particularly selected for this research based on two reasons. Firstly that it is observed as a fish that belongs to a fish group that feeds on fish scales of other fishes. McClure et al. [2] have found out that there were fish scales in the gut content of *danionin* species in the form of prey processing as well as in the form of digestion during their study of analyzing the natural diet of *danionin* fishes including the zebrafish *Danio rerio.* Secondly only a few organisms' complete genome is currently known such as human, chimpanzee and mouse, and most importantly zebrafish is among them.

Surprisingly, scales are a relatively nutritious food source that have layers of keratin [3]. Keratin is an insoluble complex polypeptide and its complex structure has resulted in keratin being a component which is highly resistant to enzymatic degradation [8]. This characteristic property of keratin has ended up in setting up bottlenecks in the industries such as leather industry that inherently need the processing of keratin [8].

So this research is intended on analyzing potential genes that have enabled the capability of keratin digestion in scale eating behaviour of zebrafish. For that, a gene prediction towards the function of keratin digestion is performed using known data for keratin digestion of keratin-feeding clothes moths and keratin beetles.

1.2 Motivation

The above mentioned observations of scale eating is an interesting topic in zoology since it is special yet rare behaviour. Hence many biological researches have been done on the topic [1, 3], but none of the bioinformatics research could be found that was done on primarily focusing the computational prediction on the potential functionality of scale digesting.

Currently, experiments are going on developing different by-products on industrial aspects of fish scales such as pharmaceutical products [4], cosmetics and food supplements, protein rich organic fertilizer [7] considering the fact that fish scales are a rich host of nutrients. Despite the fact that there are tons of fish scales being wasted and thrown away every day, fish scale processing for such industries is not expanding rapidly. One of the reasons would be that keratin is highly resistant to enzymatic degradation [8].

That being said, there is an interesting observation from nature that there exists a few numbers of organisms who are with the natural capability of digesting keratin such as the above mentioned fishes, cloth moths, keratin beetles and *Microsporum canis* fungus.

Discovering the potential genes behind this natural capability leads ways to the development of enzymes that can be used in the fish scale processing industry. Hughes et al. [8] who have studied deep into keratin digestion capability in insects state the importance of predicting the potential genes that enable this metabolism which then can be used *in development of enzymes in laboratories* to solve the real world bottlenecks encountered by the industries that

are *associated with the processing of keratin containing components*. Keratinaceous waste streams are gathered in unmanageable contents daily and they are with high potential to be converted into animal-derived biomass and development of protein rich fertilizer [16]. So the importance of discovering keratin digestion genes have drawn a major attention [16]. Hughes et al. [8] state that the catabolic pathways of keratin digestion are of great interest in the leather industry as well for the removal of hair. They further state that once the potential keratin digestion genes are identified, those proteinase inhibitors can then be used to prevent damage from clothes moths.

Discovering the genes that perform a particular functionality can be done through biological experiments, but they continue to be a laborious task which requires enormous resources [23]. This is the reason why there is a huge gap between the number of sequence data available and the number of experimentally characterised genes [5, 6, 23]. Hence developing computational approaches have become significantly necessary.

1.3 Explanation of some biological concepts

1.3.1 Gene prediction and Homology search

Gene prediction methods in bioinformatics are used to predict the genes that are responsible *for a particular functionality of an organism*. In the context of this research, this particular functionality would be keratin digestion. For gene prediction in bioinformatics, a dataset of genes of an organism that is known to carry out a particular functionality is used [10]. Then that dataset is matched against another organism who is known to have the capability of doing that same functionality. This is called a homology search [10, 14]. If a homology (similarity) match is found in that organism, that matching genes are said to be with the capability of performing that same functionality.

1.3.2 Expressed Sequence Tags (ESTs)

Expressed Sequence Tag is a short sub-sequence of a cDNA sequence which represents portions of expressed genes. In this research, as the known dataset, we use Expressed Sequence Tags of clothes moths and keratin beetles that got expressed during digestion of keratin in their guts.

1.3.3 Motifs

Motifs are short, recurring patterns in proteins that are presumed to have a biological function. A homology search becomes more accurate when motifs are used [9]. Hence, we use motifs of Expressed Sequence tags for biological function of keratin digestion in the context of this research.

1.3.4 CpG islands to identify genes

CpG islands are 'start region' of genes. CpG islands in the promoter region express a gene and a CpG island in the promoter region of a gene is methylated, expression of the gene is repressed (it is turned off) [37]. Therefore, CpG islands play a major role in identifying a gene as if it is a gene then CpG island can be found in the starting region as shown in Figure 1.2

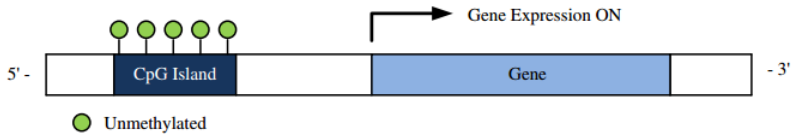


Figure 1.2: Starting region of a gene [38]

1.4 Research question and Objectives

1.4.1 Research question

The main research question that we address in this study is as follows,

Are there any potential undiscovered genes that are responsible for the ability of keratin digestion on scale eating in zebrafish?

1.4.2 Project aims and objectives

The main aim of this research is to explore whether there exist any potential undiscovered genes that have the capability of keratin digestion on scale eating in zebrafish.

Therefore the objectives of this study can be defined as follows which would build a computational pipeline to achieve the ultimate aim,

* Obtain motifs in the clusters of expressed sequence tags of clothes moths and keratin beetles that perform the function of keratin digestion

* Obtain homology match between above identified motifs and the genome of zebrafish for the function of keratin digestion

* Design and develop a model to analyze a homology match in order to claim it as a potential gene

1.5 Scope and limitations

* Data available for this research are the complete genomes of zebrafish and Expressed Sequence Tags of keratin-feeding clothes moths (920 ESTs) and keratin beetles (883 ESTs) which are taken from publicly available datasets

* Lepidophagous behaviour of zebrafish is analyzed through only the functionality of keratin digestion

* Gene prediction for keratin digestion is derived considering only the organism level of insects with zebrafish

1.6 Outline of the Dissertation

This dissertation is organized in six main chapters including the Introduction chapter. So the remainder of this dissertation is organized as follows.

With the precise introduction given in Chapter 1, this dissertation would then describe,

Chapter 2: Provides the relevant Literature Review

Chapter 3: Describes proposed research design including key concerns and architectural aspects.

Chapter 4: Is dedicated for the implementation details of the research

Chapter 5: Presents the results of the research with relevant discussions

Chapter 6: Concludes the study highlighting major contributions, limitations and future work

1.7 Summary

This chapter provides a precise introduction to the research. The motivation behind the research was justified, and then it introduced the research problem and research questions and hypotheses. The dissertation was outlined, and the limitations were given. On these foundations, the dissertation can proceed with a detailed description of the research.

Chapter 2 - Literature Review

This Chapter provides the literature review with regard to the concepts and methodologies related to the research. It would first analyzed the other sequence similarity based researches, researches that involved keratin digestion functionality and then would provide a comprehensive comparison of available EST clustering algorithms. Finally two different methods used for the prediction of CpG island are deeply analyzed.

2.1 Sequence Similarity (Homology) Search

With the advancement of the biological researches, many raw genome related sequences are extracted, yet only a few of those sequences have been experimentally characterised, meaning the functionality of the most of those sequences performed in an organism has not yet been discovered with biological experiments due to their requirement of enormous resources. Hence computational biology comes into the picture to bridge this gap between the raw sequences and prediction of their functionalities [9-11]. Out of thousands of examples of such function predictions, one example is that, almost all the functions of the genome of *Methanococcus jannaschii* have been obtained from function predictions using similarity searches and not from biological experiments [12].

Pearson [10] states that sequence similarity search is one of the first and most informative way of conducting an analysis of newly determined sequences. The functionality of such sequences can be inferred from similarity search. In this study, we are trying to predict the function of keratin digestion in scale digesting. Pearson [9] further claims that homologous search is an “effective and accurate” way and it is the most popular mechanism on inferring sequences that behave functionally same.

In researches when evaluating a homology match, generally an E-value is used. But the value for taken as the E value is context sensitive and as such, it vastly varies from one research to another [45]. C. A. Kerfeld et al. [45] who have studied on the E-value suggest that selecting the E-value for a significant homology match would depend on the sequences on which one tries to obtain a homology.

2.2 Identification of keratin digestion functionality

Identifying potential genes for keratin digestion is of greater interest and due to its high demand in keratin processing industries, the commercial value of such discoveries are also higher. Industries that develop leather products, animal-derived biomass and protein rich fertilizer etc. have high requirement of enzymes for keratin degradation from such potential genes [16].

Only in very recent past, computational biology came into the picture for identifying potential keratin digestion genes and until then biological experiments have achieved so little in isolating keratin digestion enzymes [16]. It is expected to identify potential keratin digestion genes using novel computation biological methodologies and then using that knowledge to develop enzymes inside biotechnical laboratories can pass the barriers that currently exist.

Until 2006, identifying potential genes for keratin digestion is done only on microorganism level which does not give satisfactory results in industrial aspects [8].

2.2.1 Gene prediction for keratin digestion in insects

J. Hughes et al. [8] have first tried on the identification of potential keratin genes in insects, specifically keratin-feeding clothes moths (*Tineola*) and keratin beetles (*Trox*) which had previously tried only at microorganism level. This study is primarily based on this research and we are trying to bring it to the level of fish from insects. In the study done by J. Hughes et al. [8], they have fed clothes moths and keratin beetles on a meal of keratin that contains a mixture of human hair, feathers and wool. Then after letting them to digest keratin they have extracted the expressed sequences in the gut and thorax muscle of these two insects that got expressed during the keratin digestion process.

Then J. Hughes et al. [8] have conducted a gene prediction from homology search for keratin digestion with comparisons to known protein sequences which are serine proteases. They also have claimed that *Tineola* and *Trox* are two species that substantially differ in morphology. And for homology alignment matching they have used Blastx tool. Finally for the results from sequence similarity predictions for keratin digestion they have obtained percent identical residues of 22.4% for *Tineola* and 6.8% for *Trox.*

2.3 Expressed sequence tags clustering

Expressed Sequence Tags (ESTs) are sequences that are used to explore the transcriptome which means a record of gene activity. ESTs are short fragments of DNA created in the laboratory [17]. They are widely used for gene discovery and expression analysis [18]. Below sections would provide a comprehensive literature study on main clustering algorithms comparing inherent strengths and weaknesses of each in order to identify the best suiting algorithm to be used in this research.

2.3.1 d2_cluster algorithm

d2_cluster, is an agglomerative algorithm for rapidly and accurately partitioning transcript databases discovered by J. Burk et al. [19]. In d2_cluster every sequence begins in its own cluster, and the final clustering is achieved using mergings. The criterion for merging clusters is the detection of two sequences that share a window of bases that is a threshold percent or more identical.

As it can be seen in above described steps of d2_cluster, the algorithm is designed in such a way that it can be get easily familiarized to statisticians, computer scientists, and biologists alike [19]. Hence d2_cluster, is widely used and has been established as producing valid and useful results from the scientific point of view [20]. But this algorithm fails in situations where it requires to join a valid cluster that was generated with another method or if that method introduces a false join or if requires different clustering criterion methods [19].

2.3.2 TIGR Gene Indices clustering (TGICL)

In TGICL is a pipeline for EST clustering where the sequences are first clustered based on pairwise sequence similarity, and then assembled by individual clusters to produce longer, more complete consensus sequences. It uses internal graph representations where sequences represent nodes and filtered alignments represent edges [18]. One of the fundamental advantages of this is that it performs a fast clustering of large EST datasets, where its researchers claim that sets of 150 000 ESTs can be fully clustered and assembled overnight on a single CPU [18].

However S. Hazelhurst et al. [17] state that the primary aim of programs like TGICL that perform EST clustering is on supervised or seeded clustering. Hence it can then perform clustering with an approximate matching of ESTs against a known genome [17]. Further, they state that TGICL like programs have significant computational cost overhead in its sequence assembly phase.

2.3.3 CLU: A new algorithm for EST clustering

This is based on CLU match detection algorithm, which has improved performance over the widely used d2_cluster clustering algorithm. After using match detection algorithm, it performs clustering based on inter-cluster distance which is taken from the nearest neighbor distance [20].

The set of the algorithms that used in previous generation of EST clustering such as d2_cluster, CAP3, TGICL are not primarily based on techniques that analyze properties of EST data but instead shotgun sequences [21]. S. H. Nagara et al. [21] state that new generation algorithms such as CLU have been developed specifically for EST clustering and assembly and will continue to play a central role in the analysis ESTs.

2.3.4 wcd EST clustering

wcd (name given to the algorithm which is pronounced as *wicked*) is the algorithm selected to perform EST clustering in this research.

wcd performs an efficient all-versus-all comparison of ESTs. For the clustering purposes, it uses both d2 distance function and edit distance [22]. Researchers have improved existing implementations of d2 [19] when it is used in wcd. It takes heuristics parameters for speedup but they do not affect the quality of results but increase the clustering speed and window size parameter to determine how long the overlap should be during alignment. In an overview done to wcd by S. Hazelhurst et al. [22], compared to other clustering tools it has achieved a better clustering when all of them were used in default parameters. Table 2.1 provides a summary about all the clustering algorithm that were analyzed so far.

Table 2.1: Summary of the Clustering Algorithms

Summary	d2_cluster algorithm	TGICL	CLU algorithm	wcd algorithm
General Analysis	* Discovered by J. Burk et al. (1999) * Is an **agglomerative algorithm** for rapidly and accurately partitioning transcript databases * Every sequence begins in its own cluster * Criterion for merging clusters is two sequences that share **a window of bases more than a threshold** percentage identical	* First clustering based on **pairwise sequence similarity** * Then assembled by individual clusters to produce longer, more complete sequences	* CLU match detection algorithm * First uses a **match detection algorithm** * Then performs clustering **based on inter-cluster distance** which is taken from the **nearest neighbour distance**	* In clustering, for distance purposes, it uses **both d2 distance function or edit distance** * ASn **efficient all-versus-all comparison** of ESTs
Importance and advantages	* **Easily familiarized** to statisticians, computer scientists, and biologists alike * Hence **widely used**	* **Builds a pipeline** for EST clustering * Performs a **fast clustering of large** EST datasets	* An i**mproved performance over the widely used d2_cluster** clustering algorithm. * **A new generation algorithms** meaning not like previous generation, this algorithm is	* Researchers have **improved existing implementations of d2** when used in wcd * **A new generation algorithms** * Wcd remains significantly **more sensitive** than the others EST clustering

			primarily concerned on ESTS	algorithms * **Speed up heuristics and window size** is user input parameters, so user can determine the clusters he wants * more **robust** to errors
Limitations	* Would not work if it requires to join a valid cluster that was generated with another method * Or if method **introduces a false join** * Or if requires **different clustering criterion** methods	* Primarily designed to perform EST clustering is on **supervised or seeded clustering** * To work against a known genome with **an approximate matching** of ESTs * Significant **computational cost** overhead in its sequence assembly (second) phase	* Generates a cluster consensus based on unsorted pair-wise alignments only * The quality of results is curbed by the performance **limitation of a desktop PC**	* **Parallelizing** algorithm is yet to be looked

2.4 CpG islands prediction

CpG islands are the 'start region' of genes in a genome which is rich of a C followed by a G in the 5' to 3' direction where p in CpG implies the 5' to 3' direction. As it marks start region of genes, in the process of identifying the gene mutation and gene regulation, CPG islands play an important role [24-25].

H. Shu et al. [26] discuss about the importance of keeping the focus on CpG proteins for when analysing genes. The reason that CpG islands prediction is important for this research is that using a CpG islands prediction model, a homology match in the genome of zebrafish can analyzed to see whether it is a gene or not.

2.4.1 Cutoff based algorithms to predict CpG islands

The traditional methods that were used to predict CpG islands are based on the cutoffs that were generated from the definitions created for CpG islands that involves their "GC content" and observed to expected CpG ratio (O/E). Hence the algorithms that were used to predict CpG islands are based on GC content and observed to expected CpG ratio (O/E) [27].

Those traditional algorithms used the cutoffs of GC content > 50% and O/E > 0.6 which was first defined by M. Gardiner-Garden et al. [28]. Since then, different approaches have been tried such as using different cutoff values [29] instead of the above values of the definition. J. L. Glass et al. [30] have used an approach of selecting these cutoff values from a drawn histogram of the CpG island distribution and based on the length of a segment needed to cover the nearest 27 CpGs. However, this 27-CpG requirement results in leaving out many shorter CpG islands [31].

But the problem of using cutoff values for the prediction of CpG islands is that neither a biological argument nor a formal statistical motivation was used in picking up these cutoff

values which were based on a difficult to interpret scale [27]. In fact, the results of prediction of CpG islands vastly varied from the CpG islands that were identified later through biological experiments [25, 27].

2.4.2 Markov models to predict CpG islands

Markov model based approaches have been used for sequence analysis more recently such as partition genomes into segments. In general, Markov models have been extensively used in different sequence analysis to discover functional elements in various genomes [27].

N. Dasgupta et al. [25] have first tried applying Markov models for the prediction of CpG islands. They have tried a Markov Model (MM), Hidden Markov Model (HMM), and a wavelet-based Hidden Markov Tree (HMT) to prediction positions of CpG islands in the human genome. HMT model has yielded a larger set of declared CpG islands compared to the MM and HMM. The HMM employs two hidden states: one characteristic of an underlying CpG region, the other characteristic of non-CpG data. However all those three algorithms have predicted the position of CpG islands beyond the length of the actual CpG islands [25].

H. Wu et al. [27] have also tried an approach of using Hidden Markov Model for prediction of CpG islands. They claim that since the underlying structure of the genome with base and CpG includes unobserved states which are presumed to be locally correlated along the genome, HMMs are a natural method to be considered for the prediction of CpG islands. They have designed the HMM using three states namely Alu repetitive elements, baseline, and CpG. Then to obtain transition probabilities they considered that CpG implies the probability of a C at location t followed by a G is less likely than would be predicted by chance under independence: $pCG(t) < pC(t) \times pG(t + 1)$. . For the results they could increase number of CpG islands predicted by 81% than the Genome Browser CpGs and 86% than the research done by L. Glass et al. [30].

M. Lan et al. [49] have also taken a machine learning approach to predict CpG islands of human genome. They have used an HMM of eight states which are A+, C+, G+, T+, A-, C-, G-,T-, where plus and minus indicates the transitions inside and outside the cpg islands. They have used a 200bp length definition for CpG islands which is important for our research when deciding the starting region of an obtained homology match. Furthermore, this research clearly presents evaluation metric results so that it can be easily compared against another newly introduced model can be compared against it. Figure 2.1 shows their performance measures.

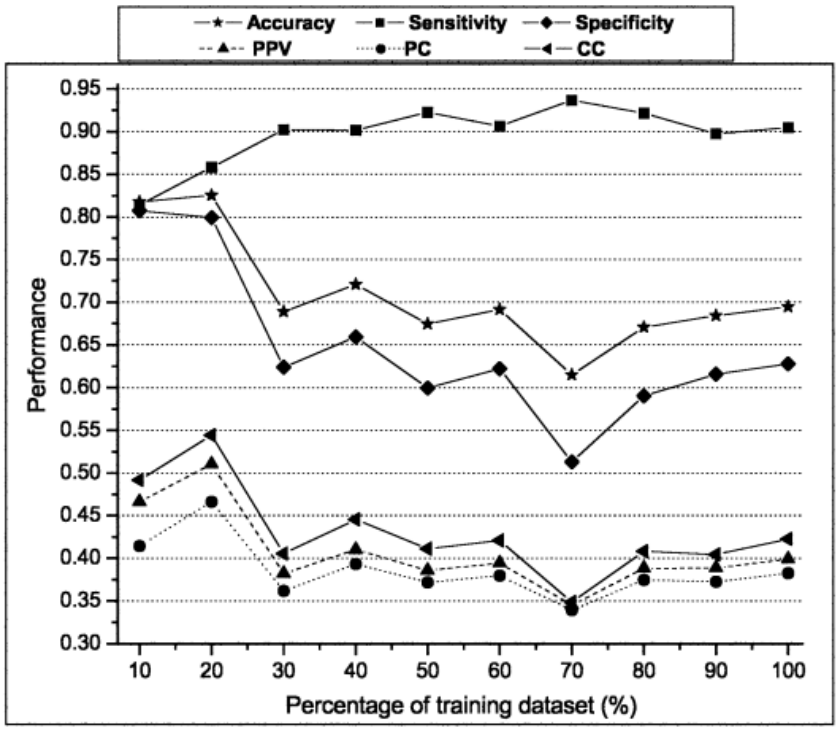


Figure 2.1 performance measures on subsets of training data set from research [49]

They have claimed that the accuracy of their system is very encouraging and it can be above 80%. However, the best accuracy they have achieved has come at the 20% of the full dataset which counts to 200 sequences.

2.5 Summary

In this chapter, a comprehensive literature review was provided that covered each component in the proposed computational pipeline. Initially, literature review was focused on the researches on homology search and identifying potential genes for keratin digestion. However, none of the bioinformatics research could be found that this research can directly be compared with. Then a review and a comparison were provided on four main EST clustering algorithms that belongs to two generations. Then for the last part of the computation pipeline, a literature review was conducted on CpG islands prediction which was helpful to gain a better understanding of the different approaches available for CpG islands prediction and to identify the best suiting approach for this research's context.

Chapter 3 - Design

The intention of this chapter is to present the research design. It would describe the fundamental approach, components of the research diagram and then at the end, different evaluation methods that are expected to be used would be discussed.

3.1 Fundamental Approach

The main goal of this research is to find out whether there exist any potential undiscovered genes in zebrafish that have given it the capability to digest keratin for its scale eating behaviour.

For that, we use Expressed Sequence Tags (ESTs) of keratin-feeding clothes moths (*Tineola*) and keratin beetles (*Trox*) that were extracted during the digestion of keratin. Then we look for homology (similarity) search of them with the genome of zebrafish for the functionality of keratin digestion. The reason that such homology match obtained is claimed to be performing keratin digestion functionality, is that the known data that is matched against the genome are extracted when they were specially expressed to perform keratin digestion.

Zebrafish and keratin-feeding clothes moths (*Tineola*) and keratin beetles (*Trox*) are highly different organisms (fish and insects) which make gene prediction using a homology search a complex task. Hence our methodology builds a comprehensive pipeline for performing a homology search.

Motif discovery of expressed sequence tags is a significant component in this pipeline. Motifs are the recurring patterns that exist between ESTs. Thus, a single motif can represent a more generalized pattern that reflects a set of ESTs. Therefore using motifs instead of ESTs itself, enhances the accuracy of homology search [35].

Since motifs reflect a set of ESTs, the better approach is to perform a clustering of ESTs prior to obtaining motifs. Then the motif discovery process is conducted in each cluster of ESTs.

Then a homology search is conducted between these motifs and the genome of the zebrafish by aligning them together. Once homology matches are found, it should be further analyzed to see whether it is satisfying the properties of a gene in order to claim the match as a potential gene for keratin digestion.

3.2 Research Design Diagram

Above described approach pipeline is shown visually in the below design diagram shown in the Figure 3.1.

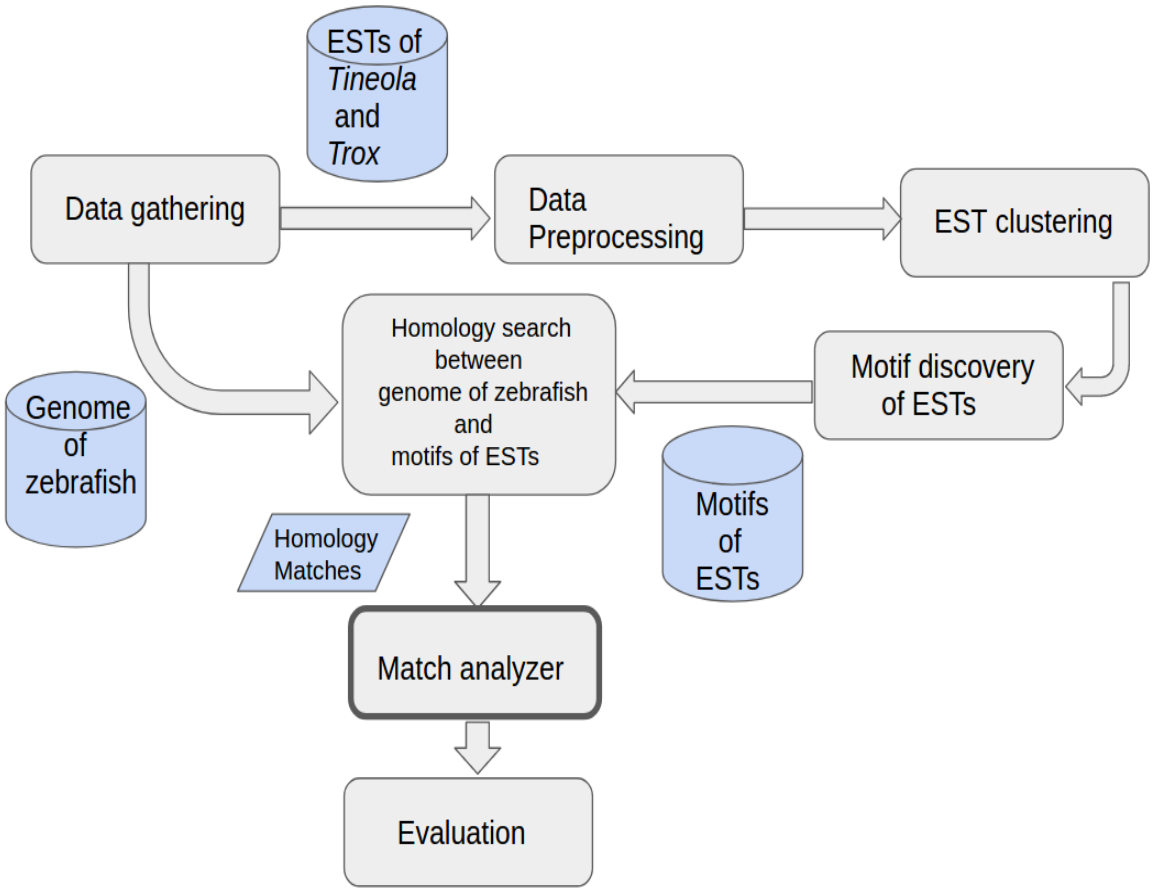


Figure 3.1: Overview of the research design

The design of the Match analyzer is described in section 3.3 and illustrated in Figure 3.2.

3.3 Homology Match Analyzer Design

Once a homology match is found, it brings the necessity to design an analyzer in order to analyze the match further and claim whether the found match is a potential gene or not. Design in Figure 3.2 explains the procedure how a homology match is claimed a potential keratin digestion gene or would simply have to discard the match as not a gene. As shown in the diagram the match is passed through a CpG-island predictor. The design of the CpG-island predictor would be described in the next section. As discussed in section 1.3.5 – 'CpG islands to identify genes', the importance of CpG islands is that they mark the starting region of a gene. CpG-island predictor would output whether the starting region of the match has properties of a gene, and if so the match is claimed as a gene and if not, the match is not a gene.

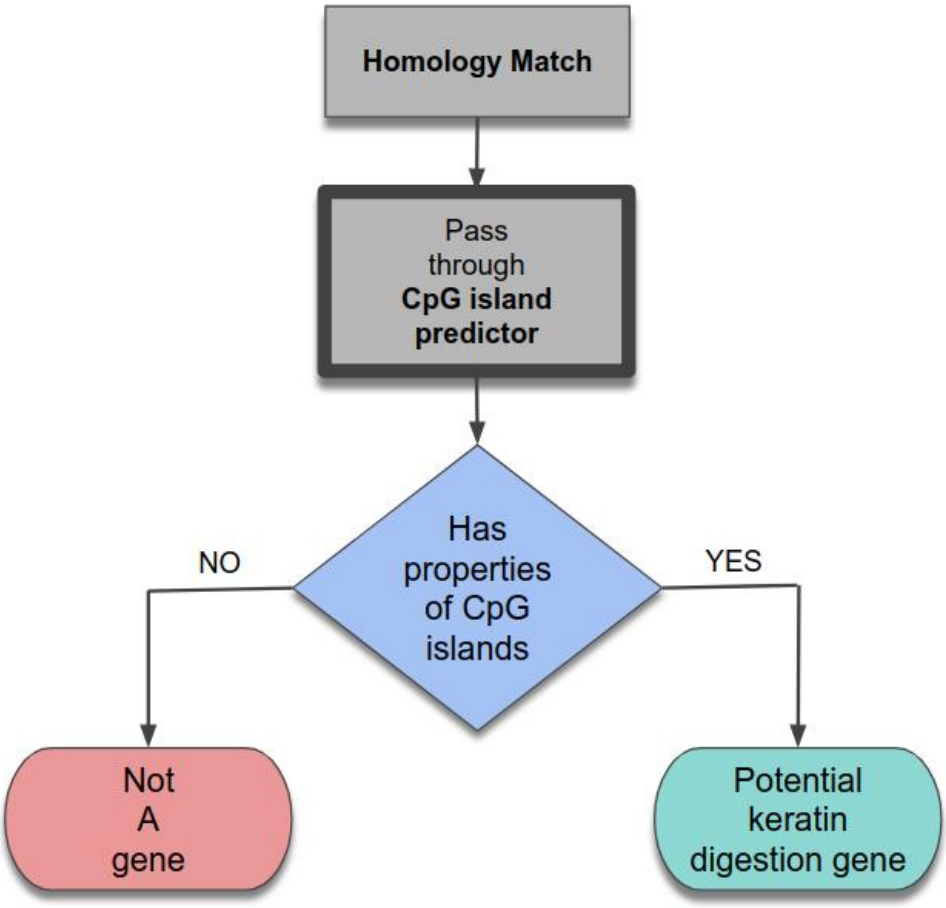


Figure 3.2: The design of Homology Match Analyzer.

3.4 Markov model for prediction of CpG islands

As described in the Homology Match Analyzer, it is very important to design a mechanism to further analyze a homology match before claiming it as a gene. For this, the concept of CpG islands is used. CpG islands mark the starting region of a gene. Therefore a CpG island predictor is designed in such a way that the starting region of the homology match can be passed through it in order to see whether the starting region satisfies the genomic properties of CpG islands.

As discussed in literature, in cutoff based method for CpG island has its inherent issue which is that neither a biological argument nor a formal statistical motivation was used in picking up these cutoff values which were based on a difficult to interpret scale the In the design of CpG island predictor, the natural properties of CpG islands were mapped to a computational model. Hence it is decided to use a Markov model for the prediction of CpG islands. The reason why a Hidden Markov model is not preferred, is that the nature of the available dataset does not support it. Dataset has two separate sets of CpG islands and non CpG islands and not a whole sequence annotated with the CpG island and non CpG islands.

CpG islands are consist of letters {A, T, C, G} but their transition probability are different from the normal other sequences of the genome that involves having higher C to G is higher than the non CpG islands.

A Markov model is a finite state machine that changes from state to state at every time instance depending on a transition probability [41]. It was introduced by after the Russian mathematician Andrey Markov and named after him [42]. In CpG islands, a letter that occurs at a particular location of the sequence does not convey any information with regard to next location of the sequence that gives the ability to predict the coming letter of the following location [27]. Hence, CpG islands satisfy the conditional independence property of Markov models.

As such, Markov model for the prediction of CpG islands to identify genes is a natural approach to be tried out. In this research, CpG island prediction is seen as a Markov model problem. The advantage using Markov model instead of traditional algorithms to predict CpG islands is that, it outputs a probability score which then can be used to match against the statistical properties of a homology match in order to be claimed as a potential gene for keratin digestion.

Generally a Markov model is defined as a triplet (Q, p, A),

where,

Q is a finite set of states

p is the initial state probabilities

A is the state transition probabilities. Each state transition a_{st} for each s, t in Q is defined as in Equation 3.1,

$$a_{st} \equiv P(x_i = t \mid x_{i-1} = s) \tag{3.1}$$

A Markov model should satisfy the property that conditional probability distribution of future states of the process depends only upon the present state, not on the sequence of states that preceded the present state [42].

In this study, prediction of CpG islands is considered as first order Markov chain problem. In a first order Markov chain, its current state $x_i$should only depend on its previous state x_{i-1}.

That is, $$P(x_i \mid x_{i-1}, x_{i-2}, \ldots, x_3, x_2, x_1) = P(x_i \mid x_{i-1}) \tag{3.2}$$

Hence, probability of a sequence X is defined as in Equations 3.3 and 3.4

$$P(X) = P(x_L, x_{L-1}, \ldots, x_3, x_2, x_1) \tag{3.3}$$

$$= P(x_L \mid x_{L-1}) * P(x_{L-1} \mid x_{L-2}) \ldots * P(x_2 \mid x_1) * P(x_1) \tag{3.4}$$

where L is the total length of the sequence X

The proposed Markov model for the CpG islands prediction of zebrafish uses sequence DNA letters { A, T, C, G} as the states. Figure 3.3 shows the state transition diagram of the proposed Markov model. Whenever one letter is followed by another letter, the proposed Markov model sees it as a state transition from the first letter state to the next letter state. For an example, let the sequence be “ATT”, when it is fed to the Markov model, it would see it as a state transition from state A to state T and then a state transition from state T to itself.

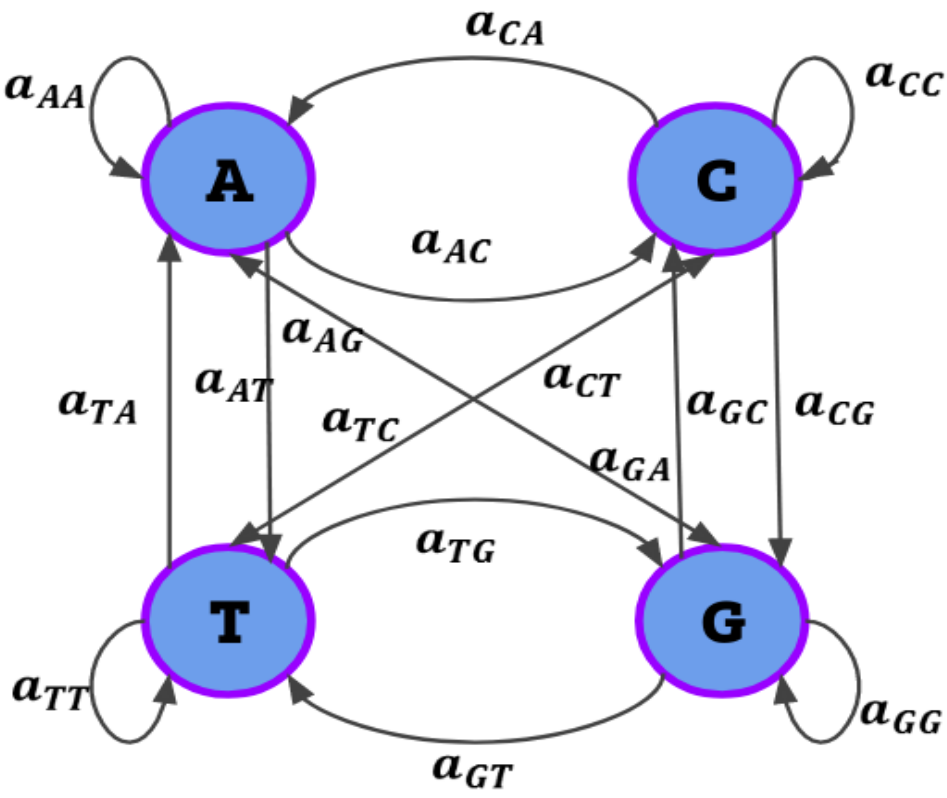


Figure 3.3 State transition diagram of the proposed model

3.4.1 Sub Models

The CpG island predictor employs two sub models: cpg model and non-cpg model

cpg model would be used for the transitions that happen **inside the CpG islands** and hence it would be trained with the **CpG islands** of zebrafish and

non-cpg model would be used for the transitions that happen **outside the CpG islands** and hence it would be trained with the **non CpG islands** of zebrafish.

State transition probabilities would be statistically defined as below for two sub models.

Let state transition probability from state $s_1$to s_2 within cpg model be $a^+_{s_1 s_2}$.

Then $a^+_{s_1 s_2}$would be as in Equation 3.5.

$$a^+_{s_1 s_2} = \frac{N^+{}_{s_1 s_2}}{\sum \quad N^+_{t_1 t_2}} \tag{3.5}$$

$N^+{}_{s_1 s_2}$ is the Number of times $s_1 s_2$ transitions happened inside the cpg islands

$\sum \quad N^+_{t_1 t_2}$is the total Number of all transitions happened inside the cpg islands

Let state transition probability from state $s_1$to s_2 within non-cpg model be $a^-_{s_1 s_2}$.

Then $a^-_{s_1 s_2}$would be as in Equation 3.6.

$$a^-_{s_1 s_2} = \frac{N^-{}_{s_1 s_2}}{\sum \quad N^-_{t_1 t_2}} \tag{3.6}$$

$N^-{}_{s_1 s_2}$ is the Number of times $s_1 s_2$ transitions happened outside the cpg islands

$\sum \quad N^-_{t_1 t_2}$is the total Number of all transitions happened outside the cpg islands

3.4.2 Decode a sequence

Once these two models are trained, then it is ready to decode a given sequence to see whether the given sequence has the properties of cpg islands or not. The given sequence should be passed through two models and the probability that it belong to either cpg model or non-cpg model is analyzed.

$$Score\ Value\ (x)\ = \frac{P(x \mid cpg\ model\ )}{P(x \mid non - cpg\ model\ )} \tag{3.7}$$

As in Equation 3.7, if the higher probability comes when the sequence x is in the cpg model, then that sequence is decoded as a cpg island and otherwise it is decoded as a non cpg island. Then log value of above $Score\ Value\ (x)$ is taken as in Equation 3.8 so that the score value would be converted to a log scale and hence whenever a given sequence is a CpG island, model would output a positive value and if otherwise, it would give a negative value.

$$Log\ Odd\ Score\ (x)$$
$$= log\ \frac{P(x \mid cpg\ model\ )}{P(x \mid non\ - cpg\ model\ )} \tag{3.8}$$

$$= \sum_{l=0}^{L-1} log\ \frac{a^{+}_{l,l+1}}{a^{-}_{l,l+1}} \tag{3.9}$$

L in Equation 3.9, is the total length of sequence $x$that is passed through the Markov model to decode. As in Equation 3.9, accumulated log ratio should be calculated for all the possible transitions in the sequence and if $Log\ Odd\ Score\ (x)\ > 0$, x is classified as CpG island and if otherwise, x would be a non CpG island.

3.5 Evaluation

For the evaluation of gene prediction researches in bioinformatics, different statistical measurements are used. The statistical measurements that we would be using are discussed in next sections.

3.5.1 Similarity Matrix Score

Similarity matrix score is obtained from the alignment matrix where two sequences are aligned based on the matching of one character to the other. The scoring mechanism uses gap penalties for mismatching gaps. So this numerical value represents the how strong two sequences are aligned together hence higher the score is, higher homology is claimed [14-15].

3.5.2 P-value

P-value is defined as the probability that a random sequence (with the same length and conforming to the background) would have position p-values such that the product is smaller or equal to the value calculated for the sequence under test.

The position p-value is defined as the probability that a random sequence (with the same length and conforming to the background) would have a match under test with a score greater or equal to the largest found in the sequence under test [43].

3.5.3 E-value

E-value that is associated to a score S is the number of distinct alignments, with a score equivalent to or better than S, that is expected to occur in a dataset search by chance. The lower the E value, the more significant the score is,

$$E = (n \times m) / (2^{S'}) \tag{3.10}$$

Equation 3.10 shows the equation for the E-value calculation where n is the total number of residues the database and m is the length of the query sequence where S' is the Score [45].

3.5.3 Percent identical residues

The basic and most frequently used method to measure similarity is percent identical residues [13]. In simple terms, percent identical residues are taken as R on R -> 1, R on K -> 0 and then the identity is indicated as a percentage of total alignment [9].

3.5.4 Evaluation of prediction model for CpG islands

For the evaluation purpose of the model to predict CpG islands would be measured in terms of accuracy, sensitivity, specificity and precision. Possible outcomes of the model can be depicted in a confusion matrix as shown in Figure 3.4

True positive (TP) would be CpG island being predicted as a CpG island

True negative (TN) would be non CpG island being predicted as a non CpG island

False positive (FP) would be non CpG island being predicted as a CpG island

False negative (FN) would be CpG island being predicted as a non CpG island

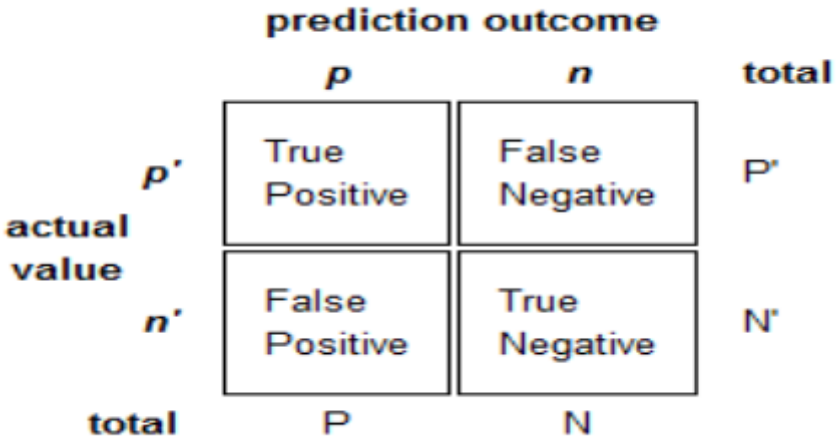


Figure 3.4: Confusion Matrix

Accuracy = (TP + TN)/ (P + N)

Sensitivity = (TP)/ (TP + FN)

Specificity = (TN)/ (FP + TN)

Precision = (TP)/ (TP + FP)

3.6 Summary

This chapter presented the research design with design diagrams for the computational pipeline, for the design of homology match analyzer and for state transition diagram of the proposed model. After describing the fundamental approach of the design, a detailed description of each component of the computational pipeline was provided by further breaking down the internal components as well. Further, the mathematical derivations that were needed for the development of CpG island predictor that uses Markov model were also presented. The chapter is concluded by stating the evaluation methods that were used to evaluate the final research outcomes.

Chapter 4 - Implementation

Through this chapter, the implementation details of the research is discussed at a finer level of detail, down to the code level. Each step in the proposed pipeline is selected and how they are actually implemented in order to obtain the expected results are presented in this chapter.

4.1 Data Collection

As stated in Chapter 1, this research is based on publicly available data. For this research three types of datasets were required which are,

(i) EST dataset of keratin-feeding clothes moths (*Tineola*) and keratin beetles (*Trox*)

(ii) Complete Genome of zebrafish

(iii) CpG islands dataset of zebrafish

Expressed sequence tags (ESTs) of keratin-feeding clothes moths (*Tineola*) and keratin beetles (*Trox*) are taken from the research done by J. Hughes et al. [8] and those ESTs have been submitted to GenBank [36]. GenBank is an online database that contains publicly available nucleotide sequences that are submitted from individual research laboratory experiments to large-scale sequencing projects [39].

The complete genome of zebrafish is needed in this research and would also be taken from GenBank [36].

CpG island dataset is taken from haowu lab [40]. Haowu lab has collections of different biological sequences and it has CpG island datasets of different organisms including zebrafish as well. Since any sequence in the genome of zebrafish other than the CpG islands are non CpG

islands, no specific separately created dataset can be found for non CpG island of Zebrafish. Hence a code is implemented which randomly gets the sequences from the genome of zebrafish. (This code is made available in Appendix A.3). One can argue that taking random sequences can include CpG islands as well. Since all the CpG islands of zebrafish is not identified yet, such attempt on excluding CpG islands from randomly generated sequences would anyway fail. Further compared to the size of the genome of zebrafish (Genome of zebrafish is 1,464,443,456 characters long), it can be safely stated that getting a random number out of 1,464,443,456 and becoming it a CpG island is unlikely and as such, the effect is highly negligible.

4.1.1 Sizes of the datasets

- EST dataset carries 920 ESTs of keratin-feeding clothes moths and 883 ESTs of keratin beetles.
- Zebrafish genome is 1,464,443,456 Base Pairs long
- CpG island dataset of zebrafish contains 326 sequences where 116 are CpG islands and 210 are non CpG islands.

Out of CpG island dataset, 77 sequences are kept for the training dataset 39 sequences are kept as the testing dataset.

Out of non CpG island dataset, 140 sequences are kept for the training dataset 70 sequences are kept as the testing dataset.

4.2 Preprocessing of Expressed sequence tags

Expressed sequence tags (ESTs) are extracted from laboratory experiments after undergoing through a huge chemical procedure [8]. Hence there can be ESTs with low quality. The quality of ESTs affect the final results in a great deal [21] it is essential to filter out the low quality ESTs. For the prepossessing of ESTs, SeqTrim tool [32] is used which would trim the low quality ESTs and the ESTs that are repeated using repeat masking.

4.3 Clustering of Expressed sequence tags

Expressed sequence tags are sequences of letters {A, T, C, G}. They do not carry a numerical value or other attribute values as such. Therefore clustering of ESTs is not a straightforward operation. But for the next step of discovering motifs in building the path for homology search, clustering of ESTs are essential as it results in obtaining stronger motifs [21].

It has been provided an extensive literature survey on EST clustering algorithms in this dissertation under section 5.3 - Expressed sequence tags clustering. From that, the following factors were able to be identified that led to choose wcd (wcd is the name given to the algorithm which is pronounced as wicked) EST clustering [22] in this research for the clustering ESTs.

- wcd algorithm provides an all-versus-all comparison which is required in the context of this research as all the ESTs are equally important with regard to the biological experiment that was conducted to extract them
- d2_cluster [19], as discussed in literature is a widely used algorithm but wcd is an improved version of d2_cluster
- Only wcd provides the ability for user to try out different user inputs in order to determine the better clusters out of different clusters provided
- The disadvantage of the algorithm which is the inability for parallelism does not stand against this research's context,

Hence wcd algorithm is decided to be used in this research.

4.4 Motif Discovery and obtaining homology matches

Since we have performed a clustering of ESTs in the previous stage, discovering motifs in each cluster results in stronger motifs. This is due to the reason that a cluster contains ESTs of similar origin and hence when motif discovery operation is performed, that does not try to generalize a recurrence pattern by force to an EST that is totally different to this similar set of ESTs which otherwise would result in low quality motifs. For the purpose of motif discovery, we would be using MEME [34, 44] algorithm.

Once the motifs are obtained for clusters of ESTs, then homology matches should be looked for. For that, the genome of zebrafish and motifs would be aligned using pairwise local alignment. In this study, BLAST - Basic Local Alignment Search Tool [48] is used, which is the most popular and most widely used local alignment tool in bioinformatics [46, 47]. Then the resultant homology matches should be through the match analyzer to analyze it further before claiming it as a potential gene for keratin digestion in scale eating.

4.5 Homology match analyzer design

Once a homology match is found, it is passed through the match analyzer to further analyze it further. It would follow match analyzer procedure explained in section 3.3 – 'Homology Match Analyzer Design' and a homology match is claimed as a potential keratin digestion gene or would not be classified as a gene based on the output given by the CpG island predictor.

4.6 CpG island predictor

The design of CpG island predictor which is used to predict the CpG islands in zebrafish is extensively described in “Section 3.4 Markov model for prediction of CpG islands”. This research has used python 3.5 to implement the design of the CpG island predictor.

Below code segment implements the Markov model according to the design that was described earlier. ***transCountMatrix*** carries the full transitions count for each possible transition where ***transMatrix*** has the state transition probabilities for all states. (Note: Full code implementation is available in Appendix A)

```
def calcMarkov(transMatrix, transCountMatrix, path, start, end):

  for i in range(start, end):

    workfile = path + str(i) + ".fasta"

    with open(workfile, 'r') as readfile:
      seq = readfile.read().replace('\n', '')
      for j in range(0,len(seq)-1):
        for k in range(4):
          for m in range(4):
            if seq[j]+seq[j+1] == COMBINATIONS[k][m]:
              transCountMatrix[k][m] += 1
  print(transCountMatrix)


  row_total = 0
  for i in range(0,4):


    for j in range(0,4):
      row_total += transCountMatrix[i][j]


  for i in range(0,4):
    for k in range(0,4):
      prob = transCountMatrix[i][k]/row_total
      transMatrix[i][k] = round(prob, 4)

return transMatrix
```

The function ***calcMarkov*** should be called for both CpG islands dataset and non CpG islands dataset as below to get the state transition probabilities of both cpg model and non-cpg model.

```
calcMarkov(transmatPlus,transmatPlusCount, filepath_cpg, start, end)
calcMarkov(transmatMinus,transmatMinusCount, filepath_noncpg, start, end)
```

Following code segment would implement the decode part of the CpG island predictor. ***log_ratio*** Function would output the log ratio value considering the value of the cpg model and non-cpg model. And ***get_log_value*** would take a sequence and for all of its possible transitions, log odd value would be calculated which was presented in Equation 3.9 in design stage.

```
def log_ratio(prev, curr):
  prev_column = LETTER_ORDER.index(prev)
  curr_row = LETTER_ORDER.index(curr)
  plus_val = transmatPlus[prev_column][curr_row]
  min_val = transmatMinus[prev_column][curr_row]

  if plus_val == 0 and min_val == 0:
    log_ratio_value = 0
  elif plus_val == 0:
    log_ratio_value = -2
  elif min_val == 0:
    log_ratio_value = 2
  else:
    ratio_value = plus_val/min_val
    log_ratio_value = log(ratio_value,BASE)
  return log_ratio_value

def get_log_value(seq):
  total = 0
  for i in range(1,len(seq)):
    if seq[i-1] in 'ATCG' and seq[i] in 'ATCG':
      total += log_ratio(seq[i-1],seq[i])
  return total
```

4.7 Summary

As previous chapter discussed on the design of the research, this chapter is used to describe the actual implementation that was used to implement that design. Initially a comprehensive summary was provided on the data collections used for this research. Then the chapter continues to present implementation details on how the preprocessing of ESTs, clustering of ESTs and motif discovery from ESTs were done. Finally, the actual implementation of CpG isand predictor was described with its code level details.

Chapter 5 - Results and Evaluation

This chapter presents the results obtained along with their evaluation. Results achieved in each step of the proposed pipeline is discussed. At the end, the reasons behind the behaviour of the CpG island predictor is analyzed where finally a summary that compares the homology results along with a similar research is presented.

5.1 Preprocessing of Expressed Sequence tags (ESTs)

The dataset of Expressed Sequence tags of keratin-feeding clothes moths (*Tineola*) and keratin beetles (*Trox*) that are used in this study are taken from the research done by J. Hughes et al. [8]. As explained previously, they have first conducted a laboratory experiment and have then extracted the ESTs which were then submitted to GenBank database. They have claimed that they have preprocessed the ESTs before submitting to GenBank database.

But, since the results obtained from processing ESTs greatly depend on the quality of ESTs, we decided to carry on a preprocessing stage of our own to check the quality. For that we used SeqTrim tool [32]. Using that, we performed a filtering of low-quality sequences and repeat masking with parameters values set to defaults (min_insert_size_paired=40, min_quality=20, min_insert_size_trimmed=40). However, none of the low quality ESTs was detected, hence the total dataset of ESTs was passed to next stage.

5.2 Clustering of ESTs

By clustering ESTs, more similar sequences can be identified as clusters. Hence motifs become stronger when motifs are obtained from clusters as motifs are recurring patterns of sequences. As discussed in the implementation section, for the clustering of ESTs, wcd [22] EST clustering was used.

In wcd, windows sizes should be properly adjusted according to the dataset. For this, E - value which is described as a measurement of evaluation in “section 3.5.2 E-value” was used for obtaining best clusters. Hence clustering was performed for different window sizes which is a user determined parameter for wcd tool. For window sizes below 58 and above 251 all the sequences clustered into a one cluster and hence the range between 58 and 251 windows sizes were considered to evaluate their E-values. Figure 5.1 is a graph that is drawn with the value of window size against the log of E-value for the respective window size.

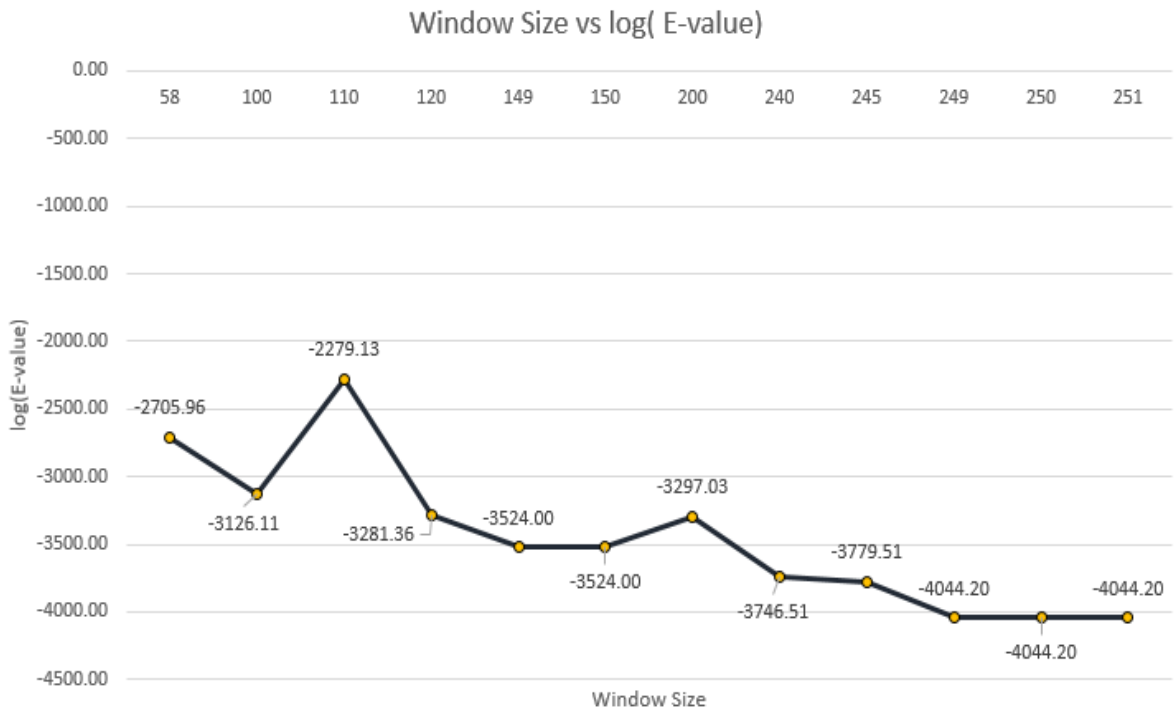


Figure 5.1 Window size vs Log(E value) graph

It can be seen that the graph is maintaining a flow that E-value was getting reduced when window size gets increased with an exception at the window sizes 110 and 200. The E-value gets to the minimum at 249 and remain constant. Hence we have selected the window size of 250 for the clustering of ESTs with wcd clustering. Once clustered, as the result, two clusters were obtained.

5.3 Motif discovery of ESTs

For the process of discovery of motifs, we used MEME algorithm [34, 44]. For the Cluster One, two motifs could be obtained. They were having E-values as low as 6.3e-4045 and 3.1e-3747 indicating how strong the obtained two motifs are.

However, MEME only accepts only up to the clusters with length of 60,000 bp but the Cluster one is of length 76985 bp. As an alternative to MEME, when it comes to larger datasets, DREME algorithm can be used but it inherently looks for short motifs [35]. As a result of that, when DREME algorithm was used for that cluster, we ended up having many small motifs as too small as 5 lengths long and further all of them were having very high E-values such as 2.3e-157, 6.3e-086, 1.3e-002, compared to the MEME algorithm which outputs a motif with an E-value of 6.3e-4045 for the Cluster One. Hence we decided that continually using MEME algorithm for Cluster Two as well would be highly beneficial compared to DREME algorithm. When wcd tool clusters the ESTs it sorts the sequences based on the similarity. Based on that similarity order, we divided the cluster into two parts so that they have lengths which are in acceptable range for MEME algorithm. Finally we could obtain two motifs with E-values which are as low as 4.6e-6456 and 2.3e-5589 supporting for the decision we made.

Figures 5.3, 5.4, 5.6, 5.8 show a graphical representation of obtained motifs to show repetition percentages of each letter and Figures 5.5, 5.7, 5.9 show the locations of where each motif has occurred in the set of ESTs. (Note Appendix B gives a bigger view of motifs with rest of the locations of motifs)

They are represented graphically having,

X - axis: Letter number of the motif (starting from 1 to length of the motif)

Y- axis: Scale 0 – 2 (The height of the letter is drawn in this scale, representing the number of times a particular letter has occurred in ESTs of a cluster. Hence, if a letter reaches the tallest position that means that letter has occurred in all the ESTs of that cluster). Figure 5.2 shows a clear view of the two axes used.

Figure 5.2: Zoomed-in of the two axes and how letter representation work

Cluster 1

Motif 1

Figure 5.3: Letter representation of Motif 1

Motif 2

Figure 5.4: Letter representation of Motif 2

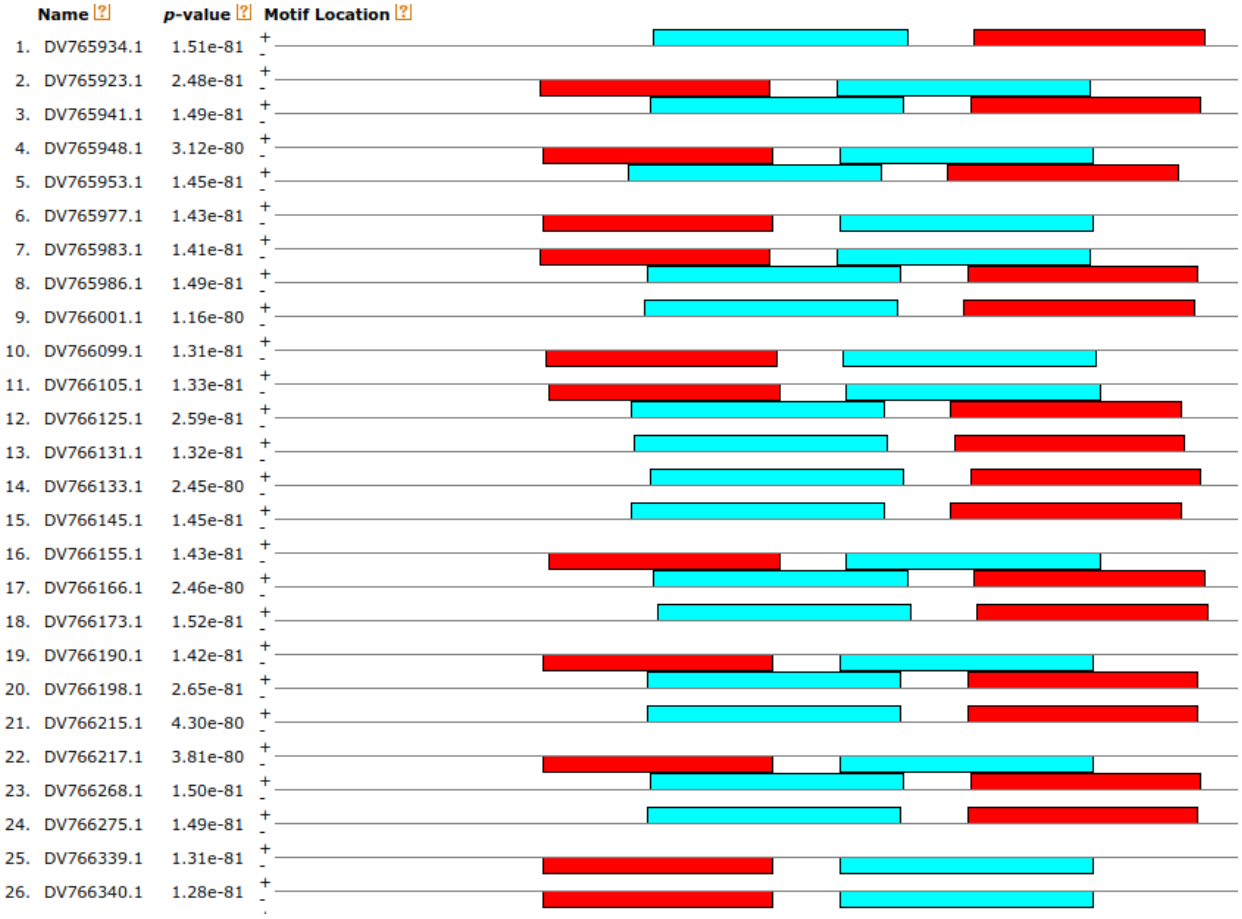


Figure 5.5: Locations of Motif 1 and 2 in ESTs with the respective e-values

Cluster 2

Part 1

Motif 3

Figure 5.6: Letter representation of Motif 3

	Name	p-value	Motif Location
1.	DV766885.1	4.42e-54	+ -
2.	DV766880.1	4.10e-54	+ -
4.	DV766889.1	2.65e-53	+ -
5.	DV766890.1	4.10e-54	+ -
6.	DV766891.1	2.62e-53	+ -
7.	DV766893.1	4.48e-54	+ -
8.	DV766898.1	4.78e-52	+ -
9.	DV766899.1	4.48e-54	+ -
11.	DV766902.1	4.12e-54	+ -
12.	DV766903.1	7.26e-53	+ -
13.	DV766905.1	4.50e-54	+ -
14.	DV766907.1	2.02e-52	+ -
15.	DV766909.1	4.12e-54	+ -
16.	DV766910.1	4.50e-54	+ -
17.	DV766911.1	4.10e-54	+ -
18.	DV766912.1	3.38e-52	+ -
19.	DV766914.1	2.45e-53	+ -
20.	DV766916.1	4.10e-54	+ -
21.	DV766918.1	3.76e-53	+ -
22.	DV766922.1	2.63e-53	+ -
23.	DV766925.1	2.63e-53	+ -
24.	DV766926.1	3.39e-54	+ -
25.	DV766927.1	4.42e-54	+ -
27.	DV766931.1	4.12e-54	+ -
28.	DV766935.1	4.12e-54	+ -
29.	DV766936.1	4.10e-54	+ -
30.	DV766939.1	1.30e-54	+ -
31.	DV766940.1	4.08e-54	+ -
32.	DV766941.1	4.10e-54	+ -
33.	DV766942.1	4.08e-54	+ -
34.	DV766945.1	4.52e-54	+ -
35.	DV766948.1	4.08e-54	+ -
36.	DV766949.1	8.21e-54	+ -
37.	DV766950.1	4.48e-54	+ -
38.	DV766951.1	4.50e-54	+ -
40.	DV766956.1	4.07e-54	+ -
41.	DV766958.1	1.21e-52	+ -
42.	DV766963.1	2.65e-53	+ -
43.	DV766964.1	4.52e-54	+ -
45.	DV766969.1	4.50e-54	+ -
46.	DV766974.1	2.43e-53	+ -

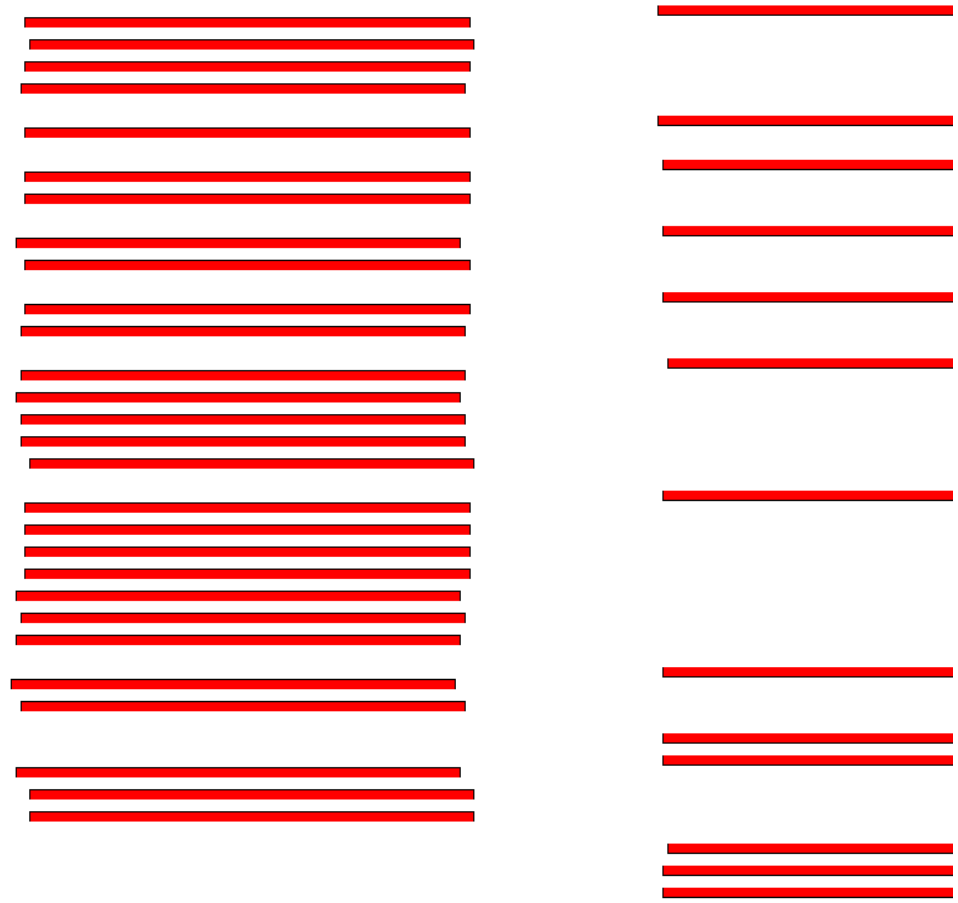


Figure 5.7: Locations of Motifs 3

Part 2

Motif 4

Figure 5.8: Letter representation of Motif 4

Figure 5.9: Locations of Motif 4 with the respective e-values

Table 5.1: E-value of each motif

Motif Number	E-value
Motif 1	3.1e-3747
Motif 2	6.3e-4045
Motif 3	4.6e-6456
Motif 4	2.3e-5589

Table 5.1 represents the E-values of all the motifs. Lower the E-value stronger the motifs are. When above E-values of motifs are analyzed, it can be seen that all the motifs obtained are strong representations of their respective clusters. Motif 3 is the strongest motif with respect to its cluster while other motifs are also having remarkably low E-values. Hence all these motifs are needed for an accurate alignment in the next step.

5.4 Alignment of Motifs with Zebrafish Genome

In this step, we aligned the obtained motifs along with the genome of zebrafish using BLAST - Basic Local Alignment Search Tool [48] as discussed in the implementation section, looking for the homology match with lowest E-value.

Finally, Motif 4 obtained a significant hit with the genome of zebrafish with the lowest E-value of 0.058. Table 5.2 summarizes the evaluation metric of the alignment result with the genome of zebrafish.

Table 5.2: Evaluation metric of alignment result

E-value	Score	Percentage Identity
0.058	41.0	93%

5.5 Visual Representation of Alignment

Figure 5.10 presents a visual representation of alignment. Two letter lines in the top are genome of the zebrafish starting at the location 14 410 and ending at location 14 440 whereas the bottom is motif aligned along it which is being outlined by a rectangular shaped box. The reason why two lines are there in the genome is that both negative and positive strand of the genome is given here. One is the complementary of the other. Out of two lines of the genome, below line is the positive strand and upper one is the complementary of it. In the motif, what is shown in red is the mismatches with letters in the genome.

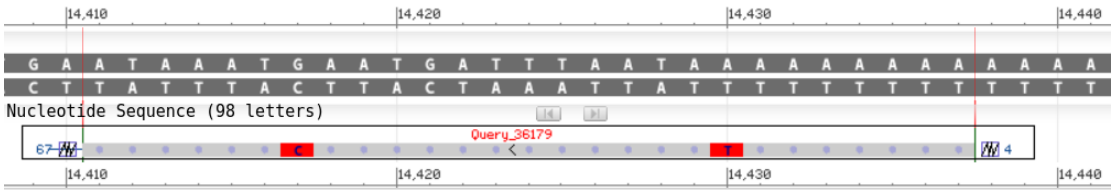


Figure 5.10: Alignment result

Table 5.3: BLAST Summary of alignment location of genome

Description	**Value**
Organism	*Danio rerio* (Zebrafish)
Chromosome	14
Clone	CH73-315I10
Region	14411..14437

Table 5.3 gives the important facts of the alignment in the genome as given from BLAST. The match has occurred in the 14th chromosome of zebrafish. Zebrafish has 25 chromosomes in its genome. The match has come from the genome library clone named CH73-315I10 starting at the location 14411 and ending at the location 14437.

5.6 CpG island predictor

The next task according to our design pipeline is finding whether the obtained homolog match is a gene or not. For that CpG island predictor has been trained so that it can then decode the starting region of the homology match obtained.

5.6.1 Training results of model for CpG island predictor: cpg sub model

After training the Markov model with CpG island dataset of zebrafish, transition probability matrix for cpg model which is shown in the Table 5.4 was obtained.

Table 5.4: State transition probability matrix for cpg model

cpg sub model	A	T	C	G
A	0.041	0.0256	0.0456	0.0718
T	0.021	0.0371	0.0578	0.062
C	0.062	0.069	0.1139	0.0717
G	0.0604	0.046	0.099	0.1161

5.6.2 Training results of model for CpG island predictor: non - cpg sub model

After training the Markov model for non CpG island dataset of zebrafish, the Table 5.5 transition probability matrix for non - cpg model was obtained.

Table 5.5: State transition probability matrix for non cpg model

non - cpg sub model	A	T	C	G
A	0.1058	0.0944	0.0584	0.0566
T	0.0823	0.112	0.0524	0.071
C	0.076	0.0573	0.036	0.0185
G	0.0511	0.0541	0.041	0.0332

If we analyze these two transition matrix, it can be clearly seen that the transitions that have happened in states C and G such as CC, CG, GG, GC are having higher value in the cpg model than in the non-cpg model successfully representing the features of CpG islands.

5.6.3 Evaluation Metrics for CpG island predictor

The evaluation metrics values presented in Table 5.6 was obtained using the testing dataset of CpG and non CpG islands of zebrafish.

Table 5.6: Evaluation Metrics

	Percentage Value
Accuracy	93.52%
Sensitivity	89.74%
Specificity	95.65%
Precision	92.11%

According to these values in Table 5.6, it can be said, without a doubt, that CpG island predictor shows high performance. However, the model has a relatively low sensitivity of 89.74% compared to the specificity which is 95.65%. This indicates that the model is more capable of correctly predicting a non CpG island than a CpG island. Two possible reasons for this behaviour of CpG island predictor can be,

1. **The dataset for non CpG island is approximately as twice larger as the dataset of CpG island.** As described earlier, non CpG islands dataset contains 210 sequences (93,810 bp) where CpG island dataset has only 116 sequences (58,914 bp). Hence the non-cpg sub model has gained more opportunity to be trained with more data and get fitted more accurately for the properties of non CpG islands than that of cpg sub model.

2. A few CpG islands may have naturally failed to show a clear difference against the non CpG islands and **such biological sequences may not adhere to the statistical properties that an artificial model tries to fit them to.**

5.6.4 Decoding the Starting region of the homology match obtained

As the final step of the designed pipeline, the starting region of the homology match is passed through CpG island predictor to be got decoded as a CpG island or otherwise. As we have identified in the literature, 200 bp length definition is used as the length of CpG islands which is needed to decide the length of the starting region sequence that should be passed through the CpG island predictor.

The match has started at location 14411 in the genome sequence. The sequence that starts at 14411-200 = 14211 and 200 long, is selected to be passed through CpG island predictor.

That sequence is,

>
acagtttaaagatatgcgctataggtgaattgaataaactaaattgttcattgtgtatgtgtgtgaataagtatgtatggatgtttcccagt
actgggttgcagctggaagggcatctgctgtgtaaaacatgctggataagttggcagttcattccactgtggcaacccatgatgaataaa
ggggctaagggaaaatga

We could obtained an interesting observation when this sequence is decoded through the CpG island predictor. The sequence happened to be having CpG island properties as predicted by the model which suggests that the homology match we obtained has properties to be claimed as a potential keratin digestion gene in zebrafish. In other terms, we can also consider it as reaching the highest goal of gene prediction research in bioinformatics.

5.7 Summary

In this Chapter, the results obtained from implementation of the proposed design was discussed. The most significant achievement of this research is that at the end of proposed pipeline, we could find a potential keratin digestion gene of zebrafish. To summarize what we have done, as in gene prediction in bioinformatics that is based on the homology discovery, we started with a known EST dataset for keratin digestion and once a homology was found, it was then evaluated with the statistical measure of E-value. Such homology is claimed to performing the same functionality as the first organism which is keratin digestion in this context.

We would like to commence the discussion of the results by comparing our result against closely related researches in the field. When it comes to computational prediction for keratin digestion, no other direct research could be found. The research done by J. Hughes et al. [8] is from which we took the EST data for keratin digestion. Their main focus laid on extracting gene expression in the gut of keratin-feeding clothes moths (*Tineola*) and keratin beetles (*Trox*) which turned it more towards a biological experiment. However, they have made a general homology search with proteases (some claim that keratin digestion enzymes are serine proteases and that was the reason to match against proteases) and have obtained percent identical residues of 22.4% for Tineola and 6.8% for Trox. Even though the percent identical residue values cannot be directly compared as they vary according the database sizes they are match against, they seem to get poorly low results as they have not built a comprehensive pipeline for homology discovery but have only performed the alignment step itself. On the other hand, like how we have predicted a potential gene, they anyway fail to predict a gene as they have not used genomes for their alignment other than presenting a homology match.

Chapter 6 – Conclusions

In this Chapter, the furthering understanding of the research objectives is explored with conclusions. Next, it would present the contributions done by the research and then limitations that were encountered were also analyzed. Chapter would commence the opportunities opened from this research for future researches.

6.1 Introduction

Keratin is an insoluble protein that is tightly packed which has resulted in having a great mechanical stability. Therefore keratin processing has become a burden to industries that involve keratin processing. Despite that, some organisms exhibit the natural capability of degrading keratin. This research is driven by this interesting natural observation. Particularly, it focuses on the keratin digestion capability in zebrafish that belongs to *danionian* group in the event of scale digestion as identifying responsible gene for keratin digestion can then lead to development of enzymes that vastly help keratin processing industries.

6.2 Conclusions about research objectives

The aim of this research was to explore whether there exists any potential undiscovered genes that have the capability of keratin digestion on scale eating in zebrafish. Hence a homology discovery is performed between the genome of zebrafish and the ESTs of keratin-feeding clothes moths and keratin beetles. As fish and insects are highly different organisms, this research builds a comprehensive pipeline for gene prediction fulfilling each and every objective we identified at the initial phase of the study.

The first objective of this research was *to obtain motifs in the clusters of expressed sequence tags of clothes moths and keratin beetles that perform the function of keratin digestion*. A

detailed literature survey was carried out on available EST clustering algorithms for selecting the best suiting algorithm for the ESTs in the context of this research. Using the MEME algorithm motifs were obtained.

The second objective was *to obtain homology match between above identified motifs and the genome of zebrafish for the function of keratin digestion.* This is a crucial one as finding such homology match largely contributes to the bioinformatics research field as none of the research could be found that utilizes a computational approach on keratin gene prediction. As a result of the comprehensive computational pipeline for gene prediction that was built, we could successfully obtain homology match with an E value that is as less as 0.058 despite the fact that the homology discovery was performed between two highly different organisms. Generally, highly different organisms show a lesser similarity between sequences that perform same functionality than the organisms of similar group as their phylogenetic and morphological structures have evolved separately.

Design and develop a model to analyze a homology match in order to claim it as a potential undiscovered gene was the third objective. The model that was developed utilizes the CpG island properties of the starting region of the genes. Sequence analysis of CpG island prediction has been seen as a Markov problem and the model works with an accuracy of 93.5% indicating the suitability of using Markov models for the CpG island prediction in zebrafish. Finally, the model predicted the starting region of the homology match obtained, having properties of CpG island of zebrafish making the finding more intriguing.

Hence it can be concluded that the homology match we found is a potential gene for keratin digestion in scale eating. Nevertheless, as in any gene prediction research, laboratory experiment should be conducted in order to bring the found gene from “computationally predicted” state to the “biologically verified” state.

6.3 Contributions

One of the major contributions made through this research is the computational prediction of potential keratin digestion gene in zebrafish. Other than that, this research significantly contribute to the research field of bioinformatics and computer science as to the best of our knowledge, this is the first attempt on,

1. Computational gene prediction for keratin digestion

2. Homology discovery performed on lepidophagous behaviour of zebrafish and

3. Development of a Markov model for CpG island prediction in zebrafish

As discussed in the literature, with regard to the keratin digestion, only the biological experiments can be found that involves isolating keratinase but then again they are only limited to the microorganisms such as bacteria and fungi. Lepidophagous behaviour of danion group including zebrafish is examined biologically through analyzing their natural diet and hence genes responsible for that cannot be identified. Furthermore, CpG island properties of zebrafish have been compared against other organisms, but no model could be found in order to particularly predict the CpG islands of zebrafish.

6.4 Limitations and Implications for further research

One of the major limitations faced during the research is the availability of the datasets. For an example data driven approaches such as the built computational pipeline, CpG island prediction model could have been made further improved if more annotated data were available. But considering the rapid pace of biological sequences becoming available, more data required might become available in near future letting the opportunity to explore more on keratin digestion genes in different organisms.

Another limitation is that the model builds to check whether the obtained match is a gene or not, is only based on the CpG island properties of starting region of genes. But to more accurately predict the starting and ending location of a gene, different computational prediction models can be developed based on different features of the genes. One such research area is the machine learning approaches for the prediction of intron and exon in biological sequences to identify the starting and ending regions of genes.

Furthermore, the obtained potential keratin digestion gene has opened the gates for many biological researches as well. One obvious research suggestion is performing biological experiments on verifying the gene. Then the researches can be extended to develop the enzymes based on the finding which are essentially required for keratin processing industries.

Bibliography

[1] I. Sazima, “Scale-eating in characoids and other fishes,” Environ. Biol. Fishes, vol. 9, no. 2, pp. 9–23, 1984.

[2] M. M. McClure, P. B. McIntyre, and A. R. McCune, “Notes on the natural diet and habitat of eight danionin fishes, including the zebrafish Danio rerio,” J. Fish Biol., vol. 69, no. 2, pp. 553–570, 2006.

[3] J. Janovetz, “Functional morphology of feeding in the scale-eating specialist Catoprion mento.,” *J. Exp. Biol.*, vol. 208, no. Pt 24, pp. 4757–68, 2005.

[4] S. Dhara, P. Datta, P. Pal, and S. D. Sarkar, “Processing and Industrial Aspects of Fish-scale Collagen: A Biomaterials Perspective,” in *Marine Proteins and Peptides*, Chichester, UK: John Wiley & Sons, Ltd, 2013, pp. 589–629.

[5] S. Lehtinen, J. Lees, J. Bahler, J. Shawe-Taylor, and C. Orengo, “Gene function prediction from functional association networks using kernel partial least squares regression,” PLoS One, vol. 10, no. 8, pp. 1–14, 2015.

[6] Z. Wang, Y. Chen, and Y. Li, “A brief review of computational gene prediction methods,” Genomics Proteomics Bioinforma., vol. 2, no. 4, pp. 216–221, 2004.

[7] B. Basu and A. K. Banik, “Production of protein rich organic fertilizer from fish scale by a mutant Aspergillus niger AB 100 __ A media optimization study,” *J. Sci. Ind. Res.*, vol. 64, pp. 293–298, 2005.

[8] J. Hughes and A. P. Vogler, “Gene expression in the gut of keratin-feeding clothes moths (Tineola) and keratin beetles (Trox) revealed by subtracted cDNA libraries,” *Insect Biochem. Mol. Biol.*, vol. 36, no. 7, pp. 584–592, 2006

[9] B. Rost, J. Liu, R. Nair, K. O. Wrzeszczynski, and Y. Ofran, “Automatic prediction of protein function,” Cell. Mol. Life Sci., vol. 60, no. 12, pp. 2637–2650, 2003.

[10] W. R. Pearson, “An introduction to sequence similarity (‘homology’) searching,” Curr. Protoc. Bioinforma., no. SUPPL.42, 2013.

[11] Q. Liao et al., “Large-scale prediction of long non-coding RNA functions in a coding–non-coding gene co-expression network,” Nucleic Acids Res., vol. 39, no. 9, pp. 3864–3878, May 2011.

[12] C. J. Bult et al., “Complete genome sequence of the methanogenic archaeon, Methanococcus jannaschii.,” Science, vol. 273, no. 5278, pp. 1058–73, Aug. 1996.

[13] W. Pearson and T. Wood, “Statistical significance in biological sequence comparison,” Handb. Stat. Genet., no. 804, 2001.

[14] S. R. Eddy, “a New Generation of Homology Search Tools Based on Probabilistic Inference,” Genome Informatics, vol. 23, no. 1, pp. 205–211, 2009.

[15] S. F. Altschul, R. Bundschuh, R. Olsen, and T. Hwa, “The estimation of statistical parameters for local alignment score distributions.,” Nucleic Acids Res., vol. 29, no. 2, pp. 351–361, 2001.

[16] L. Lange, Y. Huang, and P. K. Busk, “Microbial decomposition of keratin in nature-a new hypothesis of industrial relevance.,” Appl. Microbiol. Biotechnol., vol. 100, no. 5, pp. 2083–96, Mar. 2016.

[17] S. Hazelhurst, “Algorithms for clustering expressed sequence tags: the wcd tool,” South African Comput. J, pp. 1–14, 2008.

[18] G. Pertea et al., “TIGR gene indices clustering tools (TGICL): A software system for fast clustering of large EST datasets,” Bioinformatics, vol. 19, no. 5, pp. 651–652, 2003.

[19] J. Burke, D. Davison, and W. Hide, “d2_cluster: a validated method for clustering EST and full-length cDNA sequences,” Genome Res., vol. 9, no. 11, pp. 1135–1142, 1999.

[20] A. Ptitsyn and W. Hide, “CLU: a new algorithm for EST clustering.,” BMC Bioinformatics, vol. 6 Suppl 2, p. S3, 2005.

[21] S. H. Nagaraj, R. B. Gasser, and S. Ranganathan, “A hitchhiker’s guide to expressed sequence tag (EST) analysis,” Brief. Bioinform., vol. 8, no. 1, pp. 6–21, 2007.

[22] S. Hazelhurst, W. Hide, Z. Lipták, R. Nogueira, and R. Starfield, “An overview of the wcd EST clustering tool,” Bioinformatics, vol. 24, no. 13, pp. 1542–1546, 2008.

[23] B. Rost, J. Liu, R. Nair, K. O. Wrzeszczynski, and Y. Ofran, “Automatic prediction of protein function,” Cell. Mol. Life Sci., vol. 60, no. 12, pp. 2637–2650, 2003.

[24] H. Shu, T. Wildhaber, A. Siretskiy, W. Gruissem, and L. Hennig, “Distinct modes of DNA accessibility in plant chromatin.,” Nat. Commun., vol. 3, p. 1281, 2012.

[25] N. Dasgupta, S. Lin, and L. Carin, “Sequential Modeling for Identifying CpG Island Locations in Human Genome,” IEEE Signal Process. Lett., vol. 9, no. 12, pp. 407–409, 2002.

[26] H. Shu, T. Wildhaber, A. Siretskiy, W. Gruissem, and L. Hennig, “Distinct modes of DNA accessibility in plant chromatin.,” Nat. Commun., vol. 3, p. 1281, 2012.

[27] H. Wu, B. Caffo, H. A. Jaffee, R. A. Irizarry, and A. P. Feinberg, “Redefining CpG islands using hidden Markov models,” Biostatistics, vol. 11, no. 3, pp. 499–514, 2010.

[28] M. Gardiner-Garden and M. Frommer, “CpG islands in vertebrate genomes.,” J. Mol. Biol., vol. 196, no. 2, pp. 261–82, Jul. 1987.

[29] D. Takai and P. A. Jones, “Comprehensive analysis of CpG islands in human chromosomes 21 and 22,” Proc. Natl. Acad. Sci., vol. 99, no. 6, pp. 3740–3745, Mar. 2002.

[30] J. L. Glass et al., “CG dinucleotide clustering is a species-specific property of the genome,” Nucleic Acids Res., vol. 35, no. 20, pp. 6798–6807, Nov. 2007.

[31] R. A. Irizarry, H. Wu, and A. P. Feinberg, “A species-generalized probabilistic model-based definition of CpG islands,” Mamm. Genome, vol. 20, no. 9–10, pp. 674–680, 2009.

[32] J. Falgueras, A. J. Lara, N. Fernandez-Pozo, F. R. Canton, G. Perez-Trabado, and M. G. Claros, "SeqTrim: a high-throughput pipeline for preprocessing any type of sequence reads," BMC Bioinformatics, vol. 11, no. 1, p. 38, 2010.

[33] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," J. Mol. Biol., vol. 215, no. 3, pp. 403–410, Oct. 1990.

[34] T. L. Bailey, N. Williams, C. Misleh, and W. W. Li, "MEME: Discovering and analyzing DNA and protein sequence motifs," Nucleic Acids Res., vol. 34, no. WEB. SERV. ISS., pp. 369–373, 2006.

[35] T. L. Bailey, "DREME: Motif discovery in transcription factor ChIP-seq data," Bioinformatics, vol. 27, no. 12, pp. 1653–1659, 2011.

[36] "GenBank Home." [Online]. Available: https://www.ncbi.nlm.nih.gov/genbank/. [Accessed: 23-Jul-2017].

[37] "DNA Methylation | What is Epigenetics?" [Online]. Available: https://www.whatisepigenetics.com/dna-methylation/. [Accessed: 28-Jul-2017].

[38] R. Kakumani, M. O. Ahmad, and V. Devabhaktuni, "Prediction of hot-spots in protein sequences using statistically optimal null filters," in 10th IEEE International NEWCAS Conference, 2012, pp. 121–124.

[39] D. A. Benson et al., "GenBank," Nucleic Acids Res., vol. 41, no. D1, pp. D36–D42, Nov. 2013.

[40] "Hao Wu @ Emory Biostat." [Online]. Available: http://www.haowulab.org/. [Accessed: 08-Nov-2017].

[41] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, vol. 3, no. 1, pp. 4–16, 1986.

[42] Z. Copur, Handbook of Research on Behavioral Finance and Investment Strategies. 2015.

[43] T. L. Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers.," Proceedings. Int. Conf. Intell. Syst. Mol. Biol., vol. 2, pp. 28–36, 1994.

[44] "Introduction - MEME Suite." [Online]. Available: http://meme-suite.org/. [Accessed: 17-Nov-2017].

[45] C. A. Kerfeld and K. M. Scott, "Using BLAST to teach 'E-value-tionary' concepts," PLoS Biol., vol. 9, no. 2, pp. 1–4, 2011.

[46] T. A. Tatusova and T. L. Madden, "BLAST 2 Sequences, a new tool for comparing protein and nucleotide sequences," FEMS Microbiol. Lett., vol. 177, no. 2, pp. 187–188, May 1999.

[47] Y. Chen, W. Ye, Y. Zhang, and Y. Xu, "High speed BLASTN: An accelerated MegaBLAST search tool," Nucleic Acids Res., vol. 43, no. 16, pp. 7762–7768, Sep. 2015.

[48] "BLAST: Basic Local Alignment Search Tool." [Online]. Available: https://blast.ncbi.nlm.nih.gov/Blast.cgi. [Accessed: 10-Nov-2017].

[49] M. Lan *et al.*, "CpG-Discover : A Machine Learning Approach for CpG Islands Identification from Human DNA Sequence," Int. Joint Conf. on Neural Net., pp. 1702–1707, 2009.

Appendix A: Code Listings

A.1 Building of the Markov Model

```
LETTER_ORDER= ['A', 'T', 'C', 'G']
LETTER_SET = []

transmatPlus = [
[0, 0, 0, 0],
[0, 0, 0, 0],
[0, 0, 0, 0],
[0, 0, 0, 0]
]

transmatPlusCount = [
[0, 0, 0, 0],
[0, 0, 0, 0],
[0, 0, 0, 0],
[0, 0, 0, 0]
]

transmatMinus = [
[0, 0, 0, 0],
[0, 0, 0, 0],
[0, 0, 0, 0],
[0, 0, 0, 0]
]

transmatMinusCount = [
[0, 0, 0, 0],
[0, 0, 0, 0],
[0, 0, 0, 0],
[0, 0, 0, 0]
]
COMBINATIONS = [
['AA', 'AT', 'AC', 'AG'],
['TA', 'TT', 'TC', 'TG'],
['CA', 'CT', 'CC', 'CG'],
['GA', 'GT', 'GC', 'GG']
  ]

filepath_cpg = "/home/research/DV766842–DV767724/cpg/cpgDataset/"
filepath_noncpg = "/home/research/DV766842–DV767724/cpg/noncpgDataset/"
```

```
def calcMarkov(transMatrix, transCountMatrix, path, start, end):

  for i in range(start, end):

    workfile = path + str(i) + ".fasta"

    with open(workfile, 'r') as myfile:
      seq = myfile.read().replace('\n', '')
      for j in range(0,len(seq)-1):
        for k in range(4):
          for m in range(4):
            if seq[j]+seq[j+1] == COMBINATIONS[k][m]:
              transCountMatrix[k][m] += 1
  print(transCountMatrix)

  row_total = 0
  for i in range(0,4):

    for j in range(0,4):
      row_total += transCountMatrix[i][j]


  for i in range(0,4):
    for k in range(0,4):
      prob = transCountMatrix[i][k]/row_total
      transMatrix[i][k] = round(prob, 4)


  return transMatrix

print(calcMarkov(transmatPlus,transmatPlusCount, filepath_cpg, start, end))

print(calcMarkov(transmatMinus,transmatMinusCount, filepath_noncpg, start, end))
```

A.2 Decode a given sequence using log odd ratio

```
from math import log


LETTER_ORDER= ['A', 'T', 'C', 'G']
BASE = 10
filepath_cpg = "/home/research/DV766842–DV767724/cpg/cpgDataset/"
filepath_noncpg = "/home/research/DV766842–DV767724/cpg/noncpgDataset/"

def log_ratio(prev, curr):
  prev_column = LETTER_ORDER.index(prev)
  curr_row = LETTER_ORDER.index(curr)
  plus_val = transmatPlus[prev_column][curr_row]
  min_val = transmatMinus[prev_column][curr_row]


  if plus_val == 0 and min_val == 0:
    log_ratio_value = 0
  elif plus_val == 0:
    log_ratio_value = -2
  elif min_val == 0:
    log_ratio_value = 2
  else:
    ratio_value = plus_val/min_val
    log_ratio_value = log(ratio_value,BASE)
  return log_ratio_value



def get_log_value(seq):
  total = 0
  for i in range(1,len(seq)):
    if seq[i-1] in 'ATCG' and seq[i] in 'ATCG':
      total += log_ratio(seq[i-1],seq[i])
  return total
```

A.3 Generate non CpG islands dataset

```
workfile = "/home/research/zebrafishDataset/Danio_rerio.GRCz10.dna.chromosome.10.fa"
write_path = "/home/research/DV766842–DV767724/cpg/noncpgDataset/"

LENGTH = 450
with open(workfile, 'r') as myfile:
  seq = myfile.read().replace('\n', '')

from random import randint

for i in range(1,211):
  write_file = write_path + str(i) + '.fasta'
  rand_start = randint(0, genome_length - 500)
  end = rand_start + LENGTH
  file = open(write_file, 'w')
  file.write(seq[rand_start:end])
```

A.4 Obtain false positives and false negatives for CpG island prediction

```
def run_dataset(filepath,range_start, range_end,tag):
  FP = 0
  FN = 0
  for i in range(range_start, range_end):
    workfile = filepath + str(i) + ".fasta"


    with open(workfile, 'r') as readfile:
      seq = readfile.read().replace('\n', '')



      if (tag == "cpg" and get_log_value(seq) < 0):
        print('wrong')
        FN += 1

      elif (tag == "noncpg" and get_log_value(seq) > 0):

        FP +=1
  if (tag == "cpg"):
    return FN
  elif (tag == "noncpg"):
    return FP
```

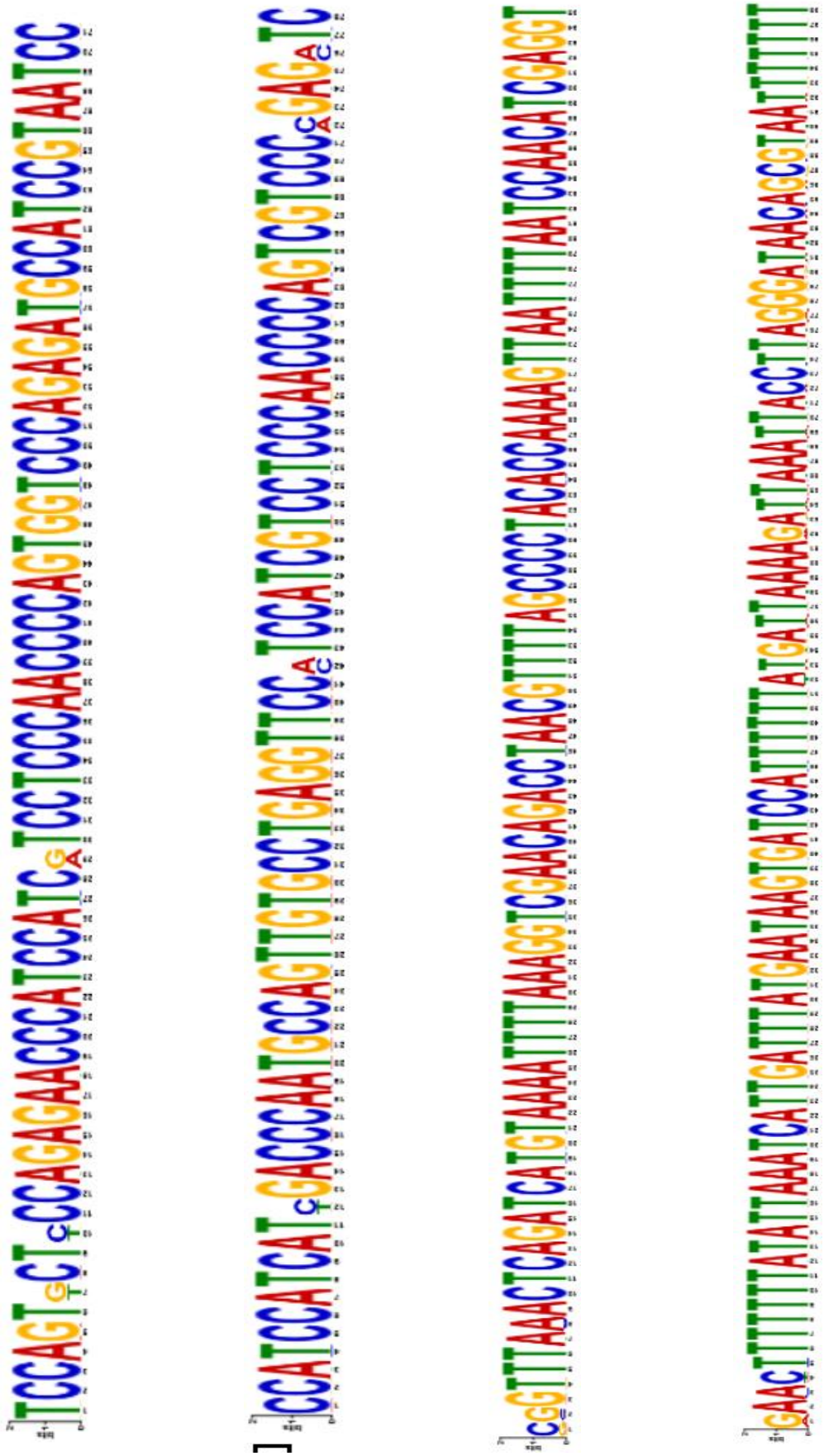
A.5 Obtain confusion matrix

```
def get_confussion_matrix(P, N, FP, FN):
  TP = P - FP
  TN = N - FN
  Accuracy = (TP + TN) / (P + N)
  Sensitivity = (TP) / (TP + FN)
  Specificity = (TN) / (FP + TN)
  Precision = (TP) / (TP + FP)

  print("Accuracy = ", Accuracy)
  print("Sensitivity = ", Sensitivity)
  print("Specificity = ", Specificity)
  print( "Precision = ", Precision)

P = 116 - 78
N = 211 - 141


get_confussion_matrix(P, N, FP, FN)
```

Appendix B: Figures

B.1 Motifs

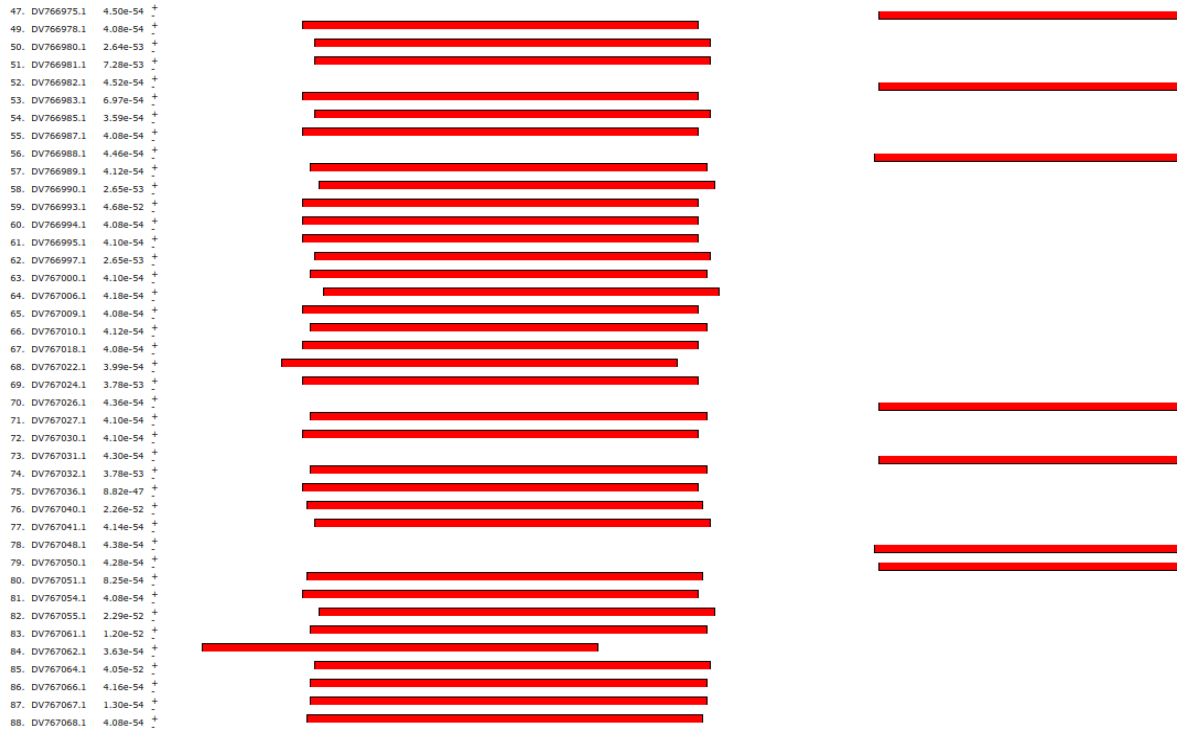
Four motifs are represented in the next page.



Motif 1 Motif 2 Motif 3 Motif 4

B.2 Locations of the motifs in Cluster One

B.3 Locations of the motifs in Cluster Two - Part 1



B.4 Locations of the motifs in Cluster Two - Part 2