Computational detection of gene regulatory signals in human genome sequence

A dissertation submitted to the University of Cambridge
for the degree of Doctor of Philosophy

by

Aroul Selvam Ramadass

Trinity Hall, University of Cambridge

and

The Wellcome Trust Sanger Institute, Hinxton, Cambridge

July 2004

DECLARATION

This dissertation is my own work and contains nothing which is the outcome of the work done in collaboration with others, except as specified in the text and acknowledgements.

The work in this thesis is not substantially the same as any I have submitted for a degree or diploma or other qualification at any other university.

Aroul Selvam Ramadass

July 2004

SUMMARY

Transcription, the first step in gene expression, is initiated from a transcription start site and terminated some distance downstream of the cleavage site. In this thesis I attempt to identify and model different regulatory signals involved in the process of transcription, towards the development of a signal based *ab initio* gene predictor.

First I attempt to identify regulatory signals in the sequence downstream of the cleavage site that may be responsible for transcription termination. Base compositional analyses reveal no significant bias in the nucleotide composition. An investigation based on free-energy minimisation Zuker algorithm indicates the possibility of a secondary structure in the sequence downstream of the cleavage site. A probabilistic machine learning algorithm based on Bayes theorem and Generalised Linear Models, Eponine, used to scan for motifs, learns a model to classify termination sites from other sequences. The model captures a few multiplex signals that might be responsible for polymerase II pause and termination. An evaluation of this termination model against annotated human chromosomes shows that the model performs better than existing methods. However a significant number of predictions also appear near the annotated start site of genes. Approximately 10% of predictions lie within genes and their density is correlated with gene length and intron size. I propose two hypotheses to explain these anomalies and discuss results from recent experiments.

Splicing is now found to be interlinked temporally and spatially with transcription and I attempt to develop a donor and acceptor site model using Eponine. Comparisons of the models with annotated sites show the models have higher positional accuracy and perform comparably with existing programs, GeneSplicer and StrataSplice.

Like transcription, translation machinery is influenced to a great extent by regulatory signals and I investigate them by scanning for motifs around translation start and stop sites using Eponine. The start model learnt only the regulatory elements and not the coding potential of exons. Despite this it performs better than the existing program NetStart, although less well than the program ATGpr.

The availability of these models creates the possibility to build an *ab initio* gene prediction program based purely on gene regulatory signals.

ACKNOWLEDGEMENTS

I thank my supervisor, Tim Hubbard, for giving an opportunity to work on this project in The Wellcome Trust Sanger Institute. I very much appreciate and am grateful to his advice and support he has given during this project. I thank Thomas Down for his help and the *Eponine* trainer that facilitated this study. Raphäel Lepläe and Mathew Pocock were grateful and patient in introducing me to the programming concepts. I thank them for this and all the help they gave me during the years I worked with them. Discussing ideas with Samiul Hasan, Yen-Hua Huang and Bernard Leong were always stimulating and I thank them for making the work interesting.

This project would not have been possible without the hard work of annotation and curation by members of Ensembl, VEGA and Sanger human Chromosome 22 group. I appreciate them for this excellent job and making the data available in the public domain.

I thank Rajkumar Sasidharan for all the support and collaboration he rendered in the ‘Domain Insertion’ project mentioned in the appendix. During my years of stay in Cambridge, I made numerous friends and it would be unfair if I mention only few. I thank every one of them for making my life enjoyable and happy.

I am grateful to Cambridge Commonwealth Trust and The Wellcome Trust Sanger Institute for funding and Trinity Hall College, Cambridge, for academic support.

I thank Sudhakaran Prabakaran, Jane E Swatton and James M Carr for correcting the manuscript.

Finally I would like to extend my gratitude to my parents, brother and sister for their love and support all these years.

CONTENTS

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
BP	Branch Point
BPS	Branch Point Sequence
CATH	Class Architecture Topology Homologous superfamily
CF	Cleavage Factor
CPF	Cleavage and Polyadenylation Factor
CPSF	Cleavage and Polyadenylation Specificity Factor
CstF	Cleavage stimulation Factor
CTD	Carboxy Terminal Domain
DRE	Downstream Regulatory Element
EAS	Eponine Anchored Sequence
EST	Expressed Sequence Tags
FP	False Positives
GFF	General Feature Format
GLM	Generalized Linear Model
GO	Gene Ontology
HMM	Hidden Markov Model
ISS	Intermediate Sequence Search
MAZ	Myc-Associated Zinc finger protein
NMD	Non-sense Mediated Decay
NN	Neural Networks
PABP	Poly(A) Binding Protein
PAP	Poly(A) Polymerase
PC	Position Constraint
PDB	Protein DataBase
PE	Pause Elements
RMSD	Root Mean Square Deviation
ROC	Range Operating Characteristics
RVM	Relevance Vector Machine
SCFG	Stochastic Context Free Grammar
SCOP	Structural Classification Of Proteins

SD	Shine-Dalgarno
SLBP	Stem Loop Binding Protein
snRNPs	small nuclear Ribo-Nucleo Proteins
SR	Splice Regulatory
SSP	Secondary Structure Profile
SVM	Support Vector Machine
TFBS	Transcription Factor Binding Sites
TP	True Positives
URE	Upstream Regulatory Element
UTR	Un-Translated Region
WM	Weight Matrix

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

1.1 Motivation

The last decade has seen a huge spurt of activity in genome sequencing. With improved technologies and reducing cost, more than 1000 viruses, 100 microbes and 11 eukaryotic whole genomes have been sequenced so far. Such a massive amount of data available in the public domain opens a whole array of possibilities to understand the mechanism of living organisms in detail. This revolution is likely to boost both the basic and applied science of various fields with opportunities for better food, health, and environment.

The highlight of all sequencing efforts is undoubtedly the announcement of the finished human genome sequence in summer 2003 by the International Human Genome Sequencing Consortium (IHGSC, 2001). This landmark achievement of a species reading its own genomic content is just the beginning rather than the end. Already progress is underway to tap this potential and understand the making and working of this complex organism. However, our current understanding is more limited and even defining complete functions of a single celled microorganism remains an uphill task. Nevertheless, recent high-throughput techniques, with supporting bioinformatics tools, have thrown out exciting results. Even complex human behaviours, like homosexuality and handedness are now linked to genes (Gibson and Dormor, 2003; Van Agtmael *et al.,* 2003). These are great surprises as scientists traditionally correlated these characters to environmental, social and cultural factors than genes. Such results emphasise the old genetic understanding that phenotype is the result of both genotype and environment even in complex human behaviours. Genotyping the expression of genes and their functions at molecular, cellular and physiological levels will answer such enigmatic questions in biology. This was emphasised again with the availability of two complete genomes – *Drosophila melanogaster* (Celniker *et al.,* 2002) and *Caenorhabditis elegans* (The *C. elegans* Sequencing consortium, 1998). Drosophila, having more complex developmental stage and nervous system, has fewer genes than the 1mm long soil nematode with only 959 cells in total.

To understand the functioning of organisms, it is necessary to know where and when a gene is expressed. The first step in this process is to identify the number of genes in the organism and map them in the genome. Unfortunately this has been a difficult task due to various issues such as intervening sequences (introns), pseudogenes and repetitive elements. In humans, we are still not clear about the exact number of genes. However research so far has helped to narrow down the number to around 30,000 (IHGSC, 2001). This is significantly lower than the 120,000 predicted sometime back (for discusssion, Ashurst and Collins, 2003; Ewing and Green, 2000; Liang *et al.*, 2000). Although the number might seem to be low for a complex organism, the number of transcripts produced from these genes is quite high as a result of alternative promoters, splicing and polyadenylation. In humans, it is estimated that an average of 2.5 alternative transcripts are produced per locus (Ashurst and Collins, 2003).

Until now, gene identification in the genomic sequence has been mainly focused on protein coding genes with less attention paid to pseudogenes, non-coding RNA genes and internal (embedded) genes. Non-coding RNA genes include an array of different types of regulatory RNA genes with newer types still appearing (Cawley *et al.*, 2004; Mattick, 2001).

Identifying, mapping and confirming the presence of these genes and different regulatory signals in the genomic sequence is referred to as *annotation*. This is done using an ensemble of different experimental and computational tools, with computational approaches usually facilitating the initial steps. Many gene prediction algorithms, such as Genewise (Birney and Durbin, 1997) rely on evidence from the alignment of EST, mRNA or protein sequences to the genome. Such algorithms generate accurate gene predictions, but only where expressed sequence data is available. Here, I am interested in *ab initio* methods that can predict from genome sequence alone. *Ab initio* gene prediction programs used in annotation can be broadly classified into comparative and non-comparative methods depending on whether they predict from an alignment of genome sequences or a single genome sequence. To date the majority of work was done using non-comparative *ab initio* algorithms and are based on different methods, namely neural networks (example programs include *GRAIL* (Uberbacher *et al.*, 1996), *GENEPARSER* (Snyder and Stormo, 1995)), discriminant analysis (*HEXON* (Solovyev *et al.*, 1995), *MZEF* (Zhang, 1997)) and hidden markov models (*GENSCAN* (Burge and Karlin, 1997)). Besides these, there are other old methods such as rule-based methods (*GENEID* (Guigo *et al.,* 1992), *GENEFINDER* (Wilson *et al.,* 1990)), linguistic methods (*GENLANG* (Dong and Searls, 1994)) and decision trees (*MORGAN* (Salzberg *et al.,* 1998)). Some programs were developed by combining different methods and *GENIE*, an example, combines hidden markov models and neural networks (Reese *et al.,* 2000). Few other *ab initio* gene prediction programs, like *QRNA*, were developed to detect non-coding RNA genes (Rivas *et al.,* 2001). Reviewing all these methods and programs is beyond the scope of this chapter and hence I refer the reader to these reviews (Mathe *et al.,* 2002; Zhang, 2002).

In general, *ab initio* gene prediction programs use sequence signals and coding measures to predict gene structures. Coding measure (a feature measured computationally but not used by the biological system) is the important component as it is likely to differentiate exons (coding sequences) from introns (intervening sequences). However, this limits the identification of pseudogenes and non-coding RNA genes and the performance of the gene prediction programs are poor even in simple cases (Rogic *et al.*, 2001). So, a gene prediction program based purely on DNA regulatory signals is likely to overcome this problem. Towards this future objective, I attempt to develop prediction models that can efficiently detect signals from genomic sequence context.

Before describing my research objectives, I devote the rest of this chapter to introduce the basics of gene structure, different regulatory signals in the DNA sequence and the process of transcription and translation.

1.2 An overview of gene structure

A typical higher eukaryotic protein coding gene, as depicted in Figure 1, has a defined promoter region with exons and introns splitting the transcription unit. Transcription initiates from a transcription start site and terminates a few hundred bases downstream of the cleavage site. Exon and intron boundaries are marked by the donor and acceptor splice site regions and on pre-mRNA maturation, introns get spliced out by the spliceosome complex. The 5’ cap and 3’ poly(A) tail added to the matured transcript play major roles in mRNA stability, export and translation initiation (Manley, 2002; Proudfoot *et al.*, 2002). Processed and stable transcripts, exported to cytoplasm, are translated by the translation machinery in the cytoplasm with start and stop codon acting as its signals. Traditionally, as noted in *in vitro* experiments, transcription, splicing, capping, polyadenylation, termination and export were considered to be independent of each other. However latest research suggests that all these processes occur co-transcriptionally with the carboxy-terminal domain (CTD) of RNA polymerase II playing a major role (for review see, Neugebauer, 2002; Proudfoot *et al.*, 2002).

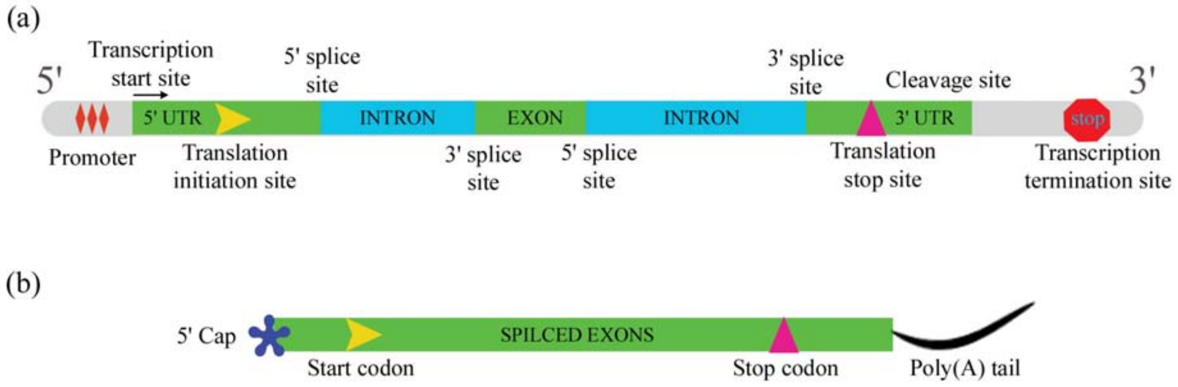


Figure 1. Schematic diagram showing (a) Typical gene structure of protein coding gene transcribed by RNA polymerase II (b) Matured RNA transcript with 5' cap and 3' poly(A) tail.

1.3 Defining transcription termination

Transcription termination has been defined to have two major steps: release of the transcript from the elongating polymerase and the dissociation of the polymerase complex from the DNA. An accurate and efficient system is required to pursue this function as the elongating polymerase would otherwise run-over into the adjacent transcription units. In yeast, many such cases have been reported in places where genes are closely spaced (Greger *et al.*, 1998). Also, terminating transcripts allow recycling of the polymerase and stops unnecessary transcription of intergenic regions. Various biological systems have been employed to understand this mechanism for many years now. All the results show termination can occur either depending upon bipartite or tripartite sequence components or on a stem-loop secondary structure basis. Here, I present a brief overview of the different termination systems identified so far.

1.4 Transcription termination in prokaryotes

Most prokaryotic genes do not have introns and the DNA is not isolated as prokaryotes do not have nucleus. Therefore, coupled transcription and translation is a common mechanism. Also, unlike eukaryotes, prokaryotic genes are transcribed by a single RNA polymerase.

Termination of transcription in prokaryotes is widely found to occur in two ways depending on the requirement of the protein factor, rho (reviewed in Henkin, 1996). In the 'intrinsic' or 'rho-independent termination', a G+C rich stem-loop structure followed by a series of U residues at the end of the transcript, hinders the proceeding polymerase and thus pauses, destabilizes and releases from the DNA (Figure 2). In 'rho-dependent termination', the protein factor, rho hexamer binds to a *rut* (rho utilization) site on the 3' end of the transcript. This RNA:protein interaction brings a change in the elongating polymerase resulting in the release of transcript and dissociation of polymerase by hydrolysing ATP as the energy source (Figure 3).

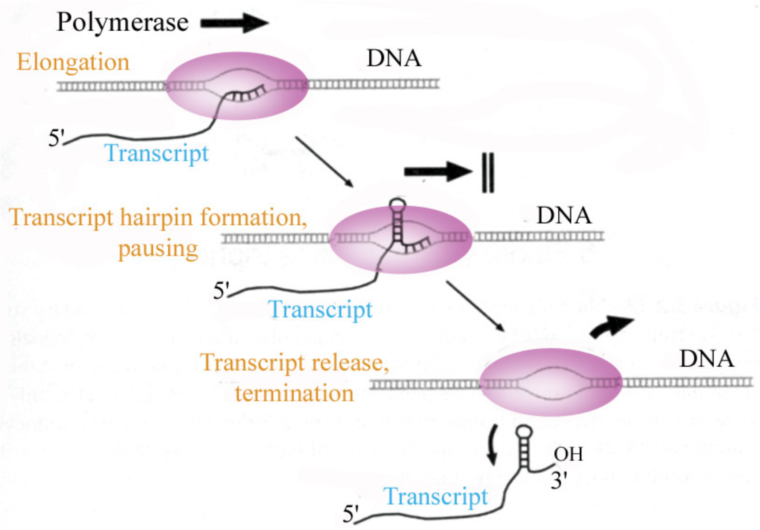


Figure 2. Rho-factor independent transcription termination in prokaryotes.

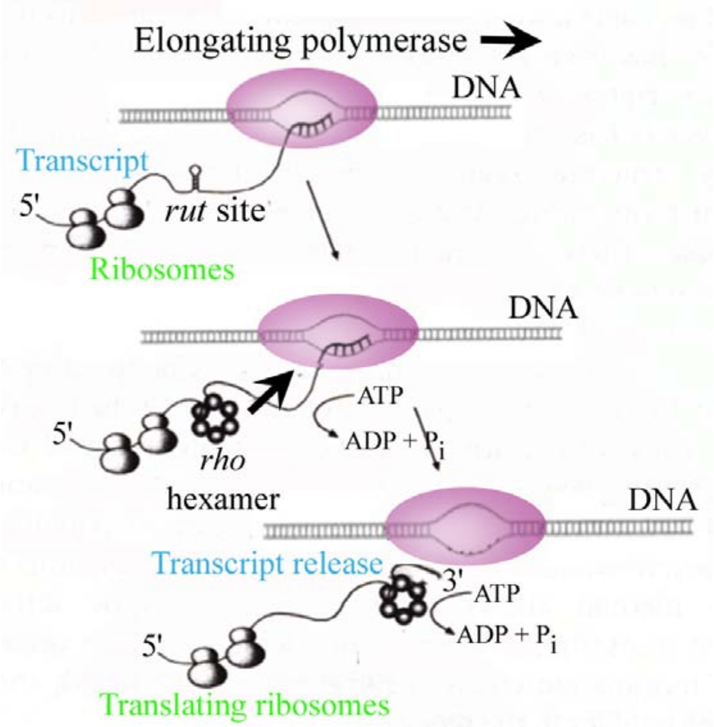


Figure 3. Rho-factor dependent transcription termination in prokaryotes.

However, in both mechanisms, termination is due to pausing of RNA polymerase at a specific site followed by destabilization of the complex due to the formation of a RNA:DNA hybrid in the transcription bubble and changes in the processivity of the polymerase (Henkin, 1996).

1.5 Transcription termination in eukaryotes

Unlike prokaryotes, eukaryotic transcription termination is complicated as there are three different types of polymerases responsible for transcribing various types of RNA molecules.

1.5.1 Polymerase I transcription termination

Transcription termination of Polymerase I, that syntheses rRNA, is mediated by protein factors reb1p in yeast (Lang *et al.*, 1994; Lang and Reeder, 1993) and TTF-I in mouse (Evers *et al.*, 1995). Polymerase I terminator sequence has two components: a binding site for the protein factor and an upstream element that codes for the last 10-12 nucleotides of the terminated transcript (Figure 4). The reb1p/TTF-I factor binds the DNA sequence element in the correct orientation and pauses the elongating polymerase. This halt stimulates the release of the transcript and dissociation of the complex. TTF-I is also found to recruit additional releasing factors for this process. However reb1p does not require any additional factors and the dissociation of the transcript depends only upon the instability of RNA:DNA hybrid in the active site of the polymerase due to stretches of A:U base pairing (Reeder and Lang, 1997).

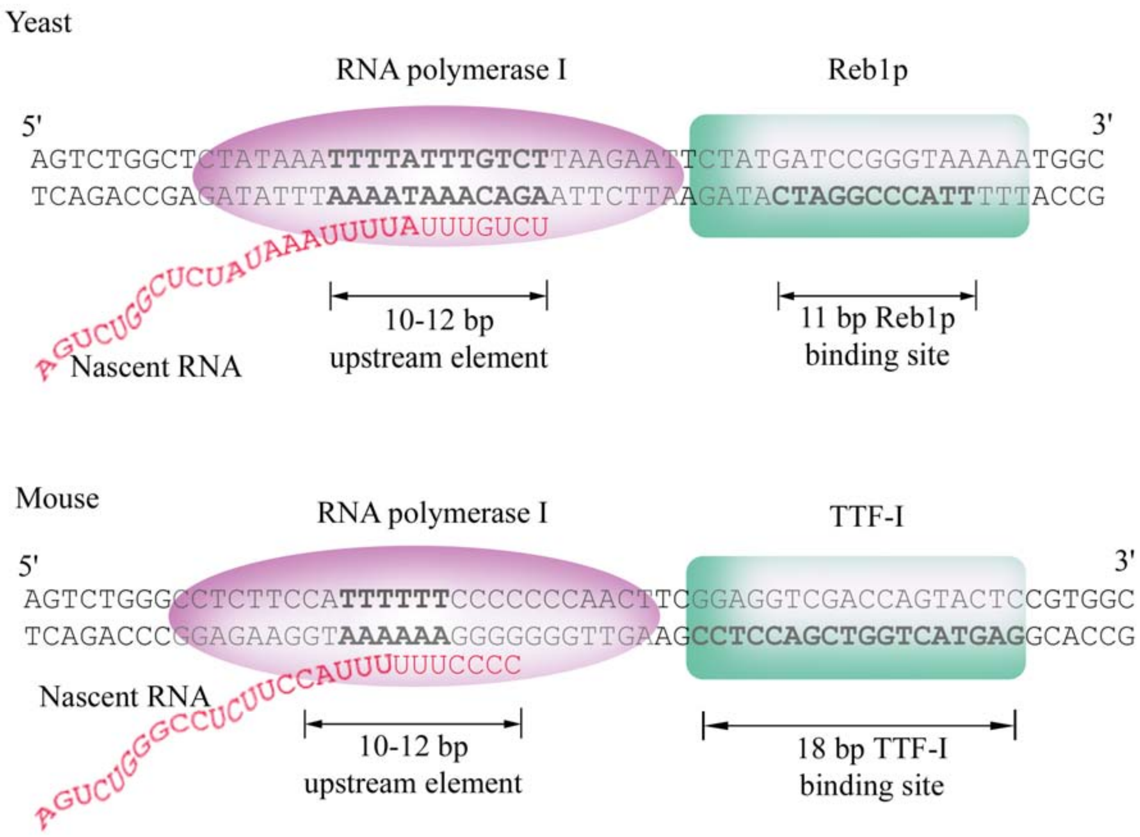

Yeast
RNA polymerase I
Reb1p
5'
3'
AGTCTGGCTCTATAAATTTTATTTGTCTTAAGAATTCTATGATCCGGGTAAAAATGGC
TCAGACCGAGATATTTAAAATAAACAGAATTCTTAAGATACTAGGCCCATTTTTACCG
AGUCUGGCUCUAUAAAUUUUAUUUGUCU
Nascent RNA
10-12 bp
upstream element
11 bp Reb1p
binding site
Mouse
RNA polymerase I
TTF-I
5'
3'
AGTCTGGGCCTCTTCCATTTTTTCCCCCCAACTTCGGAGGTCGACCAGTACTCCGTGGC
TCAGACCCGGAGAAGGTAAAAAAGGGGGGTTGAAGCCTCCAGCTGGTCATGAGGCACCG
Nascent RNA
AGUCUGGGCCUCUUCCAUUUUUUUCCCC
10-12 bp
upstream element
18 bp TTF-I
binding site

Figure 4. Structure of RNA polymerase I terminators from yeast and mouse.

Reb1p binding site was also found to have partial pausing activity for Polymerase II in the forward orientation and no activity in the reverse orientation (Lang *et al.*, 1994).

Polymerase I gene terminators are found to behave as DNA replication terminators as well. Bi-directional replication forks proceeding from the nearby *ori* site are stopped by the barrier created with TTF-I:DNA interaction. This barrier function is orientation dependent but has opposite polarity to transcription termination (Gerber *et al.*, 1997). However such a function is yet to be proved for yeast reb1p protein.

Polymerase I terminators are different from prokaryotic terminators as there are no inverted repeats and thus there is no hairpin structure formation and requirement of orientation-specific DNA binding proteins.

1.5.2 Polymerase III transcription termination

RNA Polymerase III responsible for transcription of tRNA, 5S rRNA and U6 snRNA can recognize termination sites accurately and efficiently without any requirement for protein factors (Cozzarelli *et al.*, 1983) and bring about termination with a simple cluster of four or more T residues (Bogenhagen and Brown, 1981). However, efficiency of release of paused polymerase was shown to improve with the recruitment of PTRF factor. Attempts to prove the requirement of La auto-antigen in Polymerase III transcription termination remains inconclusive (Lin-Marq and Clarkson, 1998; Maraia *et al.*, 1994; Yoo and Wolin, 1997).

1.5.3 Polymerase II transcription termination

RNA Polymerase II responsible for transcription of the remainder and vast majority of genes and is the subject of the work described in this thesis.

Polymerase II transcription termination occurs at least in three different ways depending on the gene it is transcribing, namely, snRNA and snoRNA genes, histone genes and protein coding genes. Before embarking into the details of these mechanisms, it is necessary to understand the 3’-end processing signals of protein coding genes.

The 3’-end processing involves an endonucleolytic cleavage of the nascent transcript and subsequent addition of poly(A) tail to the newly formed 3’-end. This process thought to occur for all transcribed genes along with capping and splicing of introns makes a nascent RNA matured. The 5’-cap and 3’-poly(A) tail have been found to have major roles in mRNA stability, export, translation initiation and other events. Endonucleolytic cleavage at the 3’-end of the transcript occurs at the cleavage site that has a consensus sequence of CA dinucleotide (Sheets *et al.*, 1990), flanked by a highly conserved poly(A) signal at the 12-30 bases at the upstream region and U-rich and or GU-rich motif immediately at the downstream site (Zarudnaya *et al.*, 2003) (Figure 5). In the majority of mammalian pre-mRNAs, the poly(A) signal is found to be composed of AAUAAA or AUUAAA

(MacDonald and Redondo, 2002) and has been suggested to be required for effective splicing (Cooke *et al.,* 1999) and transcription termination (Edwalds-Gilbert *et al.,* 1993; Yeung *et al.,* 1998) as well as for polyadenylation. The 160 kDa subunit of the cleavage and polyadenylation specificity factor (CPSF) binds to this hexamer element while the 64 kDa cleavage stimulation factor (CstF) binds to the U-rich sequence immediately downstream of the cleavage site. The binding of these factors is co-operative and each factor enhances the affinity of other factors towards its binding site. Once the processing site is recognized, two cleavage factors (CF I and CF II complex) get recruited and cleave the nascent transcript at the cleavage site. To the newly formed 3’-end, poly(A) polymerase (PAP) adds at least 250 nucleotides of adenine. Poly(A) binding proteins (PABP II) bind to this stretch of adenine nucleotides which enhances the stability of the tail. Although release of the transcript occurs after the cleavage at the cleavage site, RNA polymerase does not get released from the DNA at this site, but several hundred bases downstream.

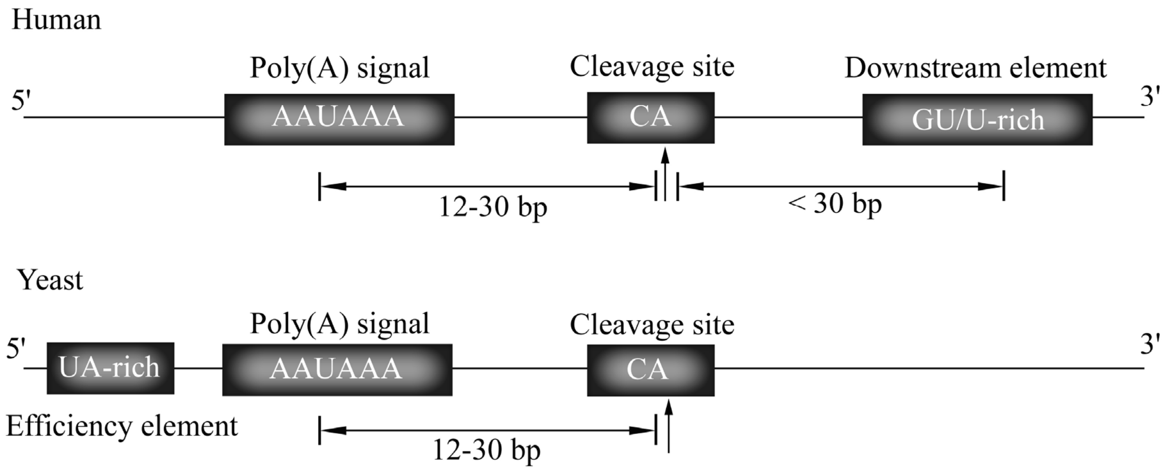


Figure 5. Schematic representation of 3'-end processing signals in human and yeast.

Determining the exact position of the polymerase release has been a challenge to study as the 3’-end product of the cleavage has very short half-life and the maturation of the 3’-end of the transcript (cleavage and polyadenylation) occurs co-transcriptionally. Fortunately, nuclear run-on assay can trap such nascent transcripts and help in analyzing transcription termination.

In the nuclear run-on technique, the nuclei transcribing a specific gene is isolated and allowed to incubate with radiolabelled ribonucleotide triphosphates for incorporation in the newly synthesized RNA molecules. This labeled nuclear RNA is then purified and hybridized to Southern blots of DNA probes carrying the gene sequence. The hybridization techniques equate directly to the polymerase density at the position of the probe and hence the point at which signal is no longer detectable corresponds to the site of termination. A gradual decrease in polymerase density always occurs downstream of the cleavage site. However in many instances before this decrease, a short higher polymerase density site is noticed. This is referred to as the *pause site*.

Now it is understood that transcription termination requires the 3’-end processing signals and a pause site. However 3’-end maturation does not require termination of the transcribing polymerase. In fact, both transcription termination and 3’-end processing processes are found to be coupled *in vivo* (Birse *et al.,* 1998; Dichtl *et al.,* 2002b) and are largely facilitated by the carboxy-terminal domain (CTD) of rpb1, the largest sub unit of Polymerase II.

Existence of pause site for termination has not been thoroughly accepted and there have been studies showing termination occurring without any requirement of pause site and with sole perturbation by poly(A) signal (Orozco *et al.,* 2002). However, several attempts have been made to identify consensus pause elements that are responsible to create a transient pause and thus enhance poly(A) signal recognition and termination.

Earlier studies identified an orientation-specific CCAAT element in the adenovirus late promoter that recruits CP1 protein and effectively terminates transcription from upstream genes (Connelly and Manley, 1989a, b). In yeast, Yhh1p, a subunit of CPF complex was identified to play this role (Dichtl *et al.,* 2002b).

In *Saccharomyces pombe*, both ura4 and nmt2 are found to possess downstream sequence elements that induce termination. These sequence elements are orientation specific and are composed of multiple and redundant signals. One of the sequence elements found in ura4 gene having pause activity, has two copies of pentanucleotide ATGTA with the last GTA playing an important role for binding an unknown factor responsible for pausing. However in the nmt2 gene, the pause elements are less compact and there is no homology with ura4 gene elements (Aranda and Proudfoot, 1999; Birse *et al.*, 1997). Similar pause sites were also found in α-globin genes and C2 and factor B genes (Yonaha and Proudfoot, 2000).

A detailed run-on assay in the mouse β-major globin gene identified a 69 bp AT-rich sequence that is active based on its position from the cleavage site (Tantravahi *et al.*, 1993). A similar experiment in human β-globin gene showed that a region 900 to 1600 bp downstream of the transcript cleavage site is essential for termination. Interestingly, it was also found that more cleavage of the nascent transcript occurs at this downstream termination region apart from the original cleavage site. These cleavages are termed as *co-transcriptional cleavage* and found to be necessary in addition to the 3' end processing signals for polymerase pause and release. However co-transcriptional cleavage was found to occur independent of 3' processing signals and thus deleting termination region does not affect 3' processing and vice versa. Nuclear run-on assay repeated on ε-globin genes found that the termination region is more diffuse than for the β-globin gene. Nevertheless the region is found to be as AT rich as the mouse globin gene, although the human region is longer (Dye and Proudfoot, 2001). Likewise, an A-rich 92 bp sequence at the 3' flanking region of human α2 globin gene is found to improve efficiency of upstream signals and thus processing events (Enriquez-Harris *et al.*, 1991).

Transcriptional studies in the intergenic region between human complement C2 and B genes showed the sequence element, GGGGGAGGGGG and the zinc-finger regulatory protein, MAZ that binds the sequence, can effectively stop transcription run-over from upstream genes and bring termination (Ashfield *et al.*, 1991). An upstream sequence element, mainly U-rich, was also found in human complement factor C2 and Lamin B2 gene (Moreira *et al.*, 1998).

Thus these experiments define various signals (CCAAT, ATGTA, AT-rich sequence, A-rich sequence and G-rich sequence) and factors (CP1, SP1 and MAZ) responsible for polymerase II transcription termination.

1.5.4 Computational detection of transcription termination signals

Apart from the experimental evidences mentioned above, a few related computational studies were also conducted around 500 bp upstream and downstream of cleavage and transcription start sites. Analysis on the cleavage site regions showed the common signals AATAAA and GT-rich sequence elements along with other signals. Prominent among them is CCCC, CCCTC and CCTCCC motifs. These motifs were also found peaking at -75 base pair and -200 to -100 bp upstream of transcription start site. Similarly the frequency of A_4 and G_4 motifs is higher before transcription start sites and after cleavage sites. Thus homo-oligomers $A_{4\text{-}5}$, $G_{4\text{-}5}$, $T_{4\text{-}5}$ and $C_{4\text{-}5}$, $C_{3\text{-}4}$ interspersed with T (CCCTC and TTCTT) and alternations of T and G (TGTGT) and GGAGG are found peaked around the 5’ and 3’ ends of genes (Nussinov, 1986a, b). Among all these signals, GTG/CAC and CTC/GAG DNA sequences are more interesting as they are frequently encountered in the regulatory DNA sequences and are likely target sites for several regulatory protein factors (Nussinov, 1986a). Another interesting result showed complementary signals on the same DNA strand have asymmetry behavior, i.e. the TGTGT peak patterns do not need to be the same for its complementary sequence, ACACA. This is more pronounced for complementary homo-polymers around transcription start site and cleavage site. This suggests some directionality in DNA bending and orientation-specific recognition by protein factors (Nussinov, 1986a). Similar signals were found in non-mammalian vertebrate DNA sequences as well (Nussinov, 1986b).

In another study (Nussinov, 1987) it was found that the distribution of the nucleotides showed opposite trends around the mammalian gene 5’ and 3’-ends i.e., R_6 motifs (stretch of 6 purine residues) are found more frequently before transcription start sites, whereas Y_6 motifs (stretch of 6 pyrimidine residues) occur less frequently. In the 3’ termini, Y_6 are less just before the end and R_6 motifs are more following it. In the non-mammalian vertebrate genes, these conditions are more pronounced. Two Y_6 peaks found at the 3’ termini might be due to poly(C) and poly(T) residues. The R_6 peaks in the gene upstream might be due to high concentration of AGGG and GGGC and to a lesser extent of A_4. This G runs might contribute to the bendability feature of the DNA molecule (Figure 6, reproduced from Nussinov, 1987).

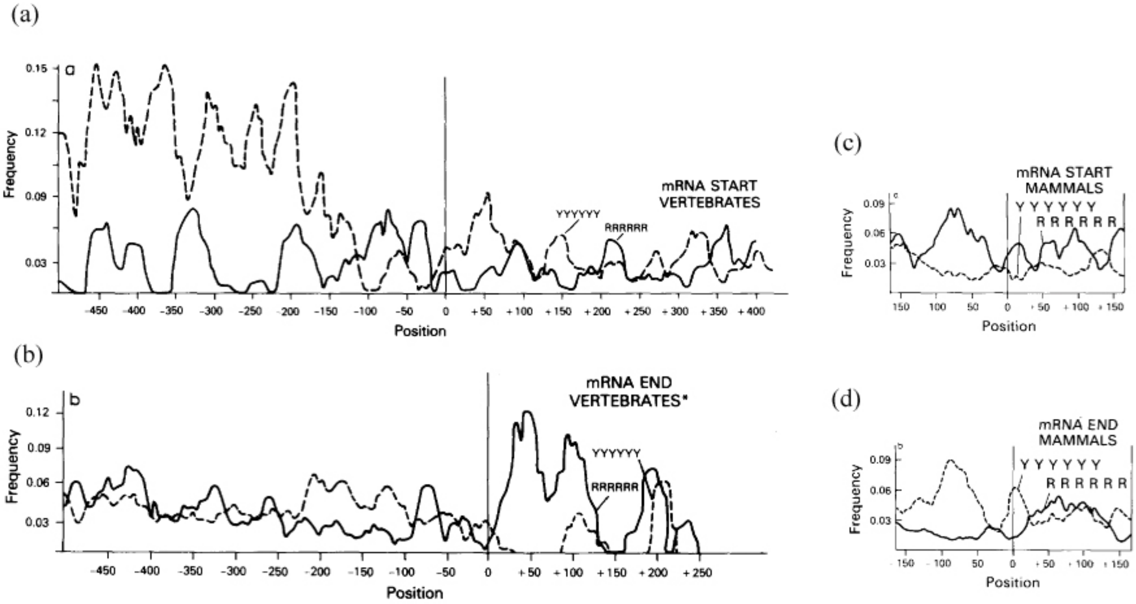


Figure 6. The nucleotide distribution of Y6 and R6 runs around transcription initiation and cleavage site. (a), (b) shows distribution in vertebrate mRNAs while (c) and (d) in mammals.

Although all these studies show that RNA polymerase II transcription termination signals are quite complicated, in general the system appears to work mainly based on two sequence components: 3'-end processing signal and pause sites. However these are not universal for all Polymerase II transcribed genes, as alternatives are found in histone and snRNA genes.

Histone genes are not spliced and the majority are not polyadenylated. The mature 3'-end of the transcript is formed by the endonucleolytic cleavage of the primary transcript and polymerase terminating in the A-rich sequence flanking the 3'-end (Briggs *et al.*, 1989). This cleavage is enacted by the stem-loop structure formed upstream of the cleavage site (roughly, 600 bp in case of H2A gene). The sequence at the stem-loops are well conserved with GGYYYU in the stem followed by a four-base loop, UYUN and the complementary sequence ARRRCC (Lanzotti *et al.,* 2002). Specialized protein factors called SLBP bind to this structure and stabilize the transcript mimicking the role of a poly(A) tail (Johnson *et al.,* 1986; Lanzotti *et al.,* 2002; Zanier *et al.,* 2002). The cleavage site efficiency is improved by a downstream element interacting with the U7 snRNP. Thus, both SLBP and U7 snRNP together recruit a complex capable of performing pre-mRNA processing reactions.

Polymerase II termination in snRNA genes require a 3' box element located 9-19 nt downstream of the end of the nascent transcript. U1 transcripts terminate just after the 3' box whereas U2 snRNA nascent transcripts are found up to 250 nucleotides downstream. These results show a 3’ box, a RNA processing signal and a downstream signal where interaction between protein-DNA leads to termination (reviewed in, Hernandez, 1992). HeLa cells with transfected constructs of 3’ box and downstream sequence elements confirmed the termination activity of these signals (Cuello *et al.*, 1999). This mechanism sounds similar to the mRNA bipartite termination process, requiring RNA processing signals and the termination elements.

However, poly(A) signal or 3’ processing signals are not essential for all cases of termination and recently it was reported in yeast that there is poly(A)-independent Polymerase II termination mechanism for snRNA and snoRNA genes with protein factors Nrd1 and Nab3 complex, Sen1 helicase and the CTD domain of Polymerase II (Steinmetz *et al.*, 2001).

1.5.5 Transcription termination models

Based on the available experimental evidence, three different models of Polymerase II transcription termination have been proposed.

(i) In the ‘RNA cleavage’ or ‘torpedo’ model, cleavage occurs firstly at the cleavage site, leaving two products: the upstream RNA later forming a matured transcript and a 3’ product still attached to the elongation complex. Rapid degradation of this 3’ product by the 5’→3’ exonuclease aided by helicase, ‘catches up’ the elongating polymerase and triggers termination (Proudfoot, 1989). However, recent evidences suggests cleavage is not necessarily required for termination to occur (Osheim *et al.,* 1999).

(ii) In the ‘polymerase change’ or ‘anti-terminator’ model, Polymerase II complex upon passage and recognition of poly(A) signal, undergoes conformational changes in the complex making it termination competent; this results in pause and release from the DNA (Logan *et al.*, 1987).

(iii) Recent experiments show that both these models are not mutually exclusive and a combination of both might exist (Proudfoot *et al.,* 2002). In the combined model, a co-transcriptional cleavage occurs at the downstream termination site first, with still interaction between the CTD of the polymerase and 3’ end processing signals remaining active. Subsequently, ‘polymerase change’ occurs in this interaction leading to cleavage at cleavage site and polymerase release.

1.6 Splicing and transcription

Recent experiments have clearly indicated that transcription and mRNA processing occur together and all the steps in the mechanisms are linked with each other, with the CTD of the RNA Polymerase II itself playing a major role. Therefore it is important to know about the splicing process, where intervening sequences were removed from the pre-mRNA to form mature mRNA, ready for translation.

1.6.1 Splicing mechanism

Exons and introns are determined by their boundary sequences with definite consensus patterns. Introns predominantly start with GT and end in AG dinucleotide. Figure 7 (reproduced from www.sanger.ac.uk/HGP/Chr22/cwa_archive/splice_site_analysis.shtml) shows the nucleotide distribution calculated from 3,673 introns from human chromosome 22. These predominant splice signals are called canonical splice sites and they form the basis for the GT-AG splicing rule (Mount, 1982). However, apart from the GT-AG rule, other intron boundaries, GC-AG and AT-AC, were also reported (Burset *et al.*, 2001). Also along with the splicing boundary signal, another consensus pattern called Branch Point Sequence (BPS) was found to be present upstream of AG dinucleotide and shown to be required for the splicing process.

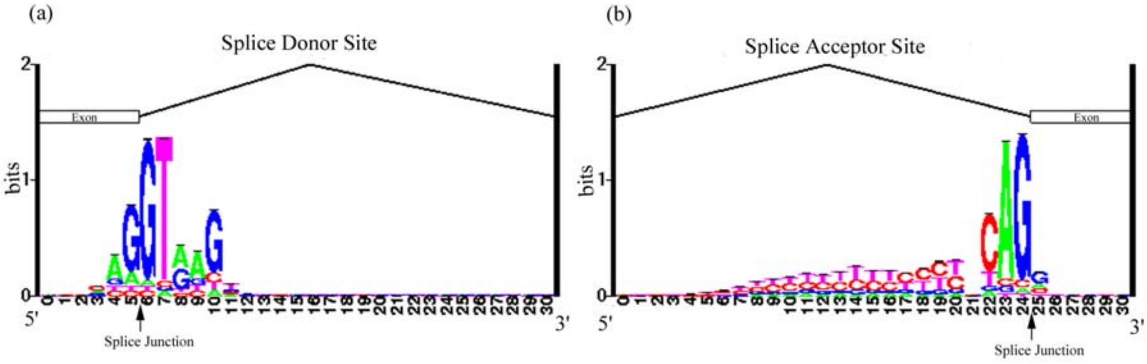


Figure 7. Nucleotide Distribution at Donor and Acceptor site analysed from 3,673 introns from human chromosome 22

Splicing of introns is mediated by a mega-Dalton RNA-protein complex formed with snRNA (small nuclear RNA) and around 50 to 100 protein molecules. The details of this complex mechanism is beyond the scope of this chapter, however, I will briefly cover some important aspects of the splicing process (Figure 8).

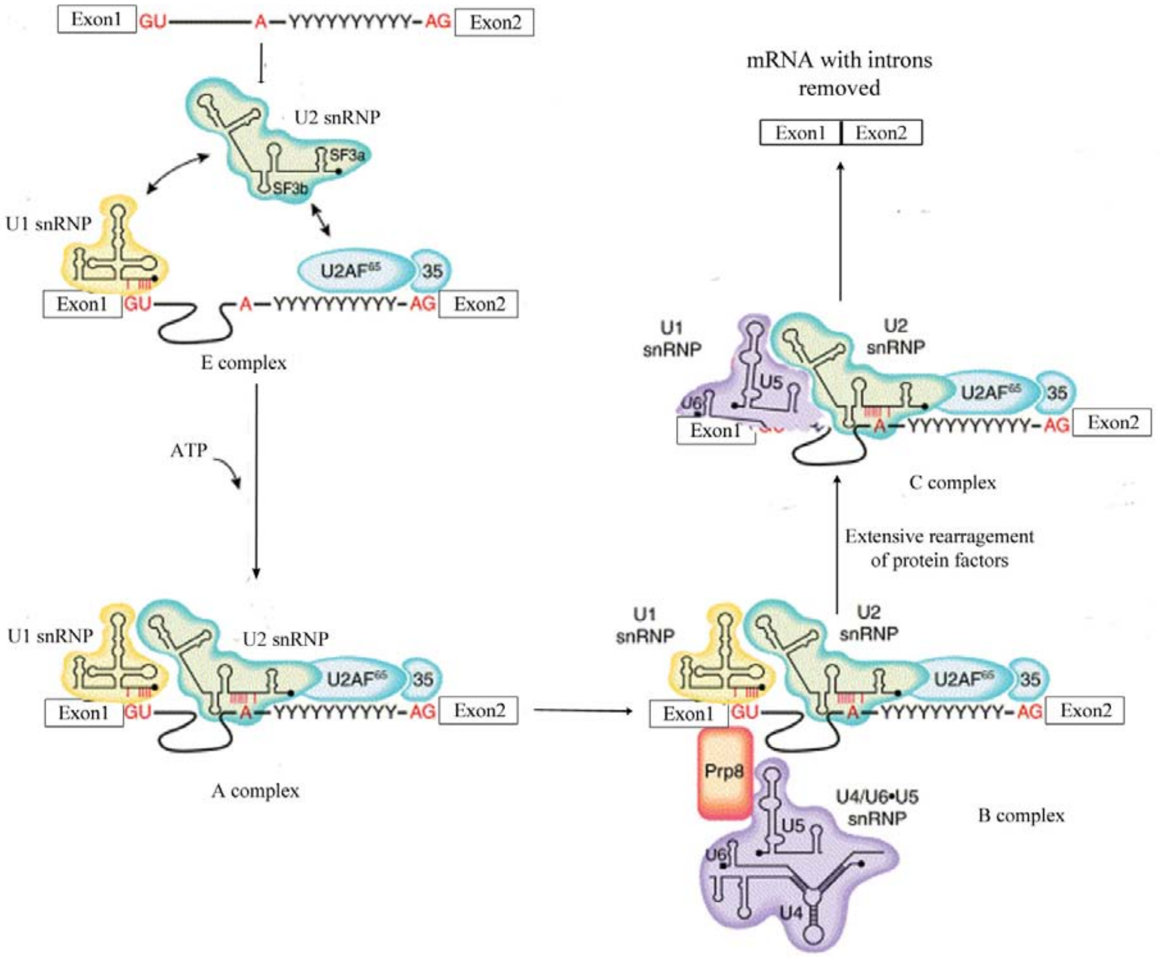


Figure 8. Splicing mechanism where introns are spliced and exons are linked.

The splicing of an intron from a nascent RNA is a two step process requiring two distinct trans-esterification reactions. Initially, cleavage occurs at the donor splice site (the site where introns start) facilitating the first base of the intron to form a lariat structure with the BPS signal present upstream of the acceptor site (the site where introns end). This step is referred to as *branching*. Next, a new phosphodiester bond is formed between the last base of the upstream exon and the first base of the downstream exon. The intron is then released (Jurica and Moore, 2003). These reactions occur within the spliceosome complex, responsible for recognizing splice sites and catalyzing the reactions. The spliceosome is largely made up of five RNA-protein complexes known as small nuclear ribonucleoproteins (snRNPs).

Before cleavage at the donor site, the signals at this site are recognized by U1 snRNP with the formation of commitment complex (E complex). This process does not require any energy component like ATP and it was noted recently that the step is not a strict requirement, as introns were found spliced efficiently *in vitro* even in the absence of U1 snRNP (Crispino *et al.*, 1996). A key role of U1 snRNP complex is to promote the association of U2 snRNP complex with the BPS signal. This interaction is dependent on two other interactions – U2AF^{65} with the polypyrimidine tract of the BPS and U2AF^{35} with the intron terminal AG dinucleotide (reviewed in Reed, 2000). This step is an ATP dependent process where six proteins, including DEAD box protein UAP56 and components of essential splicing factors, SF3a and SF3b, bind either upstream or downstream of the BPS. The association of U1 and U2 snRNPs defines complex A.

Association of the tri-snRNP complex containing U4, U5 and U6 snRNPs with the complex A is required to form complex B. This interaction was recently found to be promoted by the splicing factor SPF30 although this transition remains poorly defined (Rappsilber *et al.*, 2001). This tri-snRNP complex interacts with the donor and acceptor splice sites, recruits other factors including the highly conserved Prp8 protein, and forms the catalytic core of the spliceosome (reviewed in Jurica and Moore, 2003). Although the complete role of this catalytic core is under investigation, it is understood that the tri-snRNP brings about a series of RNA-RNA rearrangements, with the displacement of U1 snRNP from the donor splice site by U6 snRNA, creating the catalytically competent C complex. These rearrangements have been found to be directed by an RNA helicase of the DExD/D box protein family (Schwer, 2001).

The catalytically competent C complex facilitates the second trans-esterification reaction between the upstream and downstream exon with the excision of the spliced intron and mature mRNA.

In yeast, branching occurs with an almost invariant BPS signal UACUA**A**C (with branch A in bold letter), 20-30 nucleotide upstream of the acceptor splice site. The mammalian BPS is less well conserved but generally conforms to the consensus YNYUR**A**Y signal. Recognition of BPS is mediated by base pairing of an invariable sequence in U2 snRNA. It has been suggested that the Branch Point (BP) nucleotide is bulged out from this RNA duplex and this may activate the 2' hydroxyl group for nucleophilic attack. The natural BP nucleotide is adenosine; however, exceptions have been reported. For example, branching of the first intron of the human growth hormone gene and the third intron of the human calcitonin/CGRP gene occur mainly at a cytosine and uridine residue respectively (Adema *et al.*, 1988; Hartmuth and Barta, 1988).

Branching in higher eukaryotes requires other elements found near BP nucleotide. In human, the branch sites map 18-37 nucleotide upstream from the highly conserved AG dinucleotide separated by a polypyrimidine tract of variable length. The length and uridine content of the tract are important factors for branching. At a very early step in spliceosome assembly (complex E formation) the U2AF^{65} and SF1 bind the polypyrimidine tract and the BPS signal. SF1 recognizes primarily the two most conserved nucleotides in the BP sequence YNY**U**R**A**Y. In addition, a direct interaction between SF1 and U2AF^{65} has been demonstrated that may account for the coupled recognition of the BP sequence and polypyrimidine tract.

For the second trans-esterification reaction, the conserved AG dinucleotide, at the acceptor splice site, plays a highly important role and usually the first AG dinucleotide downstream of the BPS is generally used. This is probably selected by a scanning mechanism.

The limiting step in the whole splicing process lies in the recognition of the intron itself. In yeast, where introns are short, the spliceosome is thought to form directly on the intron through the process of *intron recognition* (Talerico and Berget, 1994). However, in human, where short exons are interrupted by long introns, recognition is thought to be based on an alternative model called *exon recognition* (Berget, 1995). Recognition in both models involves splicing associated SR proteins, which play a major role in bringing spliceosome components together (Graveley, 2000).

The splicing process explained so far obeys the normal GT-AG rule and the spliceosome is referred to as the U2-type. However, there is another set of donor and acceptor sites, which displays the AT-AC rule. These splice sites are utilized by a distinct spliceosome called U12 spliceosome that contains U11, U12, U4atac, U6atac and U5 snRNPs (Tarn and Steitz, 1996, 1997). Interestingly, U12-dependent system is lacking in *Saccharomyces cerevisiae* and *Caenorhabditis elegans*.

Distinct differences have been observed between U2- and U12-dependent types of introns. U12-dependent signals exhibit strongly conserved and informative donor and branch signals whereas U2-dependent ones exhibit only moderately informative signals at the donor and acceptor sites and a highly degenerate BPS. Additionally, the polypyrimidine tract found in U2-dependent introns is either not present or weaker in U12-dependent introns (Will *et al.*, 1999).

However, both the systems are not entirely independent of each other and are often found to evolve together. Recent results have found a strikingly high degree of similarity of overlap between the proteins and non-coding RNAs of both systems. These include U5, Prp8, 8 snRNP Sm proteins, SF3b components and SR proteins. Moreover similarity in secondary structures and interactions between the set of non-coding RNAs U11, U12, U4atac and U6atac and the set of U1, U2, U4 and U6 in U2-dependent systems argue that both the systems are homologous to each other (Hastings and Krainer, 2001; Schneider *et al.*, 2002; Will *et al.,* 2001; Will *et al.,* 1999).

1.6.2 Roles of splicing

In transcription: The role of introns in the genome and their probable function has been a fascinating area of study for quite sometime. Along with other functions reported, introns are considered to be a rich source of regulatory elements with the first introns having most elements (for details see, Le Hir *et al.,* 2003; Mattick, 1994; Salamov *et al.,* 1998a). For example, the 280 nucleotide regulatory elements in the first intron of the *c-myc* gene blocks transcription elongation (Pan and Simpson, 1999). In mice, intronless transgenes are transcribed 10-100 times less efficiently than their intron-containing counterparts (Brinster *et al.,* 1988; Le Hir *et al.,* 2003). In yeast, promoter proximal introns enhance transcription initiation with the association of U1 snRNA and initiating factor, TFIIH (Kwek *et al.,* 2002).

The role of CTD of RNA polymerase on the recruitment of processing factors and mRNA maturation has been well established. However, with the latest studies it was shown the communication actually goes both ways, with the assembling spliceosome providing positive feedback to the polymerase. The tat-specific factor (TAT-SF1) recruited on newly transcribed introns interacts with the kinase, pTEFb, capable of phosphorylating the C-terminal domain. This increased CTD phosphorylation is necessary for both promoter clearance and efficient transcription elongation.

Similarly, the interaction of spliceosome component with cap binding complex and poly(A) processing factors enhances the recognition of the 5' most and 3' most introns respectively. *In vitro* studies show an upstream 3'-splice site can significantly enhance use of a downstream polyadenylation site, and a downstream polyadenylation site can, likewise, increase excision of the 3'-most intron (Proudfoot *et al.,* 2002). Protein-protein interaction experiments confirm that both the snRNP protein U1A and SRm160 (SR-related matrix protein of 160 kDa), a splicing co-activator, interacts with the cleavage-polyadenylation specificity factor, CPSF 160. Furthermore, interactions between the C terminus of poly(A) polymerase and the splicing factor U2AF65 and U1A can enhance upstream 3'-splice site recognition (Vagner *et al.*, 2000).

These interactions between splicing and transcription components are not only related physically but temporally too. In α-TM, constitutive splicing factors bind to the splice site signals of exon 3, committing it to the normal splicing pathway. In regulated splicing, an alternative set of factors are thought to bind to the URE and DRE in the flanking introns, forming an inhibitory complex for constitutive splicing. Delaying the transcription of the DRE element through the introduction of some spacer sequences and hindering the regulated splicing complex formation, and removed the inhibitory effect. This indicates that on transcription, splicing factors along with its regulators are available and the decision for constitutive or regulated transcription can occur due to the lag between the transcribing polymerase and splice site and the relative distance between competing elements. Thus, the rate of transcription and the pausing of the polymerase while transcribing might decide the processing pathways (Roberts *et al.,* 1998).

In translation: In Xenopus, splicing was reported to influence translational efficiency as well without significantly altering the steady-state cytoplasmic mRNA levels. When a mature mRNA is injected directly into oocyte nuclei, it is translationally repressed after export to the cytoplasm. This repression can be overcome with a spliceable intron in the 3' UTR. Splicing can apparently enable an mRNA to escape masking of mRNPs and to actively engage ribosomes (Braddock *et al.,* 1994). In another experiment, Matsumoto *et al* found that an intron placed in the 5' UTR was highly stimulatory, whereas the same intron placed in the 3' UTR repressed translation to below the level of the corresponding intronless mRNA (Matsumoto *et al.,* 1998).

In pre-mRNA processing: Apart from influencing transcription and translation processes, adjacent introns in a pre-mRNA affect one another's splicing efficiency too. Results from related experiments form the basis for an *exon recognition* model, which depicts that the acceptor splice site of an upstream intron helps to increase the efficiency of recognition of the donor splice site of a downstream intron through components of the splicing machinery and vice versa. The interactions, which link the upstream acceptor splice site and the downstream donor splice sites, involve U1 snRNP and U2AF65 and these are thought to be mediated by SR proteins. SR proteins generally possess one or two RNA-binding domains (recognition motifs, RRM) and an arginine-and-serine rich region (the RS domain). RRMs often target SR proteins to exonic splice enhancers. The RS domain then appears to provide a molecular 'glue' allowing RS-RS interactions between interacting factors and thus facilitating the recognition of intron-exon boundary by the splicing apparatus (Graveley, 2000).

These SR proteins are also found associated with the CTD and are either referred to as CTD-associated SR-like protein or SR-like CTD-associated factor. The heptad-repeat sequence of CTD is micro-heterogeneous and that might result in different levels of phosphorylation and affect significant levels of SR-protein interaction with CTD (Graveley, 2000).

Transcription and Splicing Rate: Using a well-documented alternatively spliced intron from the highly intronic gene for fibronectin, it was shown that different types of promoters initiate various splicing pattern of transcripts (Cramer *et al.*, 1997). Over expressing various SR proteins are also found to affect the splicing patterns, sometimes antagonizing the promoter effects (Cramer *et al.*, 1999). These results are consistent with a model in which SR-protein interactions with the CTD are set up early in the transcriptional initiation process. Also, the correlation between transcriptional rate and splicing was also shown previously (Roberts *et al.,* 1998). When transcription slows down its rate on specific parts of the gene, it might influence the splicing patterns of nearby exon sequences. Thus these results emphasize, mRNA processing and transcription are interlinked.

With this understanding, it is clear that analyzing the splicing mechanism is imperative while discussing RNA polymerase II transcription and translation.

1.6.3 Computational detection of splicing signals

Consensus signals for splice sites were quickly recognized and were used to determine the gene structure. However, it was recognized later that many functional splice sites shared only a few bases of similarity and more sophisticated models were required.

Simple independent weight matrices or frequency tables that yield a probabilistic log-odds score for each base at each position in a sequence were initially developed and they are still used extensively (Staden, 1984). Weight matrices were derived from a training set of true sites to generate the frequency table and then score potential sites by summing the scores of individual bases in a pre-defined window. This was improved with the incorporation of first-order dependencies into the weight matrix framework (Zhang and Marr, 1993).

The next set of improvements came with the components of a successful gene prediction system *GENSCAN* (Burge and Karlin, 1997). It uses a maximal dependence decomposition approach, where the donor sites are broken into a set of classes based on dependencies between bases in the splice site signal and then uses a simple weight matrix to model each class individually (Burge, 1998). For acceptor sites, it uses a windowed weight array method, which models BPS region using a modification of first-order dependencies approaches that groups sets of neighboring bases together in order to avoid problems caused by limited data.

Later, multiple signals were used to identify splice site regions. *GeneSplicer* (Pertea *et al.*, 2001) combines a traditional log-odds score based on a slight variant of maximal dependence decomposition, a measure of local coding potential and a local optimality requirement. But this approach did not yield improved results.

Another approach was used to identify precise splice sites from among a number of nearby or proximal false positives. This approach used a decision tree to discriminate true and false sites and may prove useful for annotation purposes (Thanaraj, 2000). However, these models produce too many false positives per kb. Typically, if thresholds are set to detect 99% TP, then 12 FP per kb and for thresholds to include 95% TP, 6 FP per kb were reported (Levine, 2001a).

EST sequences have also been used to confirm the site signals on a large scale basis and in analysis of canonical and non-canonical introns (Burset *et al.*, 2000), though their use means the algorithm is no longer truly *ab initio*.

As an attempt to improve previous methods, another program, *Stratasplice* (Levine, 2001a) was developed in which true and false positives were differentiated using the base composition near the splice signals. The local GC content with a first-order dependence weight matrix combination model is used by the predictor to predict the human splice sites. This resulted in better prediction of splice sites of genes in GC-rich sequences.

However, all the programs developed so far, are limited in that they produce excessive false positives when applied on a genome scale. Hence, I attempt to develop a few splice site models that will do fairly on the genomic sequence and will complement the transcription termination predictor in identifying real transcription terminators.

1.7 Transcription and translation

The protein coding mRNA, transcribed by RNA polymerase, is later used for coding for protein synthesis by a process called *translation*. Transcription and translation are coupled in prokaryotes where there is no defined nucleus or nuclear membranes to separate genetic material from the cytoplasm. However, in eukaryotes it is traditionally believed that these processes occur separately with the transcribed RNA product processed and exported to the cytoplasm where the translation occurs. Also, it is often suggested that the membrane evolved to segregate splicing and translation so that they do not interfere with each other. This understanding was recently challenged with the recent finding of nuclear translation in mammalian cells (Hentze, 2001; Iborra *et al.,* 2001). Three types of evidences supported the possibility of coupled transcription and translation in the eukaryotic cell just like in bacteria. (i) Nuclei contain all the components required for protein synthesis, (ii) Isolated nuclei can incorporate radiolabelled amino acids to make new protein molecule and (iii) Nonsense-mediated Decay (NMD), which is responsible for degradation of transcripts with termination codon near to the 5’-end support the transcription and translation coupling in eukaryotic nucleus (for details see, Hillman *et al.*, 2004; Iborra *et al.,* 2004). NMD, which mostly occurs in the cytoplasm, is also found in the nucleus and this poses a challenge to the current consensus. However, this phenomenon can be explained if some translation occurred within nuclei by the protein machinery present within the nuclei. So, the present model is that ribosomes are assembled within nucleoli and are exported to both nucleoplasm and cytoplasm, where they associate with transcripts and become active. Some nuclear ribosomes are incorporated into the transcription factories and proof-read the newly made transcripts as they emerge from polymerases. Any pre-mature codon in the transcript would trigger the NMD pathway and degrade the transcripts with nearby proteasomes. If no premature stop codons are found, the transcript would be exported to the cytoplasm where it could support multiple translation initiations. Thus, there is evidence that transcription and translation mechanisms are interlinked, so understanding translation signals and modeling them may complement the transcription start site and termination models in predicting the gene structure.

One of the mechanisms by which pre-termination codons are incorporated in the transcript is a frame shift splicing mechanism, and this triggers the NMD pathway (Lewis *et al.,* 2003). This leads to the understanding that identifying translation termination codons will help to legitimate the correct splice sites and screen out the numerous splice site-like signals from the genomic DNA. Thus, translation models may supplement other models in predicting genes in the genomic DNA.

1.7.1 Translation mechanism

Explaining the translation mechanism in detail is beyond the scope of this thesis. So I will give a brief overview of the mechanism in prokaryotes and eukaryotes instead.

1.7.1.1 Translation initiation

The translation initiation mechanism in prokaryotes differs from that in eukaryotes and the process in both is more than a mere assembly of protein components. The initiation phase sets the reading frame which is normally maintained throughout all subsequent steps in the translation process. Moreover, protein synthesis is regulated at the level of initiation, which adds to its importance.

Initiation in prokaryotic polycistronic mRNA is usually selected *via* base pairing with ribosomal RNA. This initiation is regulated by *cis*- and *trans*-acting signals. In eukaryotes, translation initiation sites are reached *via* a scanning mechanism from the AUG codon near to the 5’ end of mRNA. However there are also other mechanisms through which initiation can occur. These are context dependent leaky scanning, reinitiation and internal initiation where translation initiation is directed from an AUG that is not the nearest to the 5’ end (for details refer, Gray and Wickens, 1998; Kozak, 1999, 2001; Kozak, 2002; Pain, 1996; Sonenberg and Dever, 2003).

At the start codon, the 30S ribosomal subunit forms an initiation complex with a special form of tRNA (fMet-tRNA) and a GTP-binding protein IF2. IF1 and IF3 stabilize the binding of fMet-tRNA·IF2·30S complex and thus initiate polypeptide chain formation with addition of methionine. AUG is the common initiator codon because it forms a stable interaction with CAU anticodon in fMet-tRNA. GUG and UUG are also used as start codons in >10% of bacterial genes. AUU codon is used in a single *Escherichia coli* gene. The initiation phase is completed with the 50S ribosomal subunit forming a 70S unit with fMet-tRNA occupying the P-site of the ribosome.

Start codons in prokaryotic mRNA are distinguished by an upstream purine-rich sequence that pairs with a complementary sequence in the 16S rRNA component of the small ribosomal subunit. This sequence, called the *Shine-Dalgarno* (SD) sequence, consists of three to nine contiguous bases in the mRNA that form standard base pairs (not including G·U) with bases from 1534 to 1542 (ACCUCCUUA) at the 3'-end of 16S rRNA. This SD interaction augments initiation by anchoring the 30S subunit in the vicinity of the start codon. Apart from the SD signal present nearby the start codon, several trans-acting signals and factors have been reported. However, the SD sequence is not essential in all initiations as some AUG codons are found to initiate without SD augmentation. Similar cases were reported for chloroplast mRNAs as well. In these cases, the SD sequence is generally considered to be substituted by a low GC content (hence minimal secondary structure) in the 5' UTR region (for review see, Kozak, 1999).

Efficient formation of initiation complexes requires the sequence immediately preceding the SD element to be devoid of any secondary structure. Some additional sequence elements present downstream of the AUG codon might substitute for the main SD element. These elements have patchy complementarity to 16S rRNA and include weak G·U pairings and so their significance remains inconclusive. Many prokaryotic mRNAs are polycistronic and ribosomes translating the first open reading frame will often, upon termination, slide a few bases upstream or downstream to reinitiate at the next start codon.

The eukaryotic mechanism differs with the 40S ribosomal subunit entering near the 5' end and sliding its way to identify the first AUG codon, which is recognized by base pairing with the anti-codon in Met-tRNA$_i$. AUG is the most common initiating codon; however, ACG and CUG codons are also used. Methionine is the first amino acid even when the first codon is other than AUG. Eukaryotic initiation depends on the m7G cap added to the 5' end of mRNA molecule. In vertebrate mRNAs, the initiation sites has a consensus sequence of GCC**R**CC**AUGG** with R (purine, mainly A) and G at -3 and +4 positions showing more active role (Iida and Kanagu, 2000; Kozak, 1987). Poly(A) tail and 3' UTR might also influence translation initiation (Figure 9).

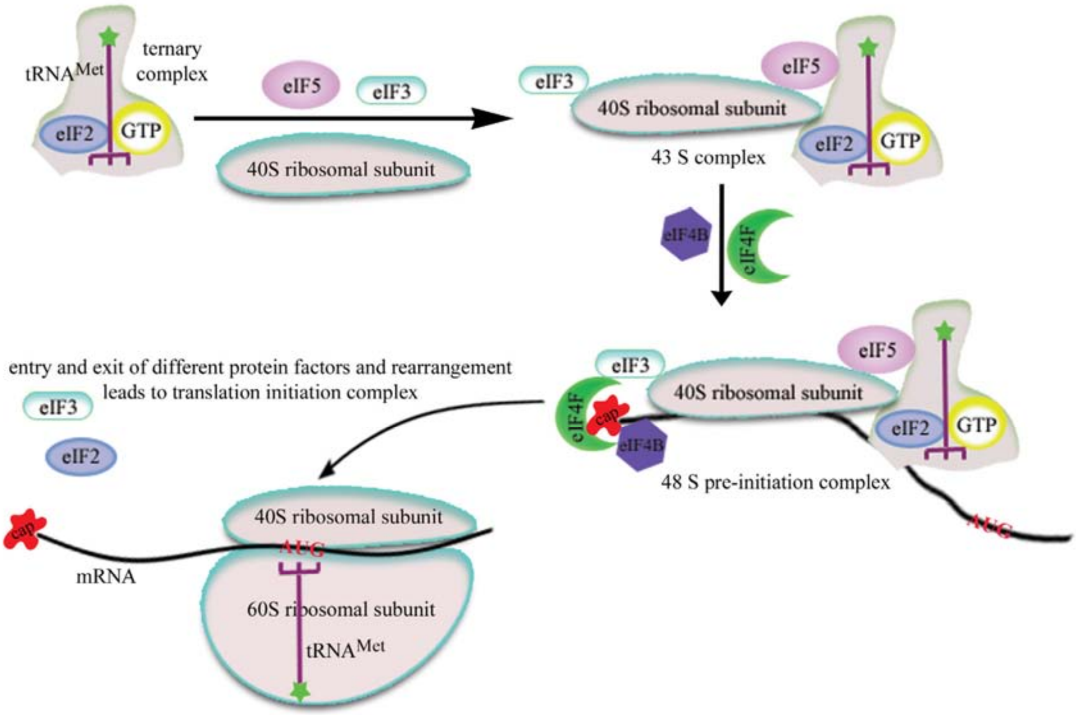


Figure 9. Translation initiation in eukaryotes

Leaky scanning allows 40S ribosomal subunits to by-pass the first AUG codon and initiate instead at the second or rarely at the third AUG codon. This is mainly due to sub-optimal context near to the first AUG codon. There is some evidence that initiation can occur with non-AUG codons as well. Re-initiation in eukaryotes occurs if the initiation complex gets terminated at some distance near to the 5' end. Scanning then continues until the next authentic AUG is reached. IRES (Internal Ribosome Entry Site) is another mechanism, wherein translation of mRNA occurs from an internal initiation site (Houdebine and Attal, 1999).

1.7.1.2 Translation termination

Translation termination is due to stop codons in the mRNA sequence. When a stop codon has been translocated into the ribosomal A-site by the action of elongation factor EF-G or eEF2, a cleavage of the ester bond between the peptide and tRNA moieties of the peptidyl-tRNA complex occurs at the peptidyl transferase centre of the ribosome. In prokaryotes, termination involves two different release factors recognizing UAA/UAG and UAA/UGA respectively, whereas in eukaryotes all the three stop codons are recognized by a single release factor. Eukaryotic release factor binding to the ribosomal A site is GTP dependent and RF3·GTP binds at this site when it is occupied by a termination codon. Then, hydrolysis of the peptidyl-tRNA ester bond, hydrolysis of GTP, release of nascent polypeptide and deacylated tRNA and ribosome dissociation from mRNA ensue (Kisselev and Frolova, 1995) (Figure 10).

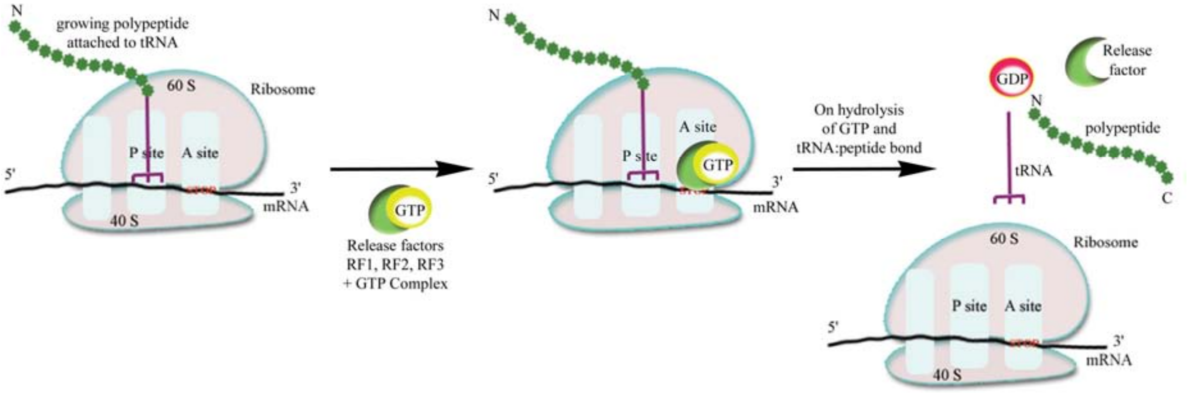


Figure 10. Translation termination mechanism mediated by release factors

Translation termination efficiency was found to be improved by the local context in yeast genes. The consensus sequence, CA(A/G)N(U/C/G)A, located downstream of the stop codon base pairs with the regions close to helix 18 and 44 of the 18S rRNA for augmenting translation termination efficiency (Namy *et al.*, 2001). In higher eukaryotes, the stop codons are biased towards purines (Cavener and Ray, 1991). Also, the CpG dinucleotide patterns present immediately downstream of the stop codons are significantly suppressed (Cavener and Ray, 1991).

The downstream context also plays a detrimental role for the UGA triplet in deciding whether it is used as a termination codon or selenocysteine codon.

Analysis of full length RIKEN mouse cDNA and eukaryotic UniGene clusters (Ozawa *et al.*, 2002) showed the following results –

(i) The occurrence of guanine at position +1 (immediately after the stop codon) was high in mammals. Adenine was high at this position in plants and Zebrafish.

(ii) The occurrence of cytosine at position +1 was low in plants.

(iii) The occurrence of cytosine at position +4 was high in mammals.

(iv) The occurrence of cytosine at position +2 was high in plants. In human positions +2, +3, +4, +7 and +13 after stop codons have some information content.

Apart from DNA signals, protein factors also influence translational efficiency. PABP1 interacts with initiation factors eIF4G and eIF4B and promotes the synergistic effect of having both a cap and poly(A) tail on translation efficiency. The translation termination factor eRF3 also interacts with PABP1 and so could relay information from the termination complex to both ends of mRNA and thus regulate subsequent translation initiation (Cosson *et al.,* 2002).

1.7.2 Computational detection of translation signals

Identifying translation start sites depends on the consensus signals identified near to the initiator codon. Several attempts have been made to correctly identify the translation start site and to screen true sites from the false sites in the genomic DNA.

In 1987, Kozak developed the first weight matrix from an extended collection of vertebrate mRNA data (Kozak, 1987). The consensus motif derived from the matrix is GGG**A**CC**ATGG**, where a single G nucleotide following the ATG codon and three A nucleotides upstream are two highly conserved positions.

Later prediction methods took the nucleotide context in the vicinity of the start site as well. These include the positional conditional probability matrix (Salzberg, 1997) and generalized second-order profile models (Agarwal and Bafna, 1998). In the Agarwal and Bafna model, an algorithmic idea of the ribosome scanning model was implemented. The search starts from the 5’ end of the mRNA and an AUG is defined as a putative start codon if followed by an ORF longer than 200 nucleotides. Likewise, in the Pederson and Nielson *NetStart* model, an Artificial Neural Network (ANN) was constructed with 100 bases upstream and downstream of AUG codon that recognizes the surrounding context (Pedersen and Nielsen, 1997b). These approaches are significantly better than weight matrix models but still generate high false positive rates.

So, to improve the prediction accuracy, Salamov *et al.*, developed a program called *ATGpr*, where the following six characteristics are applied to analyze the sequence around putative start sites:

(a) Positional weight matrix around an ATG.
(b) Hexanucleotide difference between upstream and downstream of ATG sequences.
(c) Preference for longer reading frames downstream of ATG.
(d) Signal peptide characteristic.
(e) Presence of another upstream in-frame ATG.
(f) Upstream cytosine nucleotide characteristic.

Linear discriminate analysis was used to finalize the score from these properties. The important components in the ATGpr model are the positional triplet weight matrix around AUG and the hexanucleotide difference between the upstream and downstream of the AUG in a 50 nucleotide long window (Salamov *et al.*, 1998a). Along with these properties, another program developed by Zhang *et al.* used 50 base pair downstream windows to screen for in-frame stop codons and local context to determine translation start site (Zhang *et al.*, 2000).

Recently, a method based on Support Vector Machines (SVM) (Cristianini and Shawe-Taylor, 2000) has been introduced by Zien *et al* (Zien *et al.*, 2000). To add to this, Liu *et al* used SVM as classifiers with possible amino acid patterns around start sites to differentiate true and false sites (Liu *et al.*, 2003). Similar to this, an ANN with the ability to determine coding/non-coding potential around the start codon and conversed motif was also developed (Hatzigeorgiou, 2002).

Contrary to these various translation start models, not much computational analysis has been carried out on translation stop prediction as identifying them becomes relatively easy if the correct translation start site and ORF can be determined.

1.8 Objectives of this project

With this understanding, it is clear that for a gene prediction program that works purely based on gene regulatory signals, it is necessary to have efficient methods to capture the complexity of regulatory signals linked to each process from the genomic sequences.

As a transcription start site predictor is available (Down and Hubbard, 2002), modeling transcription termination is the next important step as the task has proved challenging for nearly 25 years now. Extensive research on transcription termination through these years has still not cleared the enigma and a clear mechanism of the process is yet to be realized. So, the major aim of this project is to build a transcription termination model using the genomic sequences available with the different techniques explained in chapter 2. A successful predictor will be useful to identify the point where RNA polymerase II stops transcription and exits from the DNA sequence, and thus helping to sketch the gene structure. This is explained in chapter 3 along with some interesting results found by the model.

As explained previously, transcription is tightly linked with the splicing and translation process and thus identifying their regulatory signals may help to supplement the development of a transcription termination model. So I have set the objective of modeling splice site and translation start and stop signals as well. Chapters 4 and 5 detail the models trained to meet these objectives based on the learning techniques explained in chapter 2.

Finally, in chapter 6, I meet the objective of creating an *ab initio* gene prediction system based on DNA regulatory signals by linking the predictions of the models using GAZE (Howe *et al.,* 2002).

Apart from this goal; I worked on two other project areas as well. These are explained in the Appendices. Appendix A gives an overall view of the project with the aim of identifying domain insertions in known protein structures. Appendix B details the analysis of protein evolution based on sequence and structure conservation.

MATERIALS AND METHODS

2.1 Introduction

In this chapter I explain the different strategies used in learning regulatory signals from DNA. Learning signals or motifs from nucleotide sequences has been quite a difficult task and various methods have been adopted so far. Prominent among them is the Hidden Markov Model (HMM) widely used in speech recognition, sequence alignment and gene prediction. A Hidden Markov Model is a directed graph of states connected by transition paths and throws emission and transition probabilities. Walking through these states and probabilities, HMM models the features in the DNA sequence. However such methods were found to be helpful only in cases where features in similar sequences were aligned with each other.

This requirement makes it difficult to learn transcription termination signals as some motifs downstream of the cleavage site, responsible for polymerase pause and release, are found over a wide range of distances from cleavage site (Dye and Proudfoot, 2001). Hence, here I used another method based on the sparse Bayesian principle that can accommodate the distance variation. The method is a probabilistic generalized linear model which scans for motifs that describe the given set of sequences and learns them by constructing a model. This model can be later used to classify sequences with and without transcription termination signals. Derivation of this model is based on the conditional probability of Bayes theorem given below –

$$P(model \mid data) = \frac{P(data \mid model)P(model)}{P(data)} \qquad (1)$$

where, *data* represents a DNA or cDNA sequence. *P(model|data)* is the posterior probability that gives the probability of a sequence derived from the *model*. The posterior probability depends on the probability of the *data* given the *model* and probabilities of the *model* and *data*.

As estimating the real probabilities of the *model* and *data* are difficult, various approaches have been adopted. One such approach depends on how best the Bayesian model can fit the sequence compared with the chosen null model. Learning the Bayesian models that best fit the given set of sequences with the regulatory motifs is possible with different types of trainers, like Relevance and Support Vector Machines. Relevance Vector Machine (RVM, Tipping, 2001a, b) is a Bayesian treatment of a Generalized Linear Model (GLM) of identical functional form to the Support Vector Machine (SVM, Mackay, 2003; Scholkopf *et al.,* 1999; Vapnik, 1995). However, RVMs has the advantage of emitting a probabilistic output unlike SVMs and using fewer kernel functions to classify the data. In this project, I used an implementation of RVM called *Eponine* (Down and Hubbard, 2004) to learn gene regulatory signals.

Initially I used Eponine to identify transcription termination motifs and then extended it to learn translation start, translation stop and splice sites. During this process I tweaked the default parameters of the trainer to suit the regulatory signals to be learnt. For example, the Gaussian distribution employed to accommodate the positional distribution of motifs in the sequences in learning the termination model was changed to a Delta distribution for splice site models as the splice signals show less positional variation in their occurrence.

The features predicted by these sequence models are then linked together using a dynamic programming based gene component assembler called GAZE. GAZE combines the features and predicts a gene structure in the sequence consistent with a supplied gene structure model (Howe *et al.,* 2002).

After investigating the use of sequence models for detecting transcription termination sites, I also tried secondary structure prediction algorithms with the objective of finding any stem-loop structures that might influence transcription termination. I used two basic algorithms developed by *Nussinov* (Nussinov, 1978) and *Zuker* (Zuker and Stiegler, 1981) for this purpose.

The *Nussinov algorithm* is very simplistic and is based on *base-pair maximisation* metrics. Whereas the *Zuker algorithm,* along with base pair metrics, uses a *free energy minimisation* technique based on experimentally determined energy parameters. The minimal free energy criterion helps the selection of the best possible structures out of the ensemble of folds predicted.

None of these analyses would have been possible without the excellent databases in the public domain. Here, I have used human chromosome 22 and 20 data widely as these were the most accurately annotated chromosomes available at that time. The recently published annotations on chromosome 22 with experimental support formed an excellent source to derive training and test datasets (Collins *et al.*, 2003). Likewise, manually curated high quality annotation for chromosome 20 was extracted from the *VEGA* project (Ashurst, 2002). This project is an attempt to co-ordinate curated annotation process for the finished vertebrate genomes. Likewise, *ENSEMBL* is another excellent database that contains genome sequence data for organisms, automatically annotates it and serves the annotated sequence through the internet (Birney *et al.*, 2004). Various tools in ENSEMBL along with the supporting evidence for annotation derived from different sources helped me in this project. At other times I have used the *RefSeq* database (Pruitt and Maglott, 2001) and the *RIKEN* mouse cDNA collection (Kiyosawa *et al.*, 2003) as well .

In the remainder of this chapter, I explain the details of these algorithms and databases used in this project and conclude the chapter by briefly describing the two open source projects, *Bioperl* and *Biojava* that I used extensively from formatting sequences to building models.

2.2 Hidden markov models

Several methods have been attempted to model sequence signals around regulatory regions. They range from simple sequence composition bias to complex probabilistic machine learning methods like, Neural Networks (NN) and Hidden Markov Models (HMM).

HMM (Durbin *et al.*, 1998) is one of the common modelling systems employed to learn biological signals from DNA or protein sequences and forms the basis for many gene prediction tools. The use of HMMs involves two components – model architecture and its parameterization.

Figure 11 shows a schematic representation of a model architecture. This comprises a set of states, which might be a match state (circles labelled M), insert states (diamonds labelled

I) or delete states (squares labelled D). The states are connected by arrows that represent possible transitions between the states. A DNA sequence can be generated by moving through the model following the arrows. For instance, starting with the state M_0, which generates a nucleotide (AGCT), the next move might be to any of M_1, I_0 or D_2. M_1 or I_0 would generate a second nucleotide but D_2 would not. From these states, the model continues to the next state connected by arrows thus generating a state path and emitting a DNA sequence. Self transition (looping) is allowed for insert states and they are shown as arrows linking to themselves.

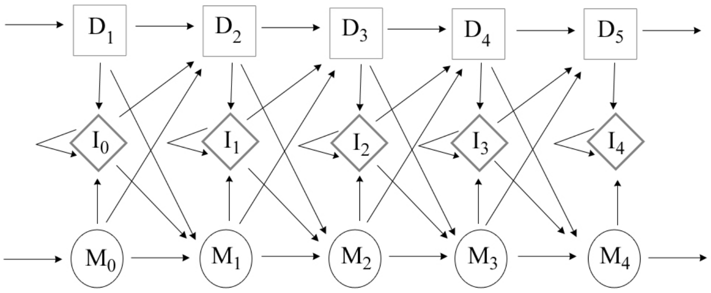


Figure 11. Schematic diagram showing a section of HMM architecture

After designing the architecture of HMM, transition and emission probabilities have to be assigned between and within states respectively. The probability parameters can be easily calculated by counting the number of times each particular transition and emission is used in the set of training sequences when all the state paths are known. However, in cases where the paths are unknown, an iterative method like, Baum-Welch algorithm, is used.

The name ‘Hidden Markov Model’ is used because the sequences are generated by a Markov process, which is defined as a process in which the probability of a particular state depends on the state immediately preceding it in a sequence. Since the state path of the model that generates the sequence is not observed the term ‘hidden’ is used.

I attempted to use HMMs to learn transcription termination signals responsible for RNA polymerase pause and release from the sequences of the end of the gene. These attempts suggest that, HMMs are not a good choice of machine learning technique for use on this problem because of two reasons. Firstly, the sequence motifs at the 3’-end of the gene responsible for termination appear only to be loosely defined, without a strong consensus and thus are difficult to model with a simple HMM architecture. Secondly, the locations of these termination motifs are present at greatly varying positions from the cleavage sites and HMMs have difficulty in modelling such criteria. Although other methods have been developed to model motifs separately with some flexibility on positions as in Meta-MEME (Grundy *et al.*, 1997), the complex architecture needed for such models needs to be built by hand or heuristic methods. This limits the range of architectures that can be explored. So here I have used the Eponine modelling system to learn transcription termination signals positioned at variable points in the DNA sequences. This system allows model architectures to be learnt from the dataset unlike most HMMs.

2.3 Eponine

Eponine is a supervised machine learning approach that can be applied to the training of a wide range of model types and embodies the principle of selecting the simplest possible model to explain the observed data. In this section I briefly explain the Eponine and its implementation. For a detailed description of the tool refer (Down, 2003).

The Eponine package applies Bayesian theory and is able to learn complex models comprised of one or more weighted constraints. Most models consist only of a simple type of constraint called DNA matrices. These matrices are short, ungapped sequence motifs, which contain a series of column distributions over the DNA alphabet. Parameterizations of the model are learnt using an RVM based trainer which takes a positive dataset with the interesting feature and a negative set without the feature. The trainer starts with an initial set of working matrices and iteratively selects only those matrices that can classify the positive dataset from the negative dataset. This tool comes in many flavours and the one I used here is called *Eponine Anchored Sequence* (EAS) method. In this method, the ‘weighted’ matrix (or constraint) is anchored from a particular ‘anchor point’ and is compounded by a probability distribution that describes the distance relative to the anchor. Constraints with a positional distribution are called *Positioned Constraints* (PC) (Figure 12a). Thus PCs consist of –

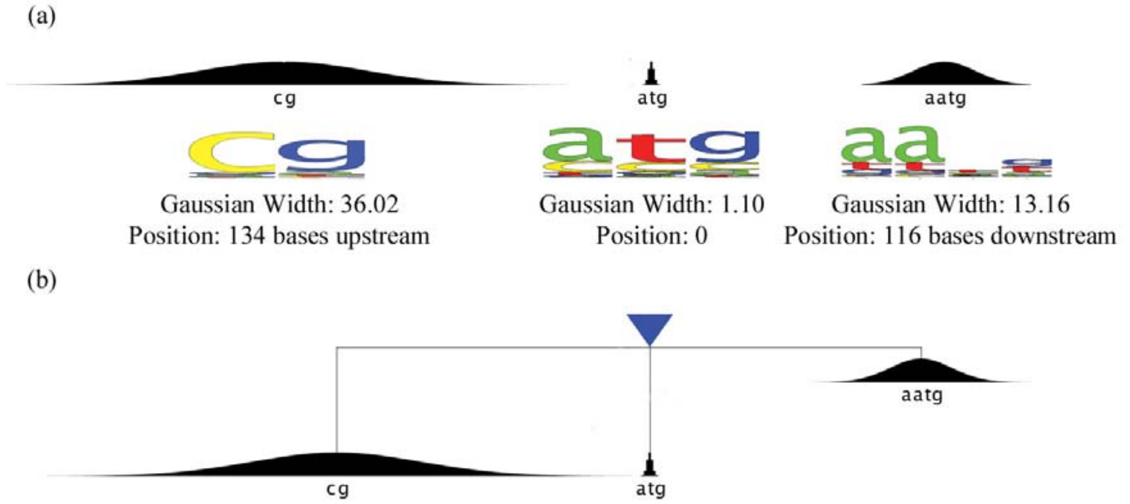


Figure 12. An example of Eponine model. (a) Position constraints along with Gaussian width and position. The nucleotide distribution in the weight matrices are represented as sequence logos. (b) Eponine model constructed from these constraints

- A preferred sequence motif, defined as a position-weight matrix.
- A probability distribution describing the localisation of the motif relative to the anchor point, as described from the integer offsets observed. A Gaussian distribution is used for this purpose as it is simple and less prone to 'overfitting' issues.

New PCs are constructed through training by the following algorithm –

- Pick a sequence given for training.
- Pick a point relative to the anchor point of the sequence.
- Take a sequence motif of 3 to 6 bases at the point and construct a weight matrix.
- Add a Gaussian distribution to the weight matrix of random width centred at the position of the sequence motif found.

After creating the novel PCs from the given data, a range of sampling strategies given below are used in further training to select the PCs that model the training data.

- Select an existing PC and adjust the emission spectrum of one column in the weight matrix by sampling from a Dirichlet distribution (Mackay, 2003) centred on current values.

- Add an extra column to the existing weight matrix till the threshold is reached.
- Remove a column from the start or end of the weight matrix till the threshold is reached.
- Adjust the width parameter of the Gaussian distribution.
- Adjust the centre position for a Gaussian distribution

The score for a PC for a given sequence, *x* is –

$$\phi(x) = \frac{1}{|W|} \log \sum_{i=-\infty}^{\infty} P(i) W(x,i) \qquad (2)$$

where, *P* is a positional probability and *W(x ,i)* is a DNA weight matrix probability for offset *i* relative to anchor point of *x*.

These PCs are then linked together to form an EAS model (Figure 12b) in the form of a *Generalised Linear Model* (GLM, McCullagh and Nelder, 1983), commonly used for classification and regression problems. A Generalised linear function ($\eta(x)$) for variable *x* (such as DNA sequence) is represented as -

$$\eta(x) = \sum_{m=1}^{M} \beta_m \phi_m(x) + k \qquad (3)$$

where, ϕ is a set of *M* basis functions defining the variable *x* (for example, a set of motifs, PCs) and β is a vector of weights (for example, relative importance given to motifs).

Finding an appropriate set of basis function to define the features of the dataset and finding a vector of weights for the given set of basis functions are the two issues to construct an EAS model.

The first problem can be tackled using sparse learning methods like Support Vector Machine (Scholkopf *et al.,* 1999; Vapnik, 1995). Sparsity is a desirable feature as they produce simple models and tend to make useful generalisation of the data. While SVMs have helped to solve biological problems, they are mainly used for numerical data. Nevertheless, deriving such functions is complex and problematic and poses a serious problem in extending to biological data. Moreover SVMs allow training of GLMs only with limited functions that explain the dataset. So to tackle this, another sparse learning method called RVM was introduced (Tipping, 2001a, b). RVM is a Bayesian approach that can train a GLM with any collection of basis functions and thus opens new possibility of solving biological problems.

In a binary classification problem, where each datum x_n has a label t_n (either 1 or 0, meaning positive or negative sequence respectively), the probability that a dataset is correctly labelled given a classifier EAS model $\pi(x)$ can be given as –

$$P(t \mid x, \beta) = \prod_{n=1}^{N} \pi(x_n)^{t_n} (1 - \pi(x_n))^{1-t_n} \qquad (4)$$

where, β is a set of weights.

Now the second problem can be tackled using the Bayes theorem explained before and $P(t \mid x, \beta)$ by inferring likely values of weights given some labelled data.

$$P(\beta \mid x, t) = \frac{P(\beta) P(t \mid x, \beta)}{P(x)} \qquad (5)$$

The probability distribution *P(β)* is our prior belief in the values of weights, *β.* The basic prior is an independent Gaussian distribution, *N*, over the weight of each basis function and can be derived by inferring the values of inverse of Gaussian, α. As the α values are inferred it is necessary to provide an additional hyperprior value and in this case a non-informative a very broad gamma distribution is used.

$$P(\beta) = \sum_{m=1}^{M} N(\beta_m \mid 0, \alpha_m^{-1}) \qquad (6)$$

When a basis function providing additional information to the model gets a non-zero value, the amount of information learnt about the labelled dataset increases and thus the probability of the model given the data. If the basis function provides no information either because of redundancy or irrelevancy, no weight is added that will lead to a significant increase in the likelihood of the function. At this juncture, by setting the α_m parameter to a large value will set the $P(\beta_m)$ to zero and thereby the posterior probability of the model is maximized. Thus a higher α makes the basis function irrelevant and removed from the model and thus simple models are derived resulting in generalisation.

Incorporating as little prior knowledge about the dataset would be an ideal way of training a model. However this will end up exploring a large amount of features of the data for basis functions leading to a computationally expensive process. So a subset of basis function called, working set, is initialized from large sets of candidate basis functions. As described above, when the trainer is run, it calculates the α values for these basis functions and those that get a higher value are removed from the set. Once the size of the subset drops below a certain limit, new functions are added from the pool and the α and β values are initialized and set for training. This is continued until the basis functions from the pool get exhausted. The trainer stops training when there is no significant difference between priors and weights between cycles and converges to an optimal solution.

2.4 Modifying Eponine parameters

2.4.1 Distribution

In splice site and transcription termination models the positional distribution model used to capture the offsets of the motifs; relative to the anchor point in a PC was extended.

In the transcription termination models, the position of the downstream sequence motif from the anchor point (in this case, cleavage site) was found to be variable both form recent experimental results and various training runs. So to accommodate the large variation in the offset values the allowed Gaussian distribution width was modified significantly from the default parameters used for other models. The beauty of the trainer is that despite being allowed to use a broader distribution, it could still learnt both a broad distribution for downstream motifs and a tight Gaussian for the poly(A) and auxiliary signals.

Similarly, in the case of splice site models, various trails lead to the conclusion that the model is simpler if a Delta rather than a Gaussian distribution is used to capture the offset values of the PC relative to the anchor point, so this function was implemented and the modelling system configured to automatically select between Gaussian and Delta during training. Also, the model training was supplemented by providing simple weight matrices as a sample set of basis functions while training.

2.4.2 Position weight matrix

In a Weight Matrix (WM), each column represents the probability distribution of the nucleotides at a particular position in the sequence. A weight matrix can be treated as a probabilistic model *M* of fixed length sequence with no gaps. Then the probability of a sequence *x* fitting this model can be given as –

$$P(x \mid M) = \prod_{i=1}^{L} e_i(x_i) \tag{7}$$

where, *L* is the length of the matrix and $e_i(x_i)$ represents probability of observing base x_i at position *i*. This model is considered as a zero order model as each position in the motif is assumed to be independent of all others. The probability is estimated as log odds score by comparing with the probability of observing *x* under a random model, *q*. The log-odds score is calculated using this formula –

$$S = \sum_{i=1}^{L} \log \frac{e_i(x_i)}{q_i(x_i)} \tag{8}$$

Position weight matrix can be viewed as trivial HMM where a series of states are separated by transitions with probability of 1. This means, after observing an emission probability over the ACGT alphabets of each state (column in the matrix) the machine moves to the next state in a fixed manner. This simple WM can be wrapped as HMM by adding a few additional states to emit a variable number of flanking sequences of each side of the motif. This simple case can then be blown up to complex HMM by adding states to deal with insertions and deletions. With this model architecture, the maximum likelihood estimate of the parameters that explain the set of datasets can be found using trainers based on the Baum-Welch algorithm (Durbin *et al.,* 1998). However in my case, instead of using these WM as HMM independently they form a set of working basis functions for the RVM to classify positive sequences from negative. I used this strategy for splice site training, as the default parameters in Eponine were not suited to derive a convergent model. Adding external WM as a set of basis functions helped me to derive sparse splice site models. This strategy is commonly used in gene prediction programs as well. Position weight matrices were also used in HMMs before with allowance for small insertions and deletions to the expected consensus motif. Pfam protein models are built with this strategy (Bateman *et al.,* 2004).

2.5 Nussinov algorithm

Single stranded RNA molecules tend to form higher order structures which are recognised by the proteins regulating various functions of the cell. The structures are mainly based on base pairing and hairpins are the most common structures found in RNA. The base pairings are conserved due to functional constraints on these RNA molecules. Secondary structures in RNA have various features and are represented in Figure 13. A stem is a double stranded (paired) region whereas a hairpin loop is where the RNA folds back on itself. An internal loop is where a short unpaired region exists between two stems. If the internal loop is asymmetrical and only one strand forms a loop, while the other continues directly from one stem to the other, it is referred to as a bulge. In a multi-branched loop, several stems come together. A pseudoknot is a long range interaction, where a loop pairs with another region.

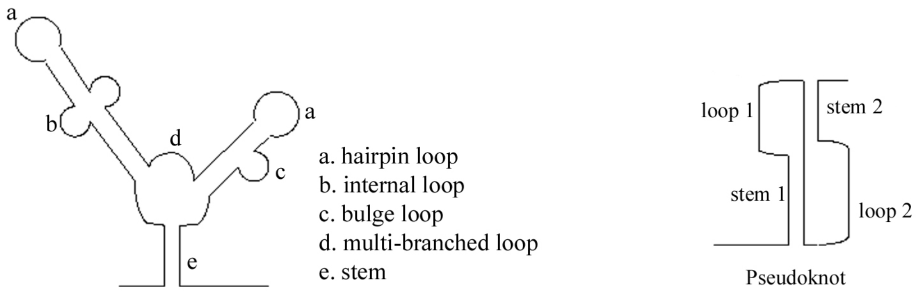


Figure 13. RNA secondary structure features.

Predicting RNA secondary structures from a single sequence is a formidable task as a simple sequence of 200 bases long has the potential to form 10^{50} possible base-paired structures (Durbin *et al.,* 1998). So there is a need to identify the correct structure from false and score them appropriately.

The simplest approach to predict secondary structures is to find the configuration with the greatest number of paired bases as defined by Nussinov (Nussinov, 1978). Testing and scoring each possible structure is numerically impossible and therefore a dynamic programming can be used to find an optimal solution. In the Nussinov algorithm this is done by extending a sub-optimal structure in four possible ways as shown in Figure 14.

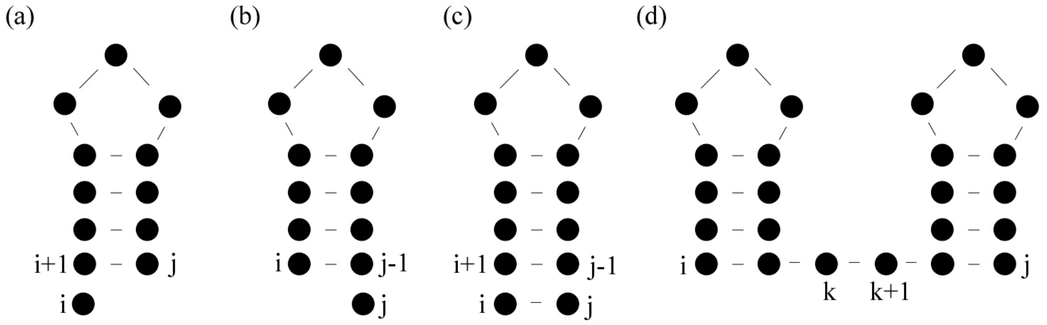


Figure 14. Four possible ways of extending a sub-optimal structure using Nussinov algorithm. (a) i unpaired (b) j unpaired (c) i, j pair (d) bifurcation.

(a) Add an unpaired base *i* to the best structure for the subsequence *i+1, j*

(b) Add an unpaired base *j* to the best structure for the subsequence *i, j-1*

(c) Add paired bases *i-j* to the best structure for the subsequence *i+1, j-1*

(d) Combine two optimal substructures *i, k* and *k+1, j*

A recursive equation for this extension of sub-optimal structure is represented as below –

$$\gamma(i,j) = \max\begin{cases} \gamma(i+1,j) \\ \gamma(i,j+1) \\ \gamma(i+1,j-1)+\delta(i,j) \\ \max_{i<k<j}\left[\gamma(i,k)+\gamma(k+1,j)\right] \end{cases} \tag{9}$$

where, $\gamma(i,j)$ is the score for the maximum number of base pairs that can be formed for sub-sequence $x_i,\ldots\ldots,x_j$ and $\delta(i,j)$ is the score of a base pair x_i and x_j. If x_i and x_j are complimentary, $\delta(i,j)=1$ else $\delta(i,j)=0$.

Although the model is simple, it requires several improvements. Firstly, the algorithm allows for hairpin loops of any length. In reality, RNA is not that flexible and a minimum of about 3 nucleotides is needed to form a hairpin. Secondly, in the scoring matrix, bases that lie on the diagonal correspond to the hairpin loops. Hence while traceback; any base-pairing solution in proximity to the diagonal should be prevented.

A further possibility of improving the Nussinov algorithm is to use *Stochastic Context Free Grammar* (SCFG) to generate a probabilistic model. The original algorithm is changed slightly to allow various probabilities in scoring and regarded as an adapted *CYK algorithm*. Details of this algorithm can be found in *Biological sequence analysis* by Durbin *et al*.

I implemented this algorithm and used it for identifying secondary structures that are responsible for transcription termination. The implementation also formed a part of the Eponine trainer for sampling secondary structure constraints in the stem-loop model explained in chapter 3.

2.6 Zuker algorithm

An improvement over Nussinov algorithm was later developed by using free energy parameters apart from base pair metrics. Energy parameters are included to score base-pair stacking, single dangling nucleotides, terminal mismatches and the lengths of hairpin loops, bulge loops, interior loops and multi-branched loops. This was aided with results from wet-lab experiments leading to different algorithms.

The first algorithm based on energy minimization using nearest neighbour energy parameters was attempted by Tinoco *et. al.*, and Delisi *et. al.* In this algorithm, free energies assigned to base pair stacks and loops and are summed to calculate the overall free energy difference of folding. Later new concepts like dynamic programming methods were incorporated and modified by many people. The popular among them is Zuker's *mfold* ('m' stands for 'multiple') program. The algorithm predicts a minimum free energy, ΔG, as well as minimum free energies for foldings that contain any particular base pair. The success of the program depends on the accuracy of the energy parameter for base pairs and recent versions use the free energy data from Mathews *et al.*, 1999 with the folding temperature of 37˚C and ionic conditions [Na^{+}] = 1M and [Mg^{++}] = 0M (Zuker, 2003).

The secondary structure is a list of base pairs, denoted by *i:j* for a pairing between the *i*th and *j*th nucleotides, r_i and r_j, where $i < j$ by convention. Generally only Watson-Crick base pairings and G:U wobble pair are treated as base pair rules. However exceptions exist. RNA has A-form helices and two helices are said to form a pseudoknot if base pairs *i:j* from one and *i':j'* from the other satisfy $i < i' < j < j'$ criterion. Pseudoknots are often excluded in the definition of secondary structures as current algorithms have difficulty in identifying them (Zuker, 2000).

Free energy minimization programs generally analyse a large ensemble of structures (called *suboptimal structures*) at different stages. To reduce this range, auxiliary information might be useful and *mfold* program employs base-pair metrics.

The base pair metrics defines RNA molecule as a collection of base pairs that occurs in its three dimensional structure. If R is represented as an RNA sequence then S is a set of ordered pairs, written as *i:j* ($1 \leq i < j \leq n$) satisfying these conditions -

1. $j - i > 3$
2. If *i:j* and *i':j'* are 2 base pairs, then either
   (a) $i = i'$ and $j = j'$ or
   (b) $i < j < i' < j'$ or
   (c) $i < i' < j' < j$ (This condition excludes pseudoknots).

The optimal structure will have the lowest free energy. Each of the various loops and stacked pairs will contribute a certain amount of energy to the secondary structure configuration. The energy of each base pair might be represented as $e(r_i, r_j)$ and the energy of the whole structure as $E(S)$ is then given by –

$$E(S) = \sum_{i,j \in S} e(r_i, r_j) \qquad (10)$$

Reasonable values of *e* at 37° are -3,-2 and -1 kcal/mole for GC, AU and GU pairs respectively. However to capture the destabilizing effects of various loops or the nearest neighbour interactions in helices and loops a more sophisticated algorithm is required.

So to achieve the minimum energy $E(i, j)$ for nucleotides, *i* and *j*, the following recurrence relation is used –

$$E(i,j) = \min\begin{cases} E(i+1,j)\cdots\cdots\cdots\cdots\cdots\cdots\cdots(Ia) \\ E(i,j-1)\cdots\cdots\cdots\cdots\cdots\cdots\cdots(Ib) \\ e(r_i,r_j)+E(i+1,j-1)\cdots\cdots(II) \\ \min\limits_{k=i+1}^{j-1}(E(i,k)+E(k+1,j))\cdots(III) \end{cases} \tag{11}$$

In this relation, there are totally $\phi(n^2)W(\cdot)$ matrices and each of them takes $\phi(n)$ time to calculate. Hence the running time of the algorithm is $\phi(n^3)$ and the memory requirement to store the W matrices is $\phi(n^2)$.

If *(i, i')* and *(j, j')* are two base pairs in the optimal pairing then,

1. *i < i' < j< j'*, i.e. the i pair precedes the j pair
2. *i < j < j'< i'*, i.e. the i pair includes the j pair

The first case will be handled by the condition III in the recurrence relation whereas the second case by conditions Ia, Ib and II.

The algorithm assumes constant energy for multi-branch loops and ignores single base stacking. If these issues are to be considered then some auxiliary information is added, at which point the algorithm gets complicated.

With the free energy determined for each base pair, a traceback algorithm (Zuker, 2003) is used to find the minimum free energy for the folding.

To predict RNA secondary structures in this project, I used the *Vienna RNAfold* package where Zuker algorithm is implemented.

2.7 Biojava and Bioperl

The open source bioinformatics toolkits Bioperl (http://www.bioperl.org) and Biojava (http://www.biojava.org) provided multiple functionalities that made my analysis much easier.

Bioperl (Stajich *et al.*, 2002) is the oldest and most downloaded distribution. It is a toolkit of PERL modules useful in building bioinformatics solutions in PERL language. The modules have a wide array of functionality, including codes for handling and indexing most popular bio-specific database and flat-file formats; for auto-generating bio-related graphics for web pages, classes and methods and for describing and manipulating biological sequences, annotations, trees, alignments and maps.

It is built in an object-oriented manner so that many modules depend on each other to achieve a task. I have extensively used the ‘Bio::Seq:IO’ and ‘Bio::Seq’ modules for accomplishing different tasks in this project.

Likewise, Biojava is another toolkit developed using Java language for analysing and presenting biological sequence data (for overview, Mangalam, 2002; Pocock, 2003; Pocock *et al.*, 2000). The toolkit has around 40 packages covering simple sequence manipulation to complex machine learning modules. The Eponine implementation is built using the Biojava package and hence in several cases I used relevant modules from Biojava to train and analyse the gene regulatory signals.

2.8 Databases

2.8.1 ENSEMBL

Various genome projects release DNA sequences into the public domain from throughout the world, making the subsequent task of assembling and annotating it difficult. ENSEMBL (http://www.ensembl.org, Birney *et al.*, 2004) is a joint project between EMBL-EBI and The Wellcome Trust Sanger Institute to develop a system which automatically tracks all the assemblies of a genome and annotates them by finding genes and other features of interest to biologists and medical researchers. This is done by taking sequences from the public domain and storing them in a large database. Automatic annotation using *pmatch* (Ohrt, 2004), *exonerate* (Slater) and *GENEWISE* (Birney and Durbin, 1997) to build genes from protein and mRNA evidence detects most genes and the results are published over the web based interfaces. A separate automatic prediction using EST evidence is carried out, although, the resulting gene structures are less reliable. Any match to the candidate genes in the public databases forms the ‘supporting evidence’ suggesting the annotations are accurate. All analysed data are stored in a relational database, which makes it easy to access. To facilitate the analysis process and access the results, ENSEMBL has created a set of PERL modules to connect to this database and query it. I have used these modules to access the data required for this project. ENSEMBL also has excellent web based interfaces and a sequence viewer (ContigView) that allowed me to add my own annotation of the region using the DAS protocol (Dowell *et al.*, 2001). This helped me to view my predictions along with other annotations available in the public domains.

2.8.2 VEGA

The Vertebrate Genome Annotation (VEGA) database (Ashurst, 2002) is a central repository for manual annotation of several finished vertebrate genome sequence. As the data is manually curated the quality of annotation is high. Curation is done on a clone by clone basis using a combination of similarity searches against DNA and protein databases as well as a series of *ab initio* gene prediction programs like GENSCAN (Burge and Karlin, 1997) and FGENESH (Salamov and Solovyev, 2000). Comparative genome analyses are also used for the annotation purposes. Thus the genomic features are added to the sequences based on supporting evidences.

Based on the evidence available, each annotated gene has been classified into the following categories –

(a) *Known* – Identical to known human cDNAs or protein sequences with an entry in LocusLink (Pruitt and Maglott, 2001) or GDB (Harger *et al.*, 2000).
(b) *Novel CDS* – Containing an open reading frame determined based on spliced ESTs and/or similarity to known genes/proteins.
(c) *Novel transcript* – Similar to novel CDS, however with an ambiguous ORF.
(d) *Putative* - Based on spliced human ESTs but without an ORF.
(e) *Pseudogene* – Similar to known proteins but with in-frame stop codons and/or frame shifts disrupting the open reading frame.

I used ‘Known’, ‘Novel CDS’ and ‘Novel transcript’ annotations from this database to extract sequences and features from human chromosome 20 and 13 for various analyses.

I also used annotation for human chromosome 22. Chromosome 22 is also available from VEGA, however as a result of it being the first human chromosome to be sequenced and annotated (Dunham *et al.*, 1999) the annotation has been extensively refined. The third generation gene annotation on chromosome 22 published in 2003 (Collins *et al.*, 2003) is one of the high quality data available for human genome sequences. For annotations, Expressed Sequence Tags (EST), comparative sequence analysis and wet-lab experimental verifications were used. Availability of this high quality annotation helped me to derive datasets for various aspects of the project.

2.8.3 RefSeq

The Reference Sequence (RefSeq) project (Pruitt and Maglott, 2001) run by the National Center for Biotechnology Information (NCBI) provides a collection of non-redundant DNA, RNA and protein sequences along with available information for those sequences. Non-redundancy is ensured by clustering identical or related sequences and representing one sequence out of each cluster. Based on the information available for a particular sequence, RefSeq records are available in four categories –

(i) *Genome annotation* – This category includes contigs, modelled mRNAs and corresponding modelled proteins.

(ii) *Predicted* – Predicted records represent genes of unknown function that are supported by full length mRNA, EST or homologous sequences.

(iii) *Provisional* – Records with known or inferred function not subjected to review.

(iv) *Reviewed* – Records with known function that are manually curated.

Reviewed RefSeqs are richly annotated with publications, gene description, UTR sequences, transcript variants and cDNA sequence removed of any vector or linker contaminating sequences. Hence in this project I used only reviewed records for analyses.

2.9 Other programs

I used several programs available in the public domain to compare with the performance of the models I created using Eponine. Describing all of them is beyond the scope of this chapter and so I will limit myself in briefly explaining them when and where required.

However here, I will give a few details about the *ERPIN* poly(A) prediction program used to compare with my transcription termination model and the GAZE method used to predict genes with Eponine model features.

2.9.1 ERPIN

ERPIN (*E*asy *R*NA *P*rofile *I*dentificatio*N*) is an RNA motif search program developed by Daniel Gautheret and Andre Lambert (Gautheret and Lambert, 2001). ERPIN reads a sequence alignment and secondary structure, and automatically infers a statistical 'secondary structure profile' (SSP). A dynamic programming algorithm is used to scan any target sequence with this SSP to find matches and score them. SSP profiles are constructed using two weight matrices – one for single strand regions in the given sequence and another for helical regions.

Helix profiles are 16-row matrices with a lod-score for each possible base-pair, while single strand profiles are generally five-row matrices with lod-scores for the four bases and the gap character. For a helix of size *n*, the profile has 16 rows and *n* columns in the matrix. For a single-strand of size *n*, the profile has five rows and *n* columns. The lod-score for a base at position *i* is given as –

$$S_i = \log\left(\frac{O_i}{E_i}\right) \tag{12}$$

where, O_i and E_i are the observed and expected frequencies respectively for the base at position *i*. ERPIN treats gaps as another base rather than issuing penalties as done in sequence alignment.

Likewise, a lod-score for each base-pair at position *i* and *i+1* (consecutive bases) is given as –

$$S_{i,i+1} = \log\left(\frac{O_{i,i+1}}{E_i \times E_{i+1}}\right) \tag{13}$$

where, $O_{i,i+1}$ is the observed frequency for the base-pair at position i and $i+1$ and E_i, E_{i+1} are the expected frequencies of individual bases.

The helix profiles capture both Watson-Crick base pairs and non-canonical base pairs, however, gap character is not allowed in helical regions.

In this project I used an ERPIN model, trained to predict poly(A) signals with the following command line –

```
erpin polya.epn <database file> 2,3 -umask 2 -umask 2 3 -cutoff 70% 74% -
unifstat –smp
```

I then compared the predictions of ERPIN with those of Eponine transcription termination model.

2.9.2 GAZE

GAZE (Howe *et al.,* 2002) is a gene prediction tool that assembles evidences of gene components or features into complete gene structures. The gene features and the model structures are supplied by the user making it completely configurable.

As described earlier, almost all the gene prediction methods first do an extensive search for signal and content information’s on the sequence to identify gene components. However, they differ in the subsequent mechanism of integration of this information’s to predict the gene structure. This unified two step process adapted by various programs introduces an inherent rigidity in extending them to incorporate new knowledge about gene structures. GAZE tackles this by separating the two steps and allowing to build a customised version of the *ab initio* gene prediction system with user defined features. In that way, GAZE is not tied to any specific signal or content sensors as well. Another key feature of GAZE is that it does not work directly with genomic DNA sequence. Instead it predicts gene structure from an input file with signal and content information marked in GFF format (WTSI). The configurations for integrating the features to form the gene are defined in another input file in XML format. Thus GAZE is a generic system that uses dynamic programming to obtain the highest scoring gene structure based on external features and configurations.

The algorithm also has a run time effectively linear with the length of the sequence without compromising accuracy.

The gene structures are scored by taking a list of features ordered by their sequence position and the rules defined in the configuration file. For example, for a sequence of 1000 bp long with the features – transcription start site @ 100 bp, donor site @ 250 bp, acceptor site @ 500 bp and Poly(A) signal @ 750 bp and a configuration allowing a gene to be formed with or without introns can lead to 2 gene structures. One, a single-exon gene without taking donor and acceptor sites and another with an intron defined with donor and acceptor site. A score of each of these gene structures are assigned and the highest scoring gene structure is defined to be the most probable given the features and configuration. Mathematically this is represented in the following equation -

$$E(\phi) = \sum_{i=0}^{n} \mathrm{Re}\, g_{t(\phi_i) \to t(\phi_{i+1})}(l(\phi_i), l(\phi_{i+1})) + g(\phi_{i+1}) \qquad (14)$$

where, $t(\phi_i)$ defines the type of feature, ϕ_i and $l(\phi_i)$ is the location of the feature, ϕ_i, in the sequence and $g(\phi_i)$ is the respective score of the feature. $\mathrm{Re}\, g_{t(\phi_i) \to t(\phi_{i+1})}$ represents the region score for interval (ϕ_i, ϕ_{i+1}) bordered on the left and right with the types of the features, referred as source (*src)* and target (*tgt)* features.

The 'target' feature and its potential origins ('source' features) define the rules of the gene structure model. In the above example, the target feature, Poly(A) signal can be preceded by an acceptor site or transcription start site and thus allowing for 2 gene structures to be built. However, an acceptable gene structure cannot be formed by allowing Poly(A) signal preceded by the donor site and thus defining a set of rules how structures can be built. Additional constraints, given below, can also be added to these rules to define more stringency.

(a) *Distance constraint* specifying the length of the segment defined by source and target features.

(b) *Phase constraint* specifying the source and target features that should occur - 0, 1 and 2 bases apart.
(c) *Interruption constraint* specifying an illegal occurrence of a feature between source and target.
(d) *DNA constraint* specifying an illegal occurrence of a DNA sequence between source and target.

A length penalty function and segment qualifier defined by these constraints add to the final score of the gene structure. The highest scoring gene structure with these rules and constraints is obtained by using dynamic programming. For more details about the scoring and algorithmic issues, please refer to this thesis (Howe, 2003).

Taking advantage of the user configurable GAZE system, in this project, I used Eponine model features to predict gene structures with the rules and constraints defined in Appendix C. Two configurations with and without translation features are used in predicting Eponine based gene prediction. I employed *phase constraint* in gene configuration with the translation feature but no *distance constraint* in both the configurations thus kept no restrictions on the maximum length of exons and introns.

2.10 Concluding remarks

In this chapter I have given an overview of all the strategies, tools and databases used to find gene regulatory signals in this project. In the following chapters I will explain in detail how the package was employed to derive transcription termination, translation start and stop and splice site sequence models. Also apart from the sequence model from Eponine, in chapter 3, I have explained the results of Nussinov and Zuker algorithms used to search for stem-loop structures in the 3’ end of the genes. I implemented Nussinov algorithm using PERL modules for this purpose. RNAfold implementation of Zuker algorithm in Vienna package was used to find stem-loop structures based on free energy metrics.

MODELLING TRANSCRIPTION TERMINATION SIGNALS

3.1 Introduction

My initial objective here was to develop a transcription termination model and thereby determine the correct gene 3’-end. In general when annotating a genome, poly(A) signals and/or cleavage sites are generally considered to be the end of a gene. Almost all the gene prediction methods available so far use this principle. However predictions of poly(A) signals along the chromosomes are likely to be dense, as the probability of occurrence of the hexamer in 3 billion human sequences is high. Existing gene prediction programs screen out false poly(A) predictions by conditioning it on numerous other parameters like coding potential, exon-intron structure and ORF length. As here the objective is to construct an *ab initio* gene model purely based on gene regulatory signals without considering such parameters, it is imperative to consider other gene 3’-end motifs apart from the poly(A) signal to limit false predictions.

In vivo and *in vitro* experiments to identify such 3’-end consensus motifs responsible for transcription termination have not been so far successful as discussed in chapter 1. However these experiments have established that the poly(A) signal and auxiliary sequences are essential for efficient termination of RNA polymerase II.

Extensive computational analysis to identify transcription termination signals has not previously been attempted. Gene 3’-end identification programs, such as *ERPIN*, *Polyadq* and *PolyAH* depend only on poly(A) variants and auxiliary motifs. Here I discuss the results of the algorithms with which I tried to detect additional RNA polymerase II transcription termination signals.

3.2 Datasets

Given the difficulty in experimentally annotating the poly(A) signal and cleavage site of each gene in the human genome, extracting a large dataset with precise location of gene 3’-ends has been a challenge. However with the recent publication of the latest version of human chromosome 22 (Collins *et al.,* 2003), a set of 422 genes with high quality annotation was obtained. The methodology adapted for annotating chromosome 22 included three main approaches –

First, a *variety of programs* with different statistical methods were used to predict gene structures cautiously, as these methods are likely to produce incorrect and over predictions. Second, a match of *Expressed Sequence Tags* (EST) of transcribed genes with the genomic sequence gave direct evidence of expressed genes. Third, *comparative genomics* with related species helped to identify conserved gene structures.

Gene structures were identified using evidence from transcribed sequences across their entire length. Full length cDNAs or assembled ESTs were aligned to genomic DNA to resolve the splice sites and confirm 3'-ends. A 3' end was judged confirmed if it had a run of at least four adenine residues at the 3' end of cDNA/EST not present in the genomic sequence. Thus the 3'-ends with the processing signals were manually verified and confirmed for *in vivo* biological function.

Based on further evidence, this set of entries with confirmed 3'-ends were classified into *complete protein coding genes*, *partial genes*, *non-coding genes* and *pseudogenes* with the following definitions.

(1) A *complete protein-coding gene* has sequence identity to human cDNAs or ESTs across its entire length and a predicted ORF of at least 300 bases.

(2) A *partial gene* had sequence similarity to cDNA, EST or peptide sequence but did not comply with complete gene criteria.

(3) *Non-coding RNA genes* included small RNAs and published complete genes that did not contain ORF of at least 300 bases.

(4) A *pseudogene* had similarity to a known gene or protein but had evidence of disrupted function.

Using these criteria, 393 complete protein coding genes, 153 partial genes, 31 non-coding transcripts, 234 pseudogenes and 125 IGLV and J gene segments (Ig gene segments) were annotated for chromosome 22. Among all these categories, 376 protein coding genes, 56 partial genes and 15 non-coding transcripts are found to have confirmed 3'-ends with the hexamer variant and cleavage site. Out of these 447 genes (376+56+15), I extracted sequences from 422 genes in the interval of -200 to +2000 bases relative to the cleavage site to form the *positive dataset*. The remaining 25 sequences (447-422) had an overlapping transcript within the 2000 base pair downstream sequence of the cleavage site. A set of 22 sequences from these 422 entries were set aside and used as an independent test set, leaving the remaining 400 sequences for training purposes.

For training Eponine Anchored Sequence (EAS) models, the RVM requires a *negative dataset* that does not have 3'-end processing signals. Choosing an appropriate negative set for training purposes is a determining criterion for making sensible Eponine models. So I extracted sequences from different sources and will briefly explain these sets while discussing the Eponine sequence models.

3.3 Nucleotide composition analysis

Figure 15 shows the average base composition of 422 sequences for 200 bases upstream and 2000 bases downstream of the cleavage site. The undulations in the graph are seen concentrated near to the cleavage site then in other regions of the sequences. The zoomed figure (Figure 16) with base compositions for -100 to +50 bases from the cleavage site shows two significant peaks for adenine nucleotide distribution. The first broad peak spans -30 to -5 base pairs while the second, peaks at position 0. Followed by the cleavage site the adenine concentration suddenly decreases with increase in thymine composition. The thymine level represents the U-rich sequence observed near cleavage sites. The guanine and cytosine concentration is generally relatively low and more equal to the background distribution.

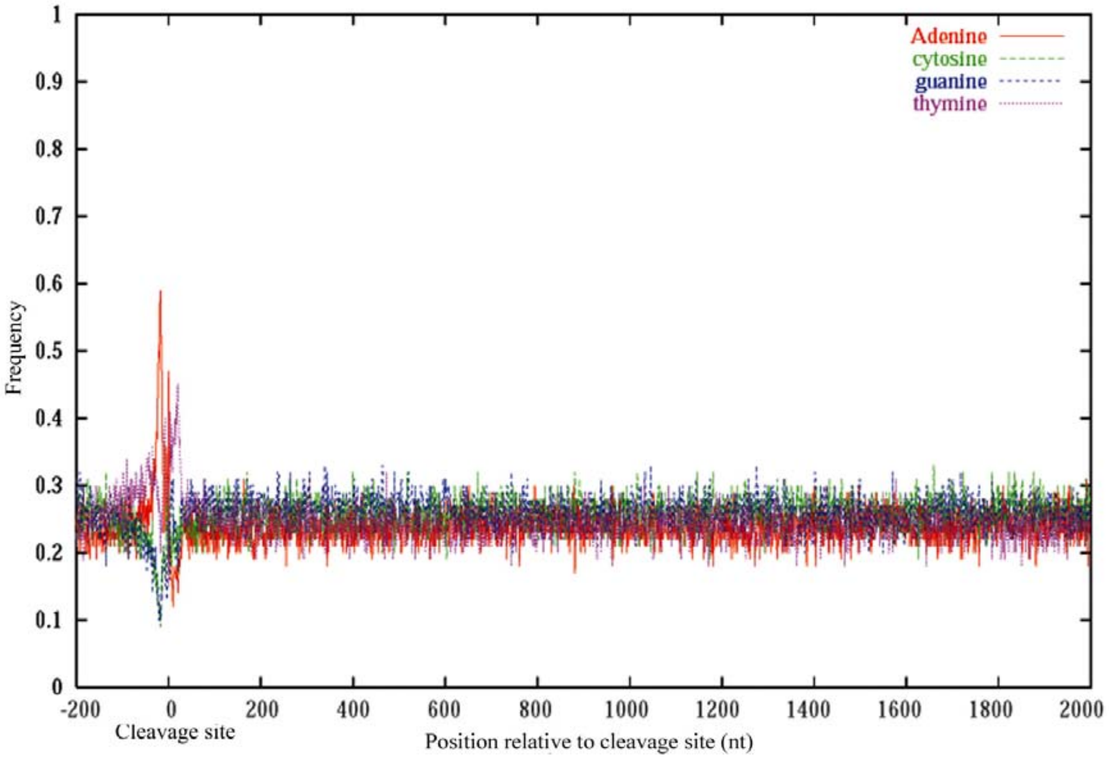


Figure 15. Nucleotide composition spanning -200 to 2000 bases relative to the cleavage site

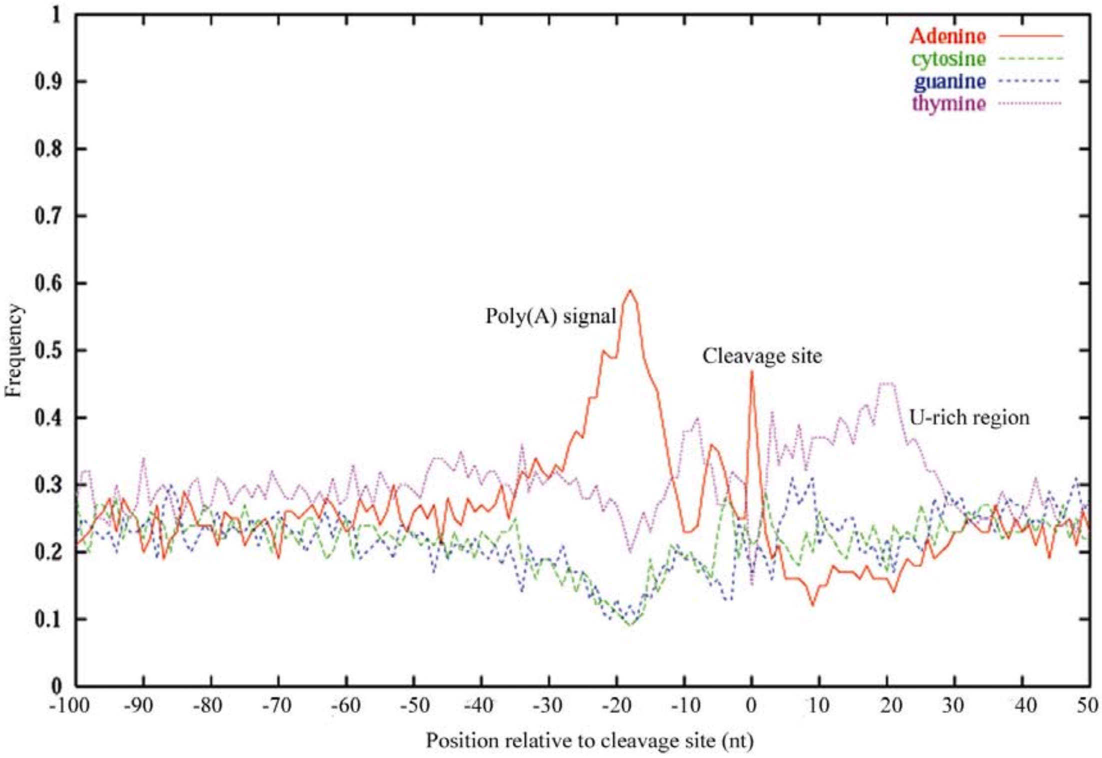


Figure 16. Nucleotide composition spanning -100 to 50 bases relative to the cleavage site

There is no difference in base composition in the sequences 50 to 2000 bases downstream of the cleavage site (Figure 15). The termination signals responsible for RNA polymerase pause and release are expected to be present in this region. However the graph shows no difference in the nucleotide distribution from background. This might be due to the un-alignable nature of termination motifs as they are known to occur at varied positions for different genes (Dye and Proudfoot, 2001). Even if there is a small bias in nucleotide composition for single sequences, by averaging, no bulk effect is seen.

Thus the nucleotide composition analysis has identified the 3’-end processing signals known previously. However I find no significant compositional variation in the downstream region where the polymerase pauses before terminating.

3.4 Secondary structure analysis

As the simple nucleotide composition analysis does not reveal the pause signal at the downstream region, I decided to search for stem-loop secondary structures known to play a major role in prokaryotic transcription termination (Henkin, 1996). In the eukaryotic genome, stem-loop structures and their role in terminating transcription has been established for histone genes. Now the question is, whether similar structures are present in eukaryotic protein coding genes or not. If present, stem-loop structures are likely to be found at the downstream sequences of the cleavage site where actual pausing of RNA polymerase II occurs. This differs from histone stem-loops as in these genes the structure was found upstream of 3’-end processing signals. As explained in chapter 1, special proteins like SLBP bind to these structures, to stabilize them and hinder poly(A) tailed histone mRNA formation. Unlike this, a potential structure at the downstream region in the protein coding gene might explain the drag, pause and queuing of the polymerase before termination. So to search for any stem-loops I used two simple algorithms developed by *Ruth Nussinov* and *Zuker* (Nussinov, 1978; Zuker, 1994).

3.4.1 Nussinov algorithm

Nussinov algorithm (Nussinov, 1978) is one of the simplest approaches to predict secondary structure of RNA molecules. The method is based on finding the configuration with the greatest number of paired bases. Apart from normal Watson-Crick base pairing used for calculation, *G:U* pairing is also allowed. Since testing and scoring each possible structure is computationally expensive, the algorithm uses a dynamic programming to find an appropriate solution. For details of the algorithm and implementation refer to chapter 2.

I scanned the 422 sequences in the positive dataset with the overlapping window sizes of 15, 25, 35 and 50 bases with an interval of 5 and 10 bases between each window. As explained in chapter 2, the initial algorithm was modified to consider the loop length parameter. The base pair metrics score was calculated for each window allowing at least 3, 5, 7 and 10 bases in the loop region. Figure 17 shows the average metrics score of the sequences scanned with overlapping window sizes of 60 bases, 10 bases between each window and loop length of 5 nucleotides.

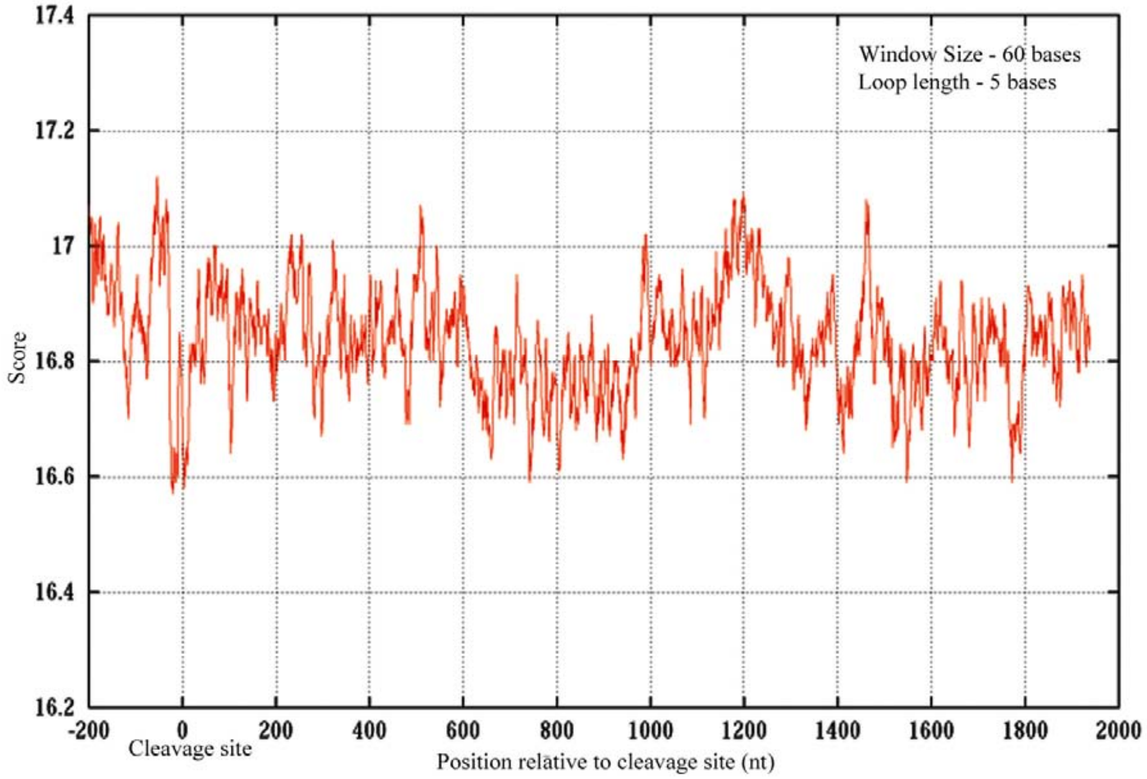


Figure 17. Averaged score values of sequences around cleavage site calculated using Nussinov algorithm

The X and Y-axis represents sequence length and score respectively with the cleavage site referred to as position 0. The graph shows the scores are spread out evenly with no significant peaks or troughs. This might be due to the following reasons –

(a) no stem-loop structure in protein coding genes.

(b) the algorithm is simple and observing the base pair maximization metrics alone might not be sufficient to detect any secondary structure.

(c) the algorithm does not consider non-Watson-Crick pairing except *G:U* base pairing.

(d) base stacking metrics that explains the structure stabilization is not used.

(e) position of stem-loops might be varied for each gene and averaged metrics score did not show any bulk effect.

The result emphasis the need for an algorithm with additional metrics and thus I used the one published by Zuker.

3.4.2 Zuker algorithm

Zuker algorithm (Zuker and Stiegler, 1981) is one of the most well-known algorithm to predict RNA secondary structures based on the free energy minimization principle. The algorithm is optimal for predicting secondary structures of RNAs with no pseudoknots. According to the algorithm, all the structures of RNAs can be decomposed into either sequential or nested structures. This algorithm is implemented in the well known RNA secondary structure programs like *RNAfold*, *mfold* and *ViennaRNA*.

Here I used the *ViennaRNA* implementation to look for the presence of secondary structures in the positive dataset. The algorithm calculates free energies for all possible structures and determines the one with the least value to be the most probable structure. The lower the predicted free energy value, the more likely the structure is thermodynamically stable and likely to persist.

I scanned the 2200 base sequence with overlapping window sizes of 20, 30, 40, 50, 60, 70 and 80 bases, skipping 5 or 10 bases between each window. The free energy scores for each window size (with a gap of 10 bases between windows) along the length of the sequences are plotted in Figure 18.

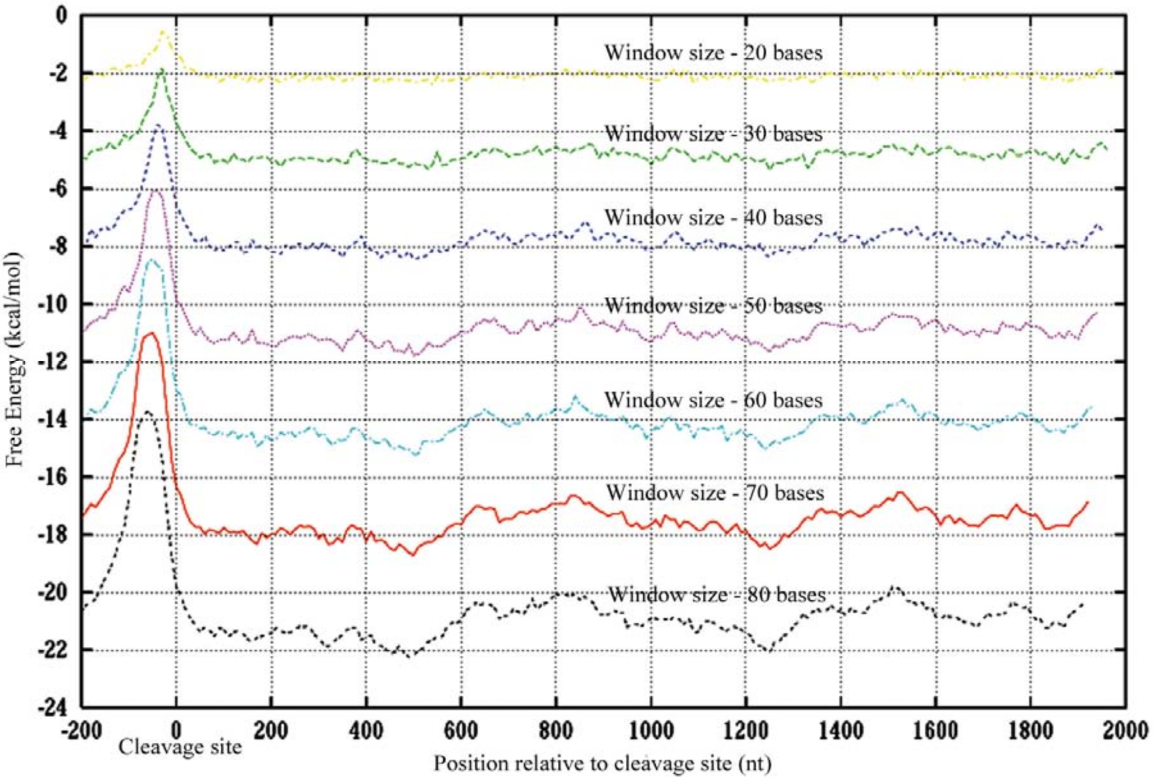


Figure 18. Averaged free energy values of sequences around cleavage site calculated using Zuker algorithm

The Y-axis represents free energy values measured as *kcal/mol* while the X-axis shows sequence length as *nucleotides*. Here, I analyse the plot by splitting it into four regions and comparing the energy values to the average value of the negative set of sequence.

(i) Region 1 extending from -200 to 0 bases shows a statistically significant peak at the poly(A) region. The higher energy value means there is less probability of a secondary structure at this region.

(ii) Region 2 from 0 to +100 bases from the cleavage site covering the U-rich sequences of the 3’-end processing signals has a free energy value less than the average at 95% confidence level. However the significance of the energy scores is not prevalent when the window size is reduced to 20 bases.

(iii) Region 3 comprising +100 to +650 bases from the cleavage site shows the energy values are less than the average value at 99% significance level when the sliding window parameter was fixed at 60 bases. The condition was found true even at lower window sizes. This shows there is possibility of a RNA secondary structure in this region. However the broad distribution of free energy scores indicate the possible stem-loops are distributed at varied positions for different genes. On averaging energy values for the 422 sequences at this region, the overall distribution is likely to get flattened rather than appearing as a sharp trough. This agrees with earlier understanding of the presence of termination related pause signals at varied distances from cleavage site (Dye and Proudfoot, 2001).

(iv) Region 4 from +950 to +1350 shows a decrease in free energy values from the average at 95% significant level in the window size of 60 bases. However the significance score reduced with diminishing window sizes indicating there might not be any secondary structure in this region.

The free energy parameters used for calculating the scores were derived from recent experiments and the values are likely to be influenced by sequence artifacts. Hence, I have plotted the GC and GT densities along the sequence calculated with the 60 bases sliding window (Figure 19 and Figure 20). GC and GT richness in the sequence affect DNA base pairing and emit low free energy values. These low energy values based on the sequence composition bias need not necessarily mean there is a RNA secondary structure. So to avoid this misinterpretation, I did correlation studies of GC and GT density with free energy values of the corresponding regions.

For region 1, I found a strong negative correlation between the free energy values and GC and GT density. Guanine and cytosine compositions are expected to be less around this region as the poly(A) signal and the cleavage site increases richness in A and somewhat in T density. This increased adenine and reduced guanine and cytosine concentration might be the reason why the free energy values in this region are remarkably high.

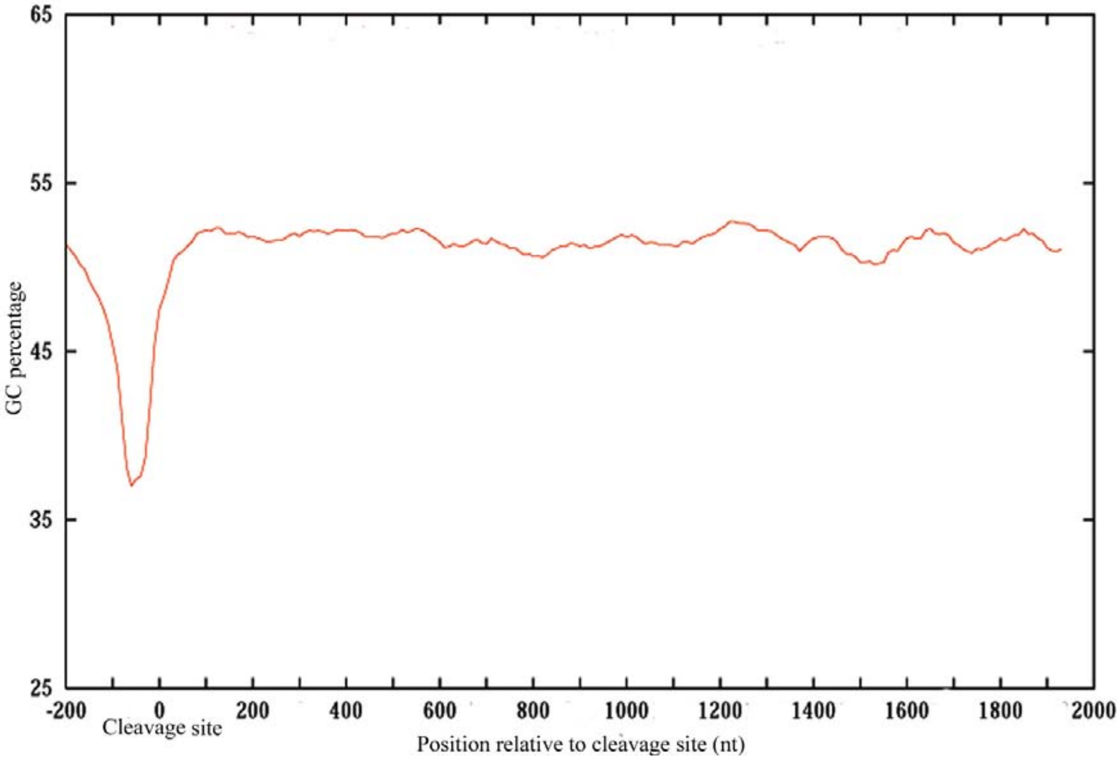


Figure 19. Percentage of GC residues in the sequences around cleavage site

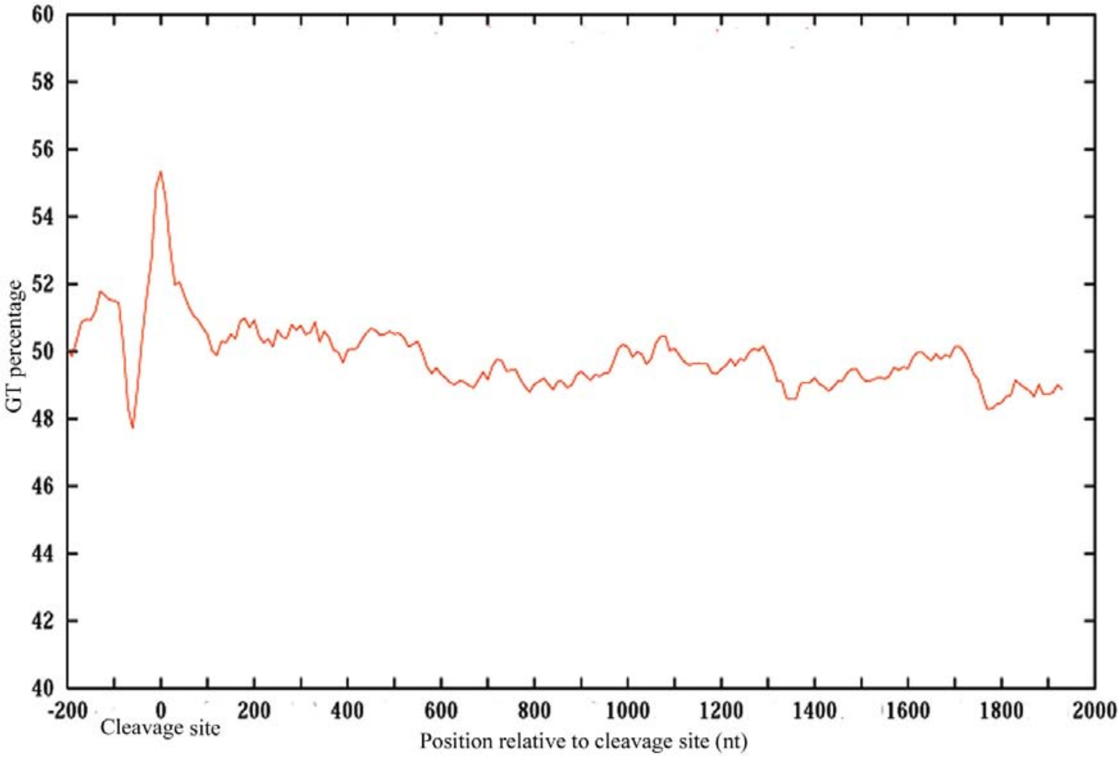


Figure 20. Percentage of GT residues in the sequences around cleavage site

Region 2 does not have any correlation between GC densities and free energy values, whereas GT density has a strong positive correlation coefficient. This is due to the U-rich sequence of the 3’-end processing signals followed by the cleavage site. Thus the energy values at this region are influenced by the base composition.

Region 4 has positive correlation between GC density and free energy values, whereas the GT density does not show any. Thus the lower than average free energy found at a 95% confidence level in this region might be due to the influence of GC base stacking.

Unlike other regions, sequences in region 3 have no correlation between GC/GT densities and free energy metrics scores. So the likely secondary structure present between +100 to +650 bases may not be due to nucleotide composition bias. However the significance of lower free energy values decreases with the size of the sliding window. Hence the results show there is scope for stem-loops in the region, however detailed biochemical experiments are required to confirm their existence.

Thus the Zuker algorithm predicts there is a possibility of RNA secondary structure from 100 to 650 bases from the cleavage site and that they are less likely to be caused by sequence artefacts. Further experiments can confirm the presence of any such structure and help us to understand the polymerase pause and release from the DNA.

3.5 Eponine transcription termination model

With the results from nucleotide composition and secondary structure analysis, I then resorted to Eponine described in chapter 2 to train a transcription termination model. Eponine is a probabilistic sequence classifier based on a relevance vector machine and requires a set of positive and negative sequences to learn informative basis functions to classify them. As explained earlier, the major positive dataset for training an Eponine model was derived from human chromosome 22. Similarly the major negative dataset was extracted from random sequences from the transcription units. However, apart from these datasets, various others were also used. Here I explain in detail how the model was constructed and cross-validated. Following it, I compare the performance of the model to the poly(A) prediction program, *ERPIN*. The Eponine model, apart from detecting the annotated gene 3’-ends, made other predictions, which are referred to as *false positives*.

Towards the end of this chapter I explain the distribution of these false positives and two hypotheses explaining the possible role of such predictions in gene regulation.

3.5.1 Training the transcription termination model

One important criterion while learning any classification model is to choose an appropriate negative set. This set of sequences forms the basis for differentiating it from the positive set provided for training. Here I attempted different sets of negative sequences which are explained below –

(i) I extracted sequences of 2200 bases each from chromosome 1 by choosing random points. A pseudo-random number was generated using PERL *rand()* function and 2200 bases from that point was dumped from ENSEMBL database in FASTA format. Similarly, another set of random sequences from chromosome 20 was extracted by generating a number between 1 and 62 x 10^6, as the length of chromosome 20 is approximately 62 mega bases.

(ii) Another set of random sequences were extracted from chromosome 1, as described earlier, after repeat masking of the whole chromosome using *Repeat Masker* (Smit and Green, 1996). Random regions of the chromosome are chosen in such a way as that no repeat masked bases were part of the 2200 base negative sequence.

(iii) RNA polymerase is not expected to terminate in exon sequences and thus another set of negative sequences was derived from chromosome 22 exons. With the quality annotation available, sequences of all the exons annotated in chromosome 22 were extracted after leaving 100 bases near the donor and acceptor splice sites. These sequences were then concatenated together to form a single sequence. Then as explained previously for random sequences, a set of sequences of 2200 bases each were randomly dumped from this single sequence.

(iv) Similarly intron sequences were extracted from all annotated last introns in chromosome 20 and 22. Intron sequences from these two chromosomes were dumped after removing the 100 bases near the donor and acceptor splice sites. This is done to avoid representing the splice signals in the negative dataset. A set of sequences, each of 2200 bases in length, was randomly picked from the concatenated intron sequences.

(v) With the gene annotation of chromosome 22, sequences from the transcription units (including exons and introns) were extracted after leaving 250 bases near to the gene 3' end. The extracted sequences were concatenated and from it, 422 random sequences of 2200 bases each were dumped to form a negative set. Although weak terminators or pause signals are likely to be present in the transcription unit and thus in the negative set, they formed one of the best training sets for learning transcription termination models.

These different sets of negative sequences along with the positive dataset were used for training the transcription model. In each case of training an equal number of positive and negative sequences were used. The cleavage site formed the anchor point for the EAS model. The models were trained for approximately 10000 cycles using the VRVM trainer as described in chapter 2. At various points in the training process, I dumped 'checkpoint' models to identify the basis functions learnt. Initial models picked more basis functions and as training progressed they gradually converged leaving fewer basis functions explaining the dataset. One such final model is shown in Figure 21. The models have generally two sets of basis functions. The positively weighted position constraints, represented in black and negatively weighted position constraints, coloured in blue. The positive constraints are cases where the motif presence is likely to determine the termination site, whereas negative constraint makes the possibility less likely.

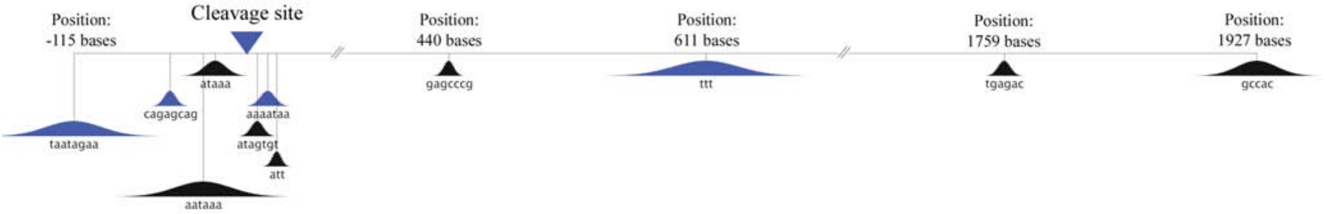
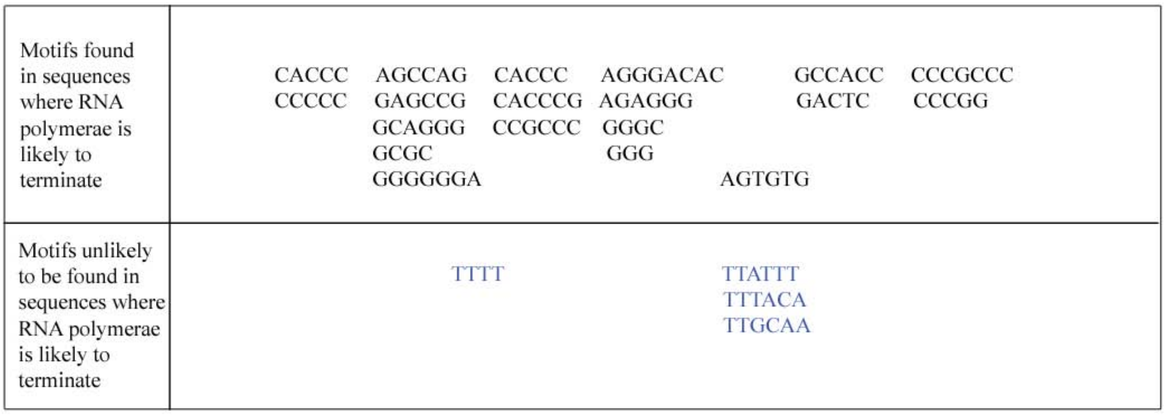


Figure 21. Transcription termination model trained from chromosome 22 sequences

Various models gave a consistent set of motifs learnt from the datasets. The poly(A) hexamer, AAUAAA was represented in the model around -20 to -30 bases as expected.

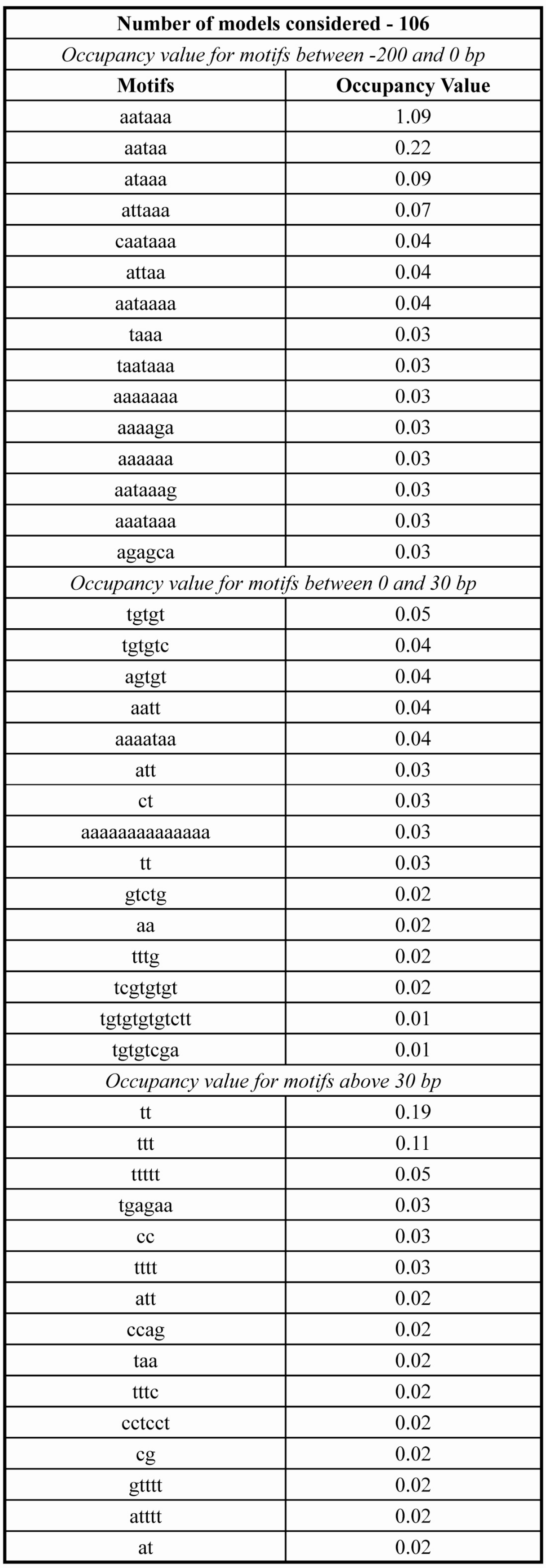
Similarly the GT/T-rich region found earlier through various studies can be found in the model immediately after the anchor site. The most interesting part I found in the models is the consistent appearance of the downstream motifs, which hereafter I refer as *Pause Elements* (PE), as such elements had earlier been shown experimentally to pause transcribing polymerase (Aranda and Proudfoot, 1999; Enriquez-Harris *et al.*, 1991; Yonaha and Proudfoot, 1999). The motif's positions varied with different models and training, however the sequence of the motifs did not change much. Various motifs found in these regions are listed in Table 1. Although there is diversity in the motifs, the positive constraints can be generalized into two types - Poly-C variants (CCCC, C with intermittent A or G) and GGAGG variants (GGAGG with intermittent A and C). Negative constraints as explained above are motifs not expected in the termination region and thus a non-T rich motif or its variants (with intermittent A/C) are likely to be present downstream of the cleavage site. The stretch of non-T residues in the downstream region is interesting as prokaryotic and polymerase I termination are likely to occur with a run of T residues. However the model indicates polymerase II termination requires the opposite and this agrees with results from experiments done on human β-globin and α2 globin genes (Dye and Proudfoot, 2001; Enriquez-Harris *et al.*, 1991). Table 2 lists the occupancy value of top 15 motifs learnt in different runs of training in a scale of 0 to 1. A value more than 1 indicates the motif is represented more than once in few models.

Table 1. Consensus motifs found in sequences between 50 and 2000 bases from cleavage site



Motifs found in sequences where RNA polymerae is likely to terminate	CACCC AGCCAG CACCC AGGGACAC GCCACC CCCGCCC CCCCC GAGCCG CACCCG AGAGGG GACTC CCCGG GCAGGG CCGCCC GGGC GCGC GGG GGGGGGA AGTGTG
Motifs unlikely to be found in sequences where RNA polymerae is likely to terminate	TTTT TTATTT TTTACA TTGCAA

Table 2. Occupancy value for motifs detected in the transcription termination models.



Number of models considered - 106	
Occupancy value for motifs between -200 and 0 bp	
Motifs	**Occupancy Value**
aataaa	1.09
aataa	0.22
ataaa	0.09
attaaa	0.07
caataaa	0.04
attaa	0.04
aataaaa	0.04
taaa	0.03
taataaa	0.03
aaaaaaa	0.03
aaaaga	0.03
aaaaaa	0.03
aataaag	0.03
aaataaa	0.03
agagca	0.03
Occupancy value for motifs between 0 and 30 bp	
tgtgt	0.05
tgtgtc	0.04
agtgt	0.04
aatt	0.04
aaaataa	0.04
att	0.03
ct	0.03
aaaaaaaaaaaaaa	0.03
tt	0.03
gtctg	0.02
aa	0.02
tttg	0.02
tcgtgtgt	0.02
tgtgtgtgtctt	0.01
tgtgtcga	0.01
Occupancy value for motifs above 30 bp	
tt	0.19
ttt	0.11
ttttt	0.05
tgagaa	0.03
cc	0.03
tttt	0.03
att	0.02
ccag	0.02
taa	0.02
tttc	0.02
cctcct	0.02
cg	0.02
gtttt	0.02
atttt	0.02
at	0.02

Also the position constraints found in the sequences 2 kb downstream of the cleavage site are repetitive (Table 1). This emphasize that the signals are present in multiplex and effective termination might depend on the cumulative effect of all the signals, agreeing with the experimental results found earlier (Aranda and Proudfoot, 1999).

3.5.2 Window size

I focused on a 2 kb window downstream of the cleavage site for modeling as the detailed experiments in human β-globin and ε-globin genes showed 2000 bp is enough for termination of polymerase II (Dye and Proudfoot, 2001). However I also tried sequences of varied lengths - 500, 1000, 1500, 2000, 3000 and 4000 bases downstream of the cleavage site. As I increased the downstream window size from 500 to 2000 bases the performance of the models improved. However there was no improvement when the window size moved from 2000 to 4000 bp. This suggests that 2000 bases are enough for termination in most cases. What signals found downstream in the 2000 to 4000 base sequences appear to be repetitions of the positive and negative constraints explained before. Even if a larger window is included, the trainer did not learn any position constraints beyond 2600 bases from cleavage site and thus model a compact transcription termination region.

3.5.3 Cross validation

As explained above, the inherent problem with most comparison based classifiers is the use of a proper negative dataset sequence while training. The motifs detected by the models may represent biases in the negative dataset used, rather than signals in the positive dataset. So to cross check if the position constraints discussed in the earlier models are consistent with signals in the positive dataset, I used different negative datasets for the training. Different training parameters gave similar motifs and the positions of the motifs are also found to be conserved. This suggests that the motifs learnt by the classifier are not biased by the training datasets used and are conserved in the sequences and may have some biological function.

The positive dataset for the model discussed above is derived from chromosome 22. To cross validate the PE detected by the earlier model and to provide evidence that they are not just due to some strange distribution of chromosome specific sequences; new models were trained using a positive dataset from chromosome 20. Gene annotation from VEGA database was used to extract 200 bases upstream and 2000 bases downstream of the cleavage site from chromosome 20. One of the models trained with this dataset is given in Figure 22. The signals detected by this training are similar to the motifs described in Figure 21 and Table 1. The similarities between these two independently trained models suggest that these PE motifs may be a general feature of human gene 3’-ends.

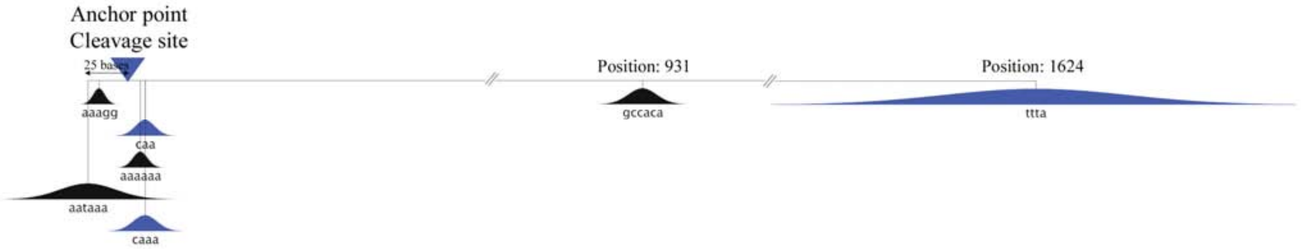


Figure 22. Transcription termination model trained from chromosome 20 sequences

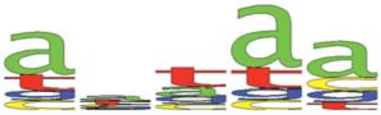
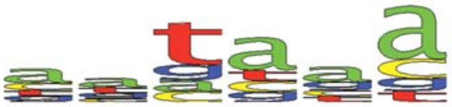
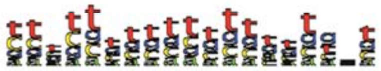
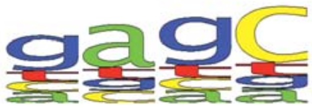
To investigate whether the PE motifs detected by the model were likely to be due to repetitive elements found in the human genome, the positive and negative datasets were repeat masked using *Repeat Masker* and training was done on a masked set. Models from various training cycles still learnt the PE discussed above. Thus the pause elements are not part of any repeat sequence although they occur multiple times in the 2000 bases downstream of the cleavage site.

Chromosome 22 has a higher GC content than the average for human chromosomes and the PE represented in the model might mimic the CpG island in the training datasets. To rule out this possibility, the positive dataset was scanned for CpG island using *CpG Report* (Micklem) with the default parameters of 100 bases sliding window, minimum length of 200 bases CpG island, 0.6 ratio for observed-expected value and 50% of G and C composition in the window. The scanning found that only 62 sequences out of 422 sequences had CpG island-like sequences even at a conservative threshold score of 80. Thus with the maximum of only 14.69% of the positive dataset containing CpG island-like sequences, the likelihood of learning a CpG motif is less. Hence the PEs are not mimics of CpG island signals.

3.5.4 Model refinement

Different training cycles showed the PE sequence motifs are present at varied positions downstream of the cleavage site. Hence I decided to try to refine the models by optimizing parameters to better by capture the position distribution of the PE. Important among the parameters modified is the Gaussian distribution width. As explained in chapter 2, the position of the basis function relative to the anchor point is captured as a Gaussian distribution. Since the default width allowed for the maximum expansion of the Gaussian distribution was found not to be ideal for learning the multiplex motifs of the PE, the width was optimized with various training cycles. Models trained with the new parameters gave PE with wider distribution and higher constraint weights. One such model is shown in Figure 23. The parameter file used to create the model is given in Appendix C. The new optimized parameters also removed the negative constraints present in earlier models and replaced them with positive constraints. Table 3 lists the positional constraints and Gaussian widths of the motifs seen in Figure 23.

Table 3. Position contraints learnt while training chromosome 22 sequences

MOTIFS	POSITION	CONSTRAINT WEIGHT	GAUSSIAN WIDTH
	-29	13.78	17.24
	-22	7.43	7.12
	8	9.72	0.73
	309	20.80	123.22

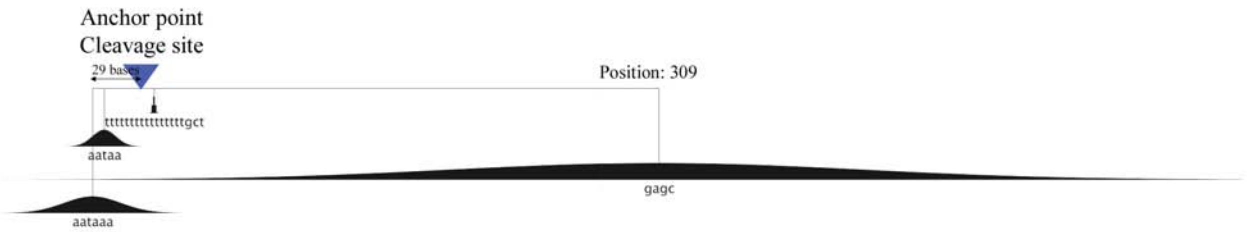


Figure 23. Transcription termination model trained from chromosome 22 sequences with modified parameters

Eponine package is available in different flavours apart from the anchored sequence model widely used in this project. I tried three different versions for learning transcription termination signals and those are –

(i) Windowed Sequence Model – In this method, an ‘anchor point’, positioned in EAS training on the cleavage site, is not used. Position constraints that are learnt from the set of training sequences using the same principles of the EAS are only *between* basis functions. I used an equal number of sequences (422 sequences) from confirmed 3’- gene ends of chromosome 22 and transcription unit as positive and negative set for training respectively. The trainer was allowed to run for approximately 4000 cycles and a model for every 500 cycles was dumped and tested for its performance. Interestingly, the motifs represented by the basis functions are different to those of the EAS model and the model performed poorly. This indicates that the windowed sequence model might not be suitable for the problem under consideration.

(ii) Hierarchical Sequence Model – This method works similarly to EAS except that position constraints are allowed to form anchor points for other position constraints in a hierarchical manner. This extension to the basic classifier may be helpful to better model situations where there are minor positional differences between features for given sets of sequences. For instance, TATA box might be used as a major constraint to classify promoters from other sequences. However, there are variations within different types of promoter sequences and few constraints can in turn be used to capture different promoter elements, say different transcription factor binding sites. Each transcription factor binding site has variations between different cell lines and few constraints can distinguish them (Figure 24). Thus the overall model with major constraints classify the dataset; and each major constraint is made from minor position constraints; and each minor constraint is made of other constraints capturing the few differences within the different set of sequences, making a hierarchical structure. This method was used to train models from the 422 positive and negative sequences as before, allowing the trainer to run for 9000 cycles. The model captured similar signals as the anchored model and a hierarchy structure was learnt near the cleavage site. However no hierarchical structure was found in the sequences downstream of the cleavage site.

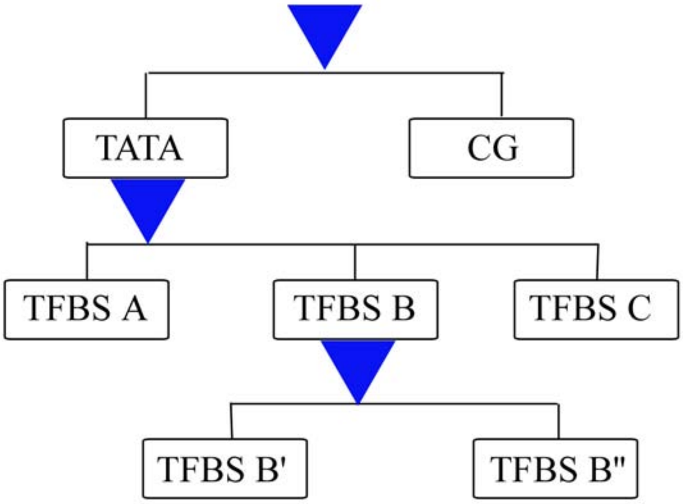


Figure 24. Schematic representation of hierarchical sequence model. Most promoter sequences have TATA box and CpG islands. Each promoter element can have specific DNA binding site for transcription factors to dock (represented as TFBS). Variations in each TFBS can in turn be modeled giving a hierarchical view of classifying different promoter types.

(iii) Dinucleotide Sequence Model – This classifier model capture features using the same strategy as that used in EAS except that the basis functions can also be derived based on dinucleotide compositions. The model found similar signals as those identified earlier by the anchored model.

(iv) Stem-Loop Model – As noted earlier, there is a possibility of there being secondary structure in the downstream region where RNA polymerase is likely to terminate and the algorithms have difficulty in capturing them as they are likely to vary its position for different genes. To address this I exploited the Eponine positional variance constraint by implementing the Nussinov algorithm with the Eponine trainer to detect secondary structures. However, training on the chromosome 22 datasets with this combination turns out to be very computationally expensive. However, the combination worked for histone genes where a stem-loop structure at the end of the gene is known to terminate transcription. I collected 100 histone stem-loop structures from Rfam database (Griffiths-Jones *et al.*, 2003) and trained it against random sequences. The Eponine trainer with the use of Nussinov algorithm basis function successfully picked a constraint that distinguished histone 3'-ends from other sequences.

Likewise, in the earlier analysis, I tested the model with a toy dataset of 100 sequences of 180 bases each. Each sequence is designed by using a known sequence of 60 bases with potential to form stem-loop structure, flanked with random sequences extracted from chromosome 20. This set of sequence is trained against a random sequence dataset to learn stem-loop constraints that can classify both the data. The trainer picked 3 constraints – 1 positive flanked by 2 negative constraints on training. The positive constraint is due to the higher base pair metrics score for the known sequence (than the random sequence) in the positive dataset. However the base-pair metrics score flanking the known sequence is lower than the random sequence and hence these differences between positive and negative sequences were captured as two negative constraints by the trainer.

Although the stem-loop model was successful in picking secondary structure constraints that were able to distinguish positive and negative sequences in the two cases described, the method was found to be computationally expensive for training on transcription termination datasets. Scanning 2000 bases downstream of cleavage site to identify a secondary structure constraint that can classify it from random sequences was found to be computationally expensive when implemented using Nussinov algorithm. Future work to improve the algorithm or adapt a different algorithm for scanning might prove to be useful in constructing higher order models.

3.6 Performance of the model

To test the performance of the model, it is necessary to define true positives and false positives. Any predictions with higher score value among the predictions made on the transcription unit and lying within 2500 bases from the cleavage site in the same strand as the gene are considered to be *True Positives (TP)*. Any predictions within the transcription unit in the same strand as the gene but not within 2500 bases from the cleavage site are considered as *False Positives (FP)*. Internal predictions on the reverse strand, intergenic predictions and predictions within 2500 bases of the cleavage site but in the reverse strand are ignored (Figure 25).

With these definitions, I tested the EAS model on chromosome 20 annotations and Figure 26 shows the performance as a Receiver Operating Characteristics (ROC) curve (ROC-Curve). ROC curve was plotted using the values given in Table 4.

Table 4. Coverage and accuracy values of transcription termination model along chromosome 20. ROC curve was constructed using these values.

Threshold	Coverage (%)	Covearge values	Accuracy (%)	Number of True positives	Number of False positives
0.99	90.38	188/208	11.58	140	1069
0.992	87.01	181/208	14.43	127	753
0.994	79.80	166/208	15.34	95	524
0.996	71.15	148/208	16.50	67	339
0.998	36.53	76/208	16.91	23	113
0.999	12.5	26/208	18.18	6	27
0.9992	6.73	14/208	17.64	3	14
0.9994	4.32	9/208	10.0	1	9
0.9996	0.96	2/208	0.0	0	2

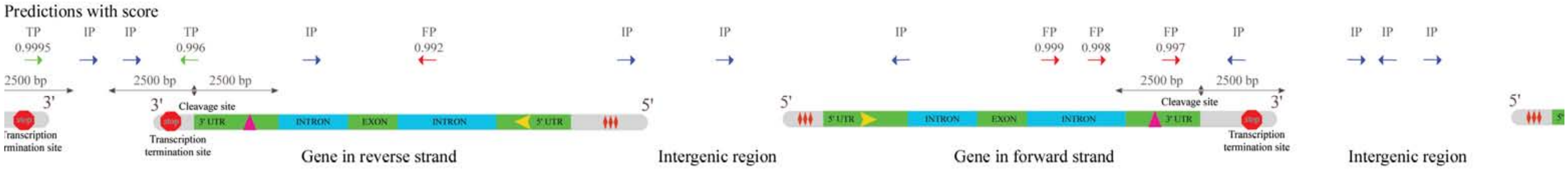


Figure 25. Schematic diagram showing criteria used for determining True positives (TP), False Positives (FP) and Ignored Predictions (IP).

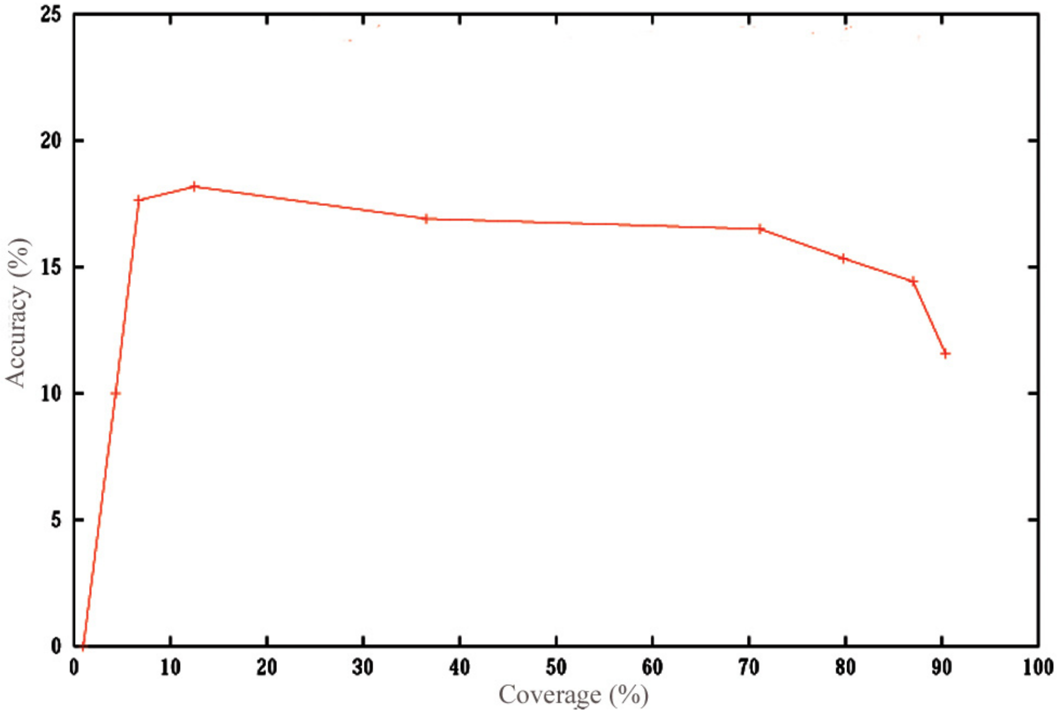


Figure 26. ROC curve on transcription termination sites in chromosome 20 for Eponine model.

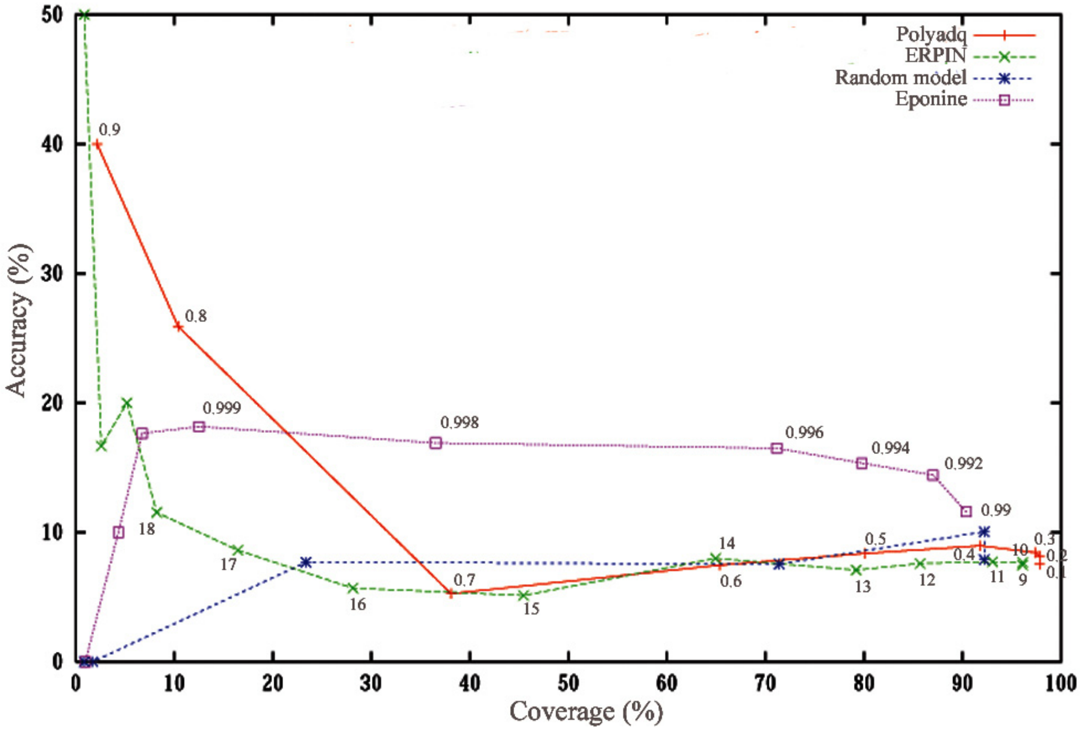


Figure 27. ROC curves on transcription termination sites for Polyadq, ERPIN and Eponine in comparison with random model.

I compared the performance with two other existing programs (ERPIN and Polyadq) to predict 3’-processing signals and a random model (Figure 27). Before discussing the performance I will briefly explain how random predictions are made.

Chromosome 20 without contig gaps is roughly 59 Mbp in length and hence 59,000,000 pseudo-random numbers between 0 and 1 were generated using PERL *rand()* function. The number generated is the score for each nucleotide in the chromosome and if the value exceeded 0.99 (threshold used in EAS model), it was recorded as a random prediction and the corresponding position in the chromosome was dumped. The strand for the predictions are generated using another *rand()* function. This procedure resulted in 294485 forward strand and 295229 reverse strand random predictions being collected. All the predictions are dumped in GFF format and compared with annotations of chromosome 20.

Figure 27 shows the ROC curve for Eponine, ERPIN, Polyadq and the random model. Both Polyadq and to lesser extent ERPIN have the highest accuracy at low coverage, however their performance drops to random at about 20% and 40% respectively. The accuracy of Eponine predictions seems less correlated with score, with no high accuracy peak at low coverage, however Eponine predictions remain at twice the accuracy of random at about 70% coverage. For comparable level of coverage, Eponine makes approximately 41% and 38% less false positives compared to Polyadq and ERPIN respectively, since at this level of coverage it is the only algorithm still performing better than random.

To determine if the genes identified by ERPIN, Polyadq and Eponine are same or different; I took predictions of the models at cut-off values of 9 (default threshold), 0.1 (default threshold) and 0.99 (comparable Eponine threshold) respectively. The result showed most of the gene termination sites were identified by all the three programs (might due to high coverage) and Eponine has detected all the predictions of EPRIN and misses only 14 genes predicted by Polyadq.

Among the positional constraints learnt by the EAS model, the 3’-end processing signals (poly(A) signal and GT rich motif) play an important role as can be seen from the constraint weight in Table 3. This is consistent with the results from experiments where upon deletion of the poly(A) signal the elongating polymerase failed to terminate and caused a run-over

(Edwalds-Gilbert *et al.*, 1993; Yeung *et al.*, 1998). To investigate the poly(A) signal requirement, I made a few models without DNA sequences spanning the 3'-end processing signals. Comparison of models developed from DNA sequences spanning 100 to 2000, 300 to 2000, 500 to 2000, 1000 to 2000 and 300 to 3000 bases from the cleavage site showed all the models performed less well than the models with 3'-processing signals. Thus the poly(A) signal appears to be a significant constraint in the model and required to make valid predictions along the chromosome. However the PE found by the model should not be underestimated as they are found to improve prediction accuracy.

3.7 Positional accuracy of the model

The density of predictions along the chromosome with respect to the annotated cleavage sites is shown in the Figure 28. As it can be noted, most of the predictions are associated with the annotated sites. Apart from the huge peak, there are predictions on either side of the peak with a distribution equal to the background prediction density. The model is good in detecting the directionality of the transcription termination site and the figure shows only the predictions matching the same strand as that of the annotation. Likewise, positional accuracy of ERPIN and Polyadq are shown in Figure 29. These methods are also good in predicting the transcription termination sites accurately.

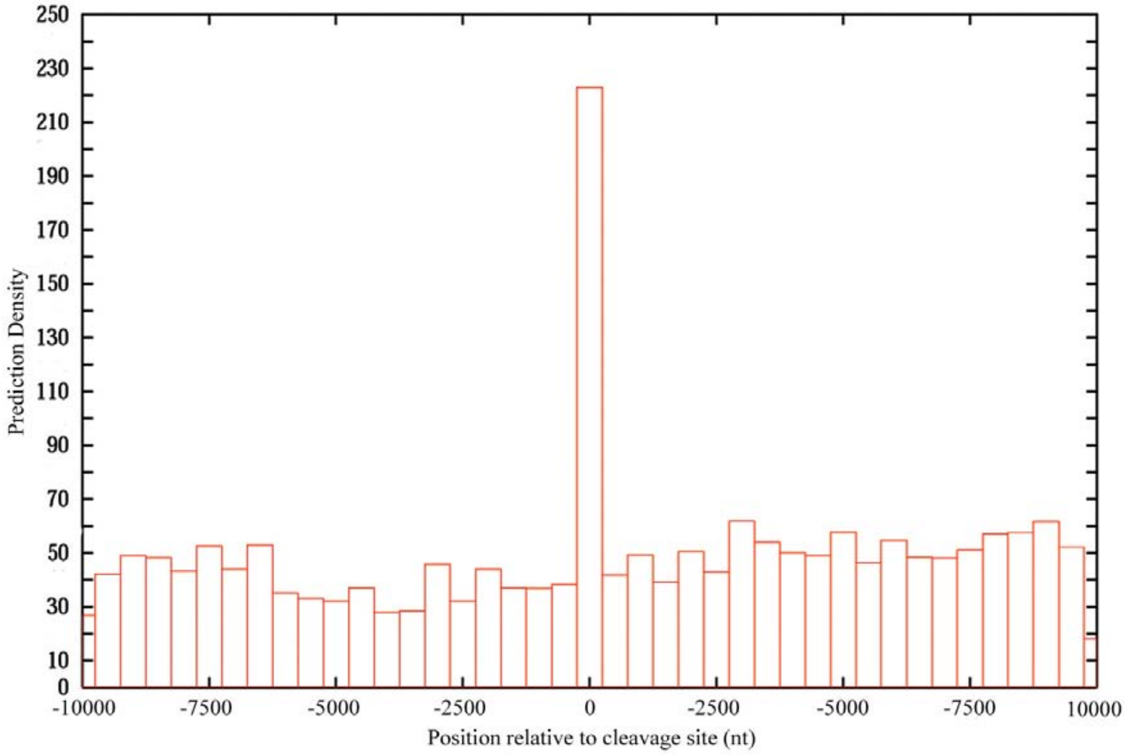
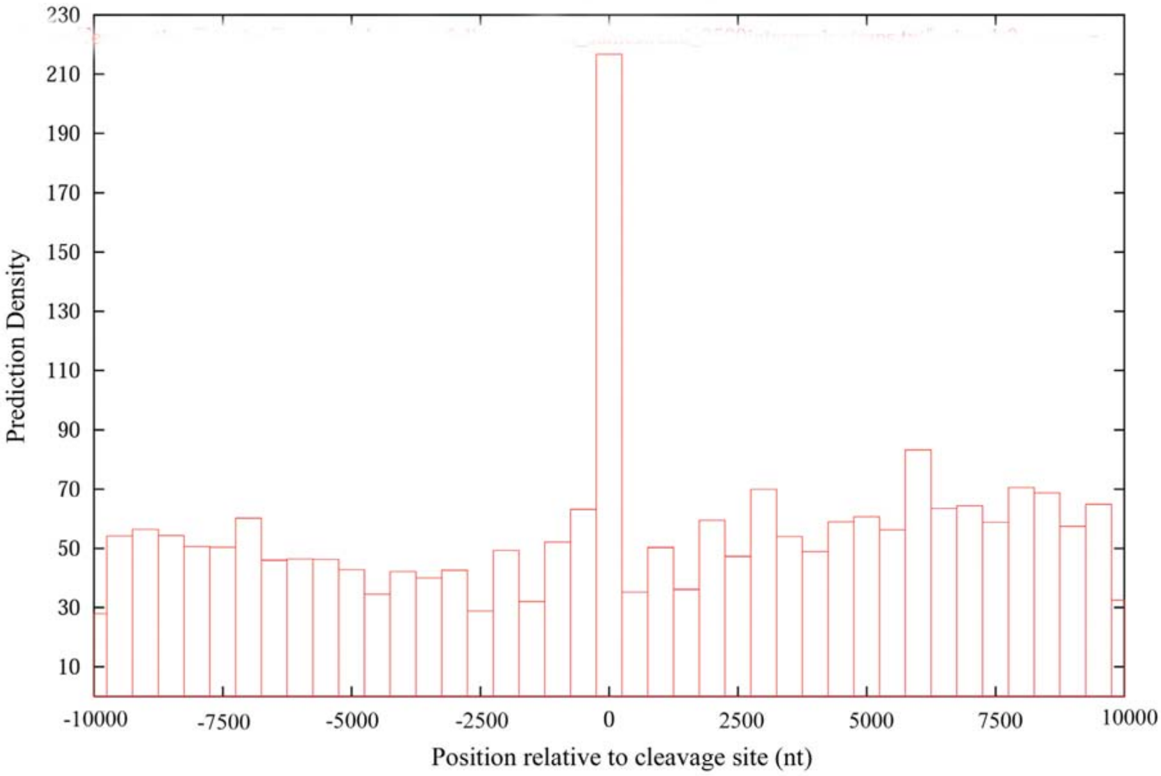


Figure 28. Prediction density for transcription termination model along chromosome 20.



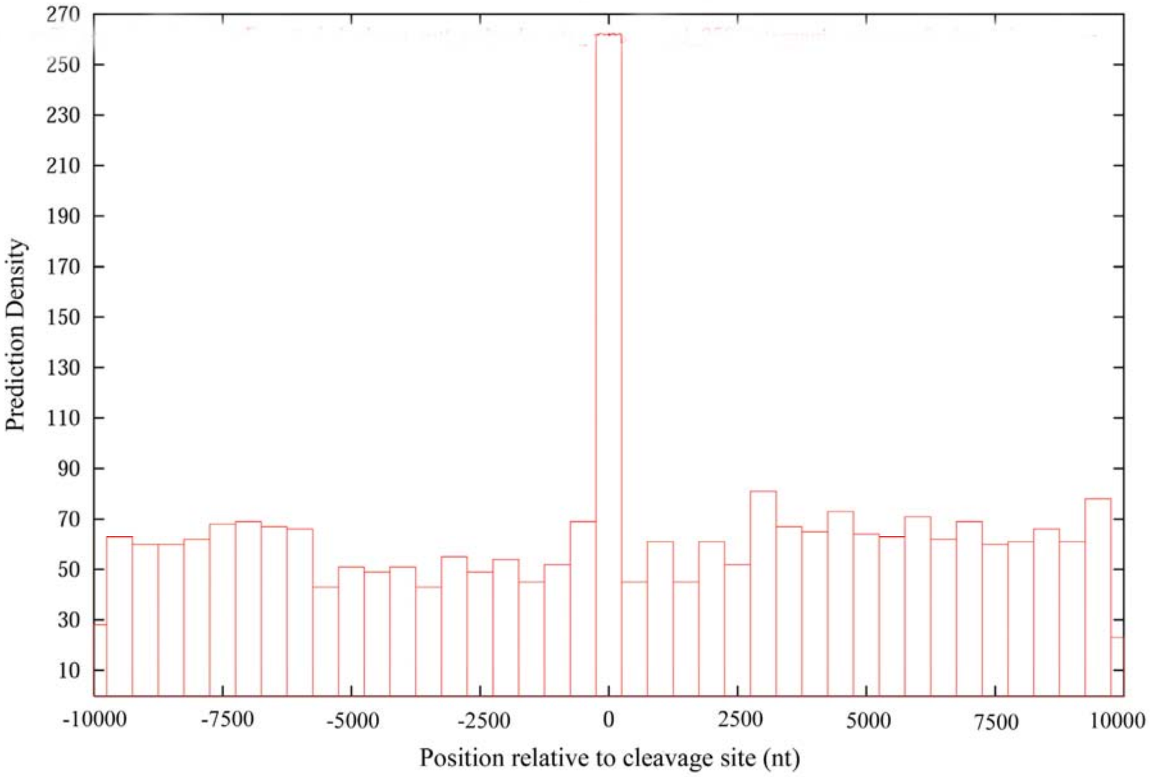
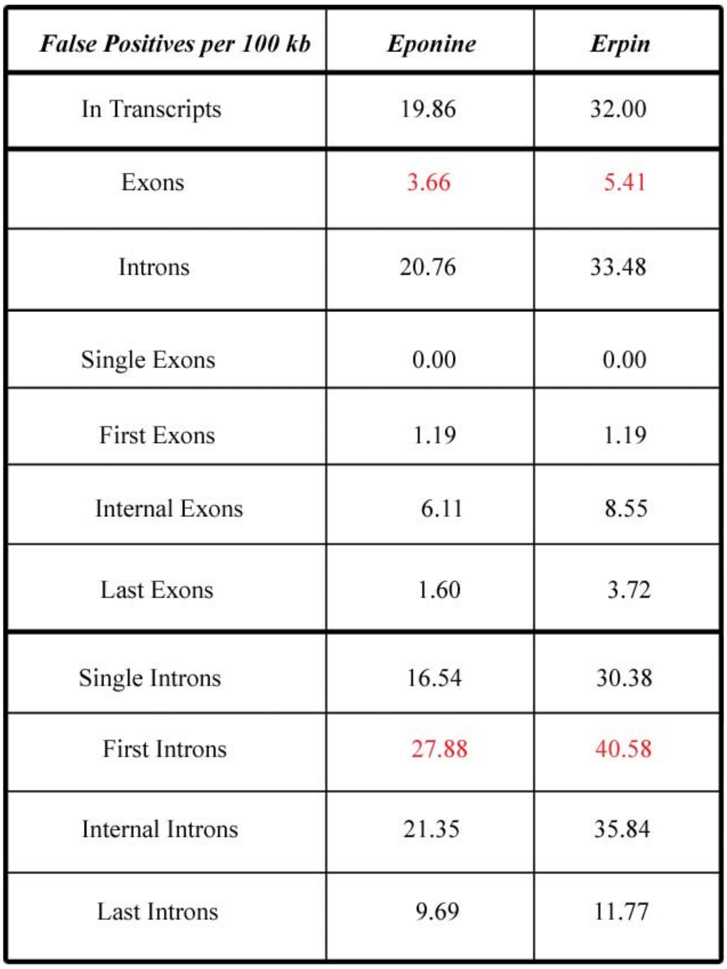


Figure 29. Prediction density along chromosome 20 for (a) ERPIN and (b) Polyadq.

3.8 Internal predictions

I hereafter concentrate on the distribution of predictions made within the gene as termination-like signals lying within the transcription unit are likely to challenge the transcription machinery leading to premature termination. Understanding the features of these predictions will help to differentiate them from correct gene ends and increase the accuracy of the model. Nearly 10% of total predictions along the chromosome are found within the gene in the same strand. To investigate whether the internal predictions made with Eponine model are linked with its learning protocol, an independent program, ERPIN was used for comparison. To facilitate this comparison, values are reported as predictions per 100 kb of genome sequence, subdivided into gene feature categories (Table 5).

Table 5. Distribution of false positives within transcripts, exons and introns.



False Positives per 100 kb	*Eponine*	*Erpin*
In Transcripts	19.86	32.00
Exons	3.66	5.41
Introns	20.76	33.48
Single Exons	0.00	0.00
First Exons	1.19	1.19
Internal Exons	6.11	8.55
Last Exons	1.60	3.72
Single Introns	16.54	30.38
First Introns	27.88	40.58
Internal Introns	21.35	35.84
Last Introns	9.69	11.77

The predictions per 100 kb of gene are 19.86 and 32.00 for Eponine (threshold: 0.99) and ERPIN (default parameters) respectively indicating the good performance of Eponine model assuming most of these predictions represent true false positives. The predictions were then subdivided into those present in exons and introns of the gene. The predictions per 100 kb of intron were found significantly higher than in the exons. The reason for this huge bias towards introns is unknown. On further classifying the distribution of predictions in exons between single (gene with just one exon), first, internal and last exons, showed internal exons have more predictions. However, since the number of predictions per 100 kb of exons is low, deriving conclusions from these small figures holds no importance.

In the same way, distribution of predictions found in introns, were classified between single (genes with just one intron), first, internal and last introns. The numbers indicate that first introns have significantly more predictions compared to internal and last introns. This holds true for ERPIN program predictions as well. The bias towards first introns is puzzling and so far there is no experimental evidence to explain the phenomenon. The table shows last introns have dramatically lower prediction rate as I excluded any predictions within 2500 bases from the cleavage site for this analysis. The average ERPIN predictions per 100 kb of single and first intron was calculated to be 39.26 and this value is equal to the 39 false positives per 100 kb specificity reported by the authors of the program earlier (Gautheret and Lambert, 2001).

Thus almost all internal predictions found within gene are present in introns and first introns have more predictions than internal and single introns.

Interesting experimental results were found in a recent chromatin immuno-precipitation assay by Affymetrix using high-density oligonucleotide arrays representing all nonreptitive sequences on human chromosomes 21 and 22 for Transcription Factor Binding Sites (TFBS) (Cawley *et al.,* 2004). The assay was designed to identify TFBS for Sp1, cMyc and p53 factors and found a minimal of 12,000 sites for Sp1, 25000 sites for cMyc and 1600 sites for p53 in the human genome. Only 22% of these predictions are found near 5' termini of the gene while 36% lie within or near 3' to well characterized genes and remaining 24% in intergenic regions. The TFBS not linked to 5' termini of the gene are found correlated with noncoding RNAs. A significant proportion of these RNAs are co-regulated with protein coding genes and activated by retinoic acid. The unexpected number of TFBS with just 3 transcription factors observed under one environmental induction condition suggests that there may be a large number of transcription units still to be identified in the genome. As the novel transcripts identified are found to be regulated by the same mechanism as protein-coding genes, they are expected to have similar transcription termination sites indicating that some of the excess predictions made by EAS model along the chromosome might be biologically significant.

The Sp1 transcription factor is known to bind G-rich elements (resembling PE) and is able to pause the elongating polymerase (Yonaha and Proudfoot, 1999). Identification of Sp1 TFBS within genes supports the idea that some of the internal predictions may be functional pause sites (Cawley *et al.,* 2004). Likewise, the PE in the EAS model resemble the MAZ (Myc-Associated Zinc Finger Protein) binding site (GGGGAGGGGAC) and MAZ sites have been shown to pause polymerase better than Sp1 sites. Also, MAZ protein has been found to be necessary but not sufficient for efficient 3’ end formation (Yonaha and Proudfoot, 1999).

All of the experiments described above support the idea that some of the internal predictions made by the EAS model may not be false positive predictions but may be functional *in vivo*.

The EAS model has poly(A) signal as its major constraint and an internal prediction means similar signals are present within the transcription unit. A SELEX experiment to determine the branch point sequence from HeLa cell nuclear extract yielded a sequence motif AAUAAAG, that proved to be functional both as polyadenylation and branch site in a competitive manner (Lund *et al.,* 2000). Earlier experiments have also shown the competition between spliceosome and polyadenylation factors while the RNA polymerase is elongating the gene (Takagaki and Manley, 1998; Takagaki *et al.,* 1996). The complexes compete for the branch point signal in the acceptor site and depending on the local concentration of factors and strength of the signal either splicing or polyadenylation occurs. Thus the conserved branch point signal and its associated poly(T) tract might mimic the poly(A) signal and the poly(T) tract of the 3’-processing signals. I suspected that this could have caused the EAS model to make internal predictions. However I found no significant increase in the density of predictions near branch point signals.

I also suspect that in at least in a few cases, the internal predictions made by the model are not really false positives and may instead act as terminator signals for alternative transcripts of the same gene. As the number of alternative transcripts is difficult to quantify, even for well annotated chromosomes, further analysis will be needed to clarify this.

3.9 GO correlation

Initial observations of lists of genes having internal predictions showed they are enriched in a subset of genes. To investigate the nature of genes having high number of internal predictions, I used the Gene Ontology (GO) database (Gene Ontology Consortium, 2004). From the annotations of chromosome 20, 176 genes were mapped to a GO identifier and 45 of these genes have 10 or more internal predictions so I used GO to find their biological role. The analyses showed, 32 out of 45 genes have *Cell growth and or maintenance* function corresponding to GO identifier, *GO: 0008151*. Thirty two (32/45) is higher than the random expectation of 24 from 91 genes with the same GO id in the 176 genes dataset. There were much smaller differences in the use of more specific GO terms, and hence no functional annotation of genes with large number of internal predictions could be determined.

Figure 30 shows the wide variation in the density of internal predictions with an average of 10-18 internal predictions per 100 kb of transcription unit. This number is less than the prediction made by a comparable program, ERPIN. I did another GO ontology search for a set of 27 genes that have 20 or more internal predictions per 100 kb but found no common functional annotation. Unsurprisingly, the number of internal predictions was found to be correlated with transcript and intron length. Figure 31 shows transcript length and internal prediction rate has a linear correlation. However Figure 32 shows shorter introns of less than 1000 bases have high propensity to have more internal predictions. In introns of less than 1 kb, an average 18 internal predictions per 10 kb is present. This is significantly higher than 2.5 internal predictions per 10 kb of large introns. The reason for such bias is not known although manual investigation of some predictions suggested that they might be terminators of alternative transcripts.

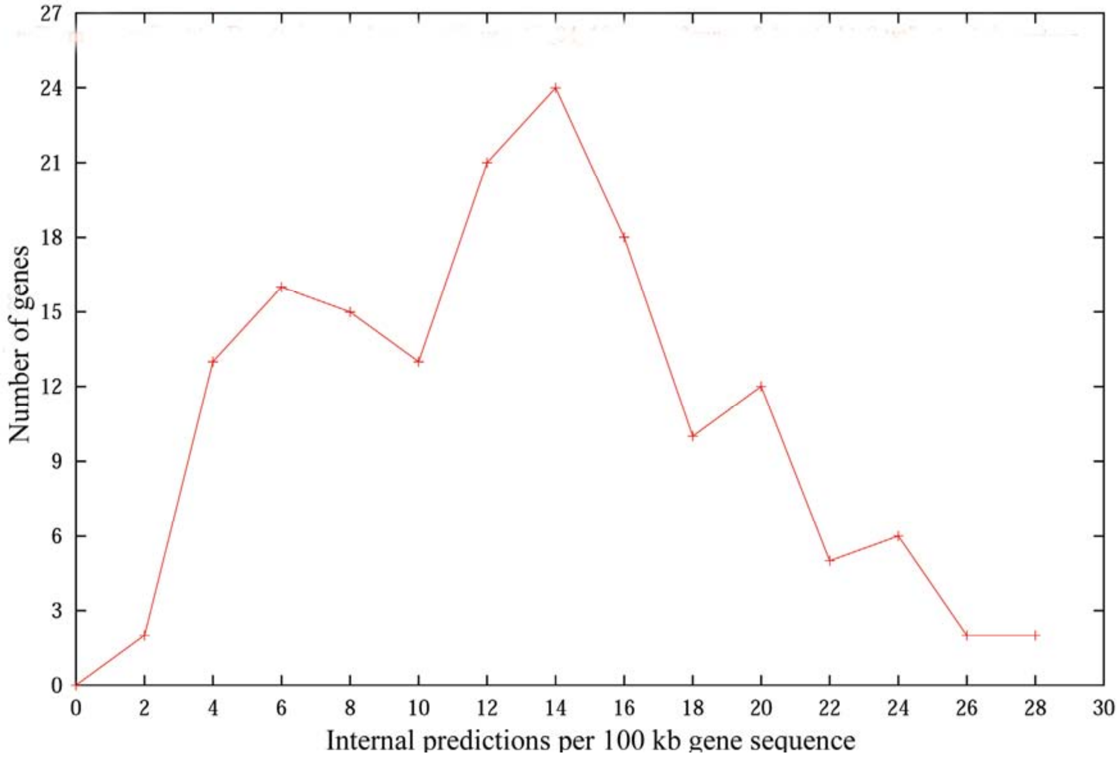


Figure 30. Internal predictions per 100 kb of gene sequence in chromosome 20 for Eponine model.

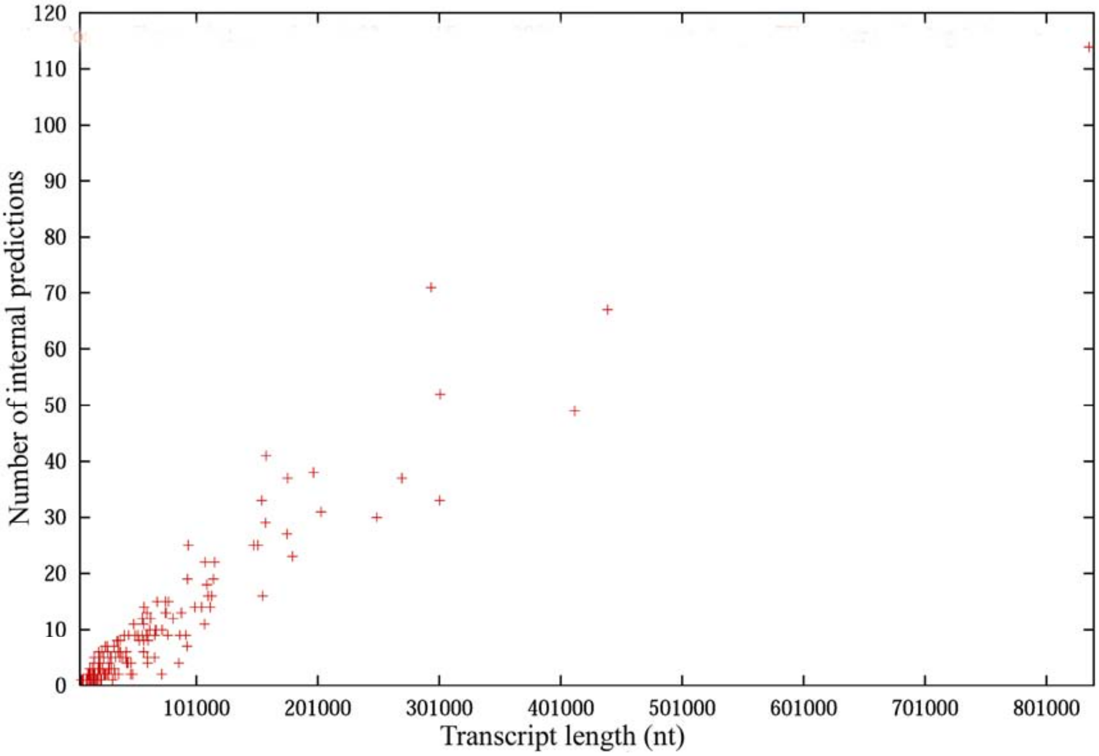
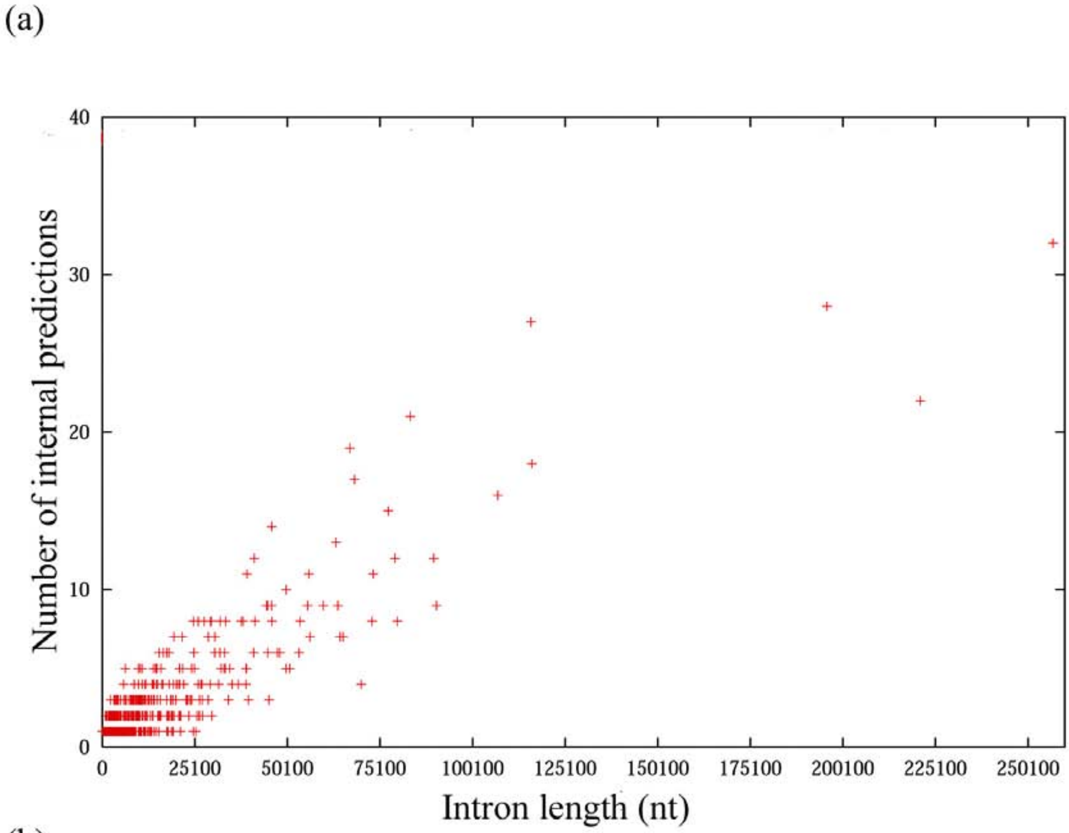


Figure 31. Internal predictions per transcript in chromosome 20 for Eponine model.



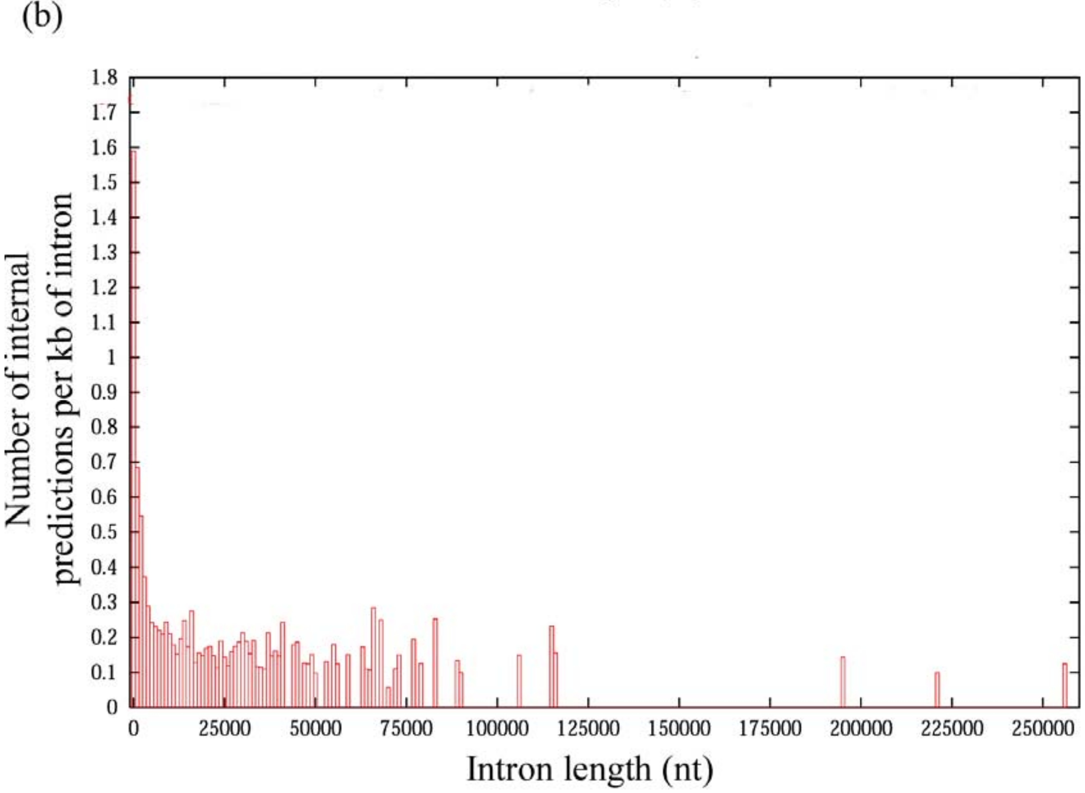


Figure 32. Internal predictions of Eponine model in introns of chromosome 20 (a) Number of internal predictions versus intron length (b) Number of internal predictions normalised over intron length.

3.10 Predictions near annotated gene start sites

Analysis of the density of predictions with respect to different gene features shows an unexpected concentration near the start of genes. The predictions found near the annotated gene start site are found to be close to its promoter elements and are unlikely to be a transcription termination site of the preceding gene. This was confirmed by considering only genes with no known annotated genes in the 2500 bases upstream of the start site and a minimum transcription unit length of 2500 bases. With this criterion, I selected 399 genes from the 584 annotated genes in chromosome 20, and 639 genes from the 1003 annotated genes of chromosome 6. The position density graphs shown here are calculated for these subsets of annotated genes. Figure 33 shows the density of predictions near annotated gene start sites of chromosome 20 in the same and opposite strand of the gene. A significant number of predictions are found just upstream (within 500 bases) of gene start site in the same strand orientation as the gene and the density is generally higher upstream than within the transcription unit (Figure 33a). Density predictions in the opposite strand show a significantly reduced number of predictions at the start site (position 0) and no significant increase in predictions elsewhere (Figure 33b). These results were generally consistent across different trained Eponine models, however predictions of two other transcription termination models also showed a number of predictions in the opposite strand just downstream of the annotated start site (Figure 34). Similar results were obtained on a different independent dataset of chromosome 6 (Figure 35a and Figure 35b). Also to investigate whether the predictions near gene start site are specific to models derived using the Eponine package, or more generally predicted, results form the two independent programs, Polyadq and ERPIN were similarly analysed for chromosome 20 sequences (Figure 35c; Figure 35d; Figure 35e; Figure 35f). Although there is no significant same strand prediction peak in the upstream region as seen for Eponine models, the density of predictions do appear to be relatively higher in the upstream region than in the transcription unit.

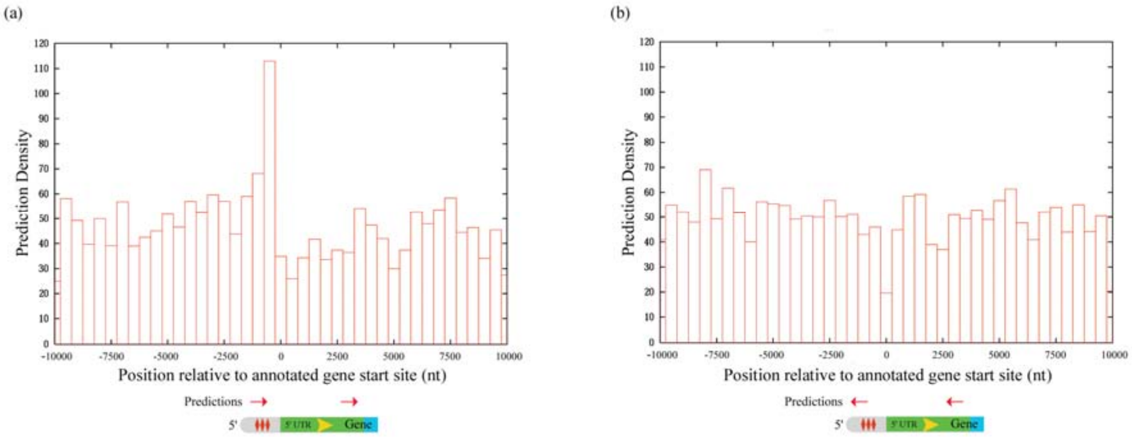


Figure 33. Prediction densities for transcription termination model near chromosome 20 annotated gene start sites. (a) Density of predictions in the same strand as of the gene (b) Density of predictions in the reverse strand as of the gene.

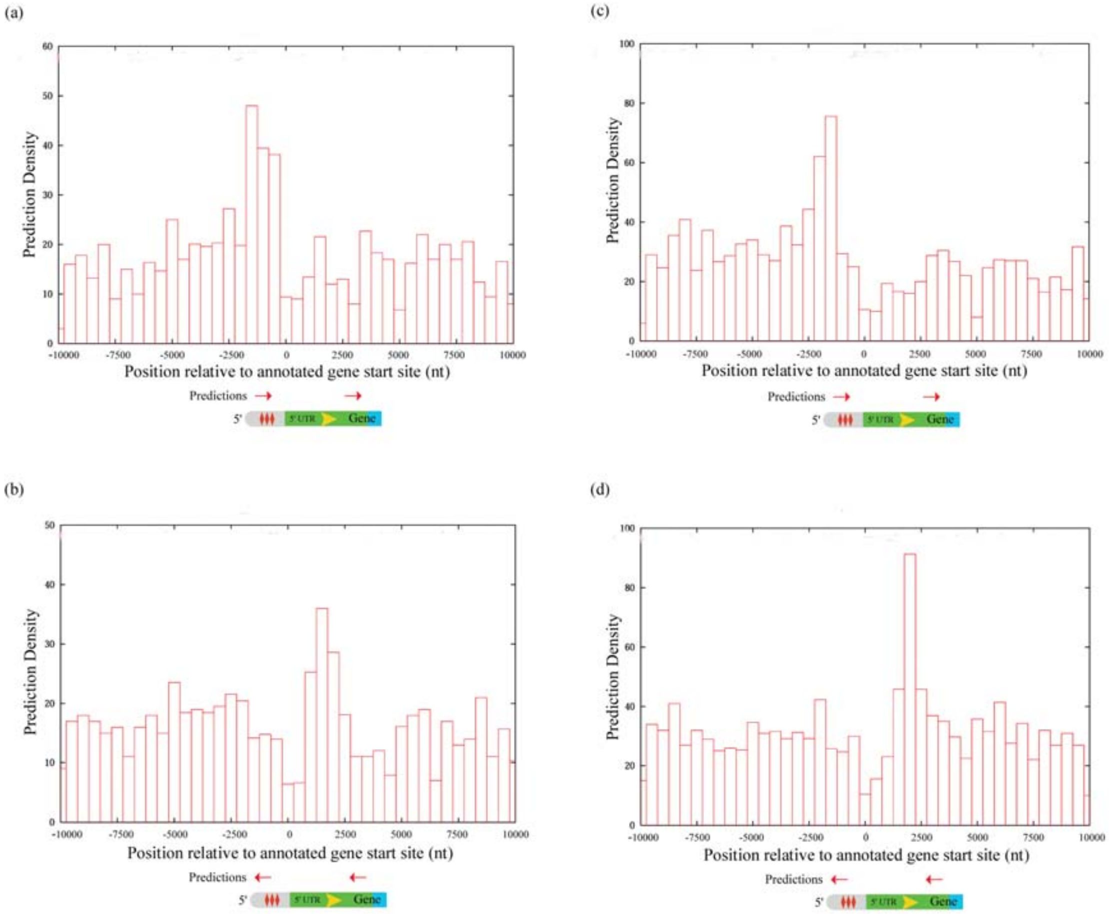


Figure 34. Prediction densities for two other transcription termination models near chromosome 20 annotated gene start sites (a), (c) Densities of predictions in the same strand as of the gene. (b), (d) Densities of predictions in the reverse strand as of the gene.

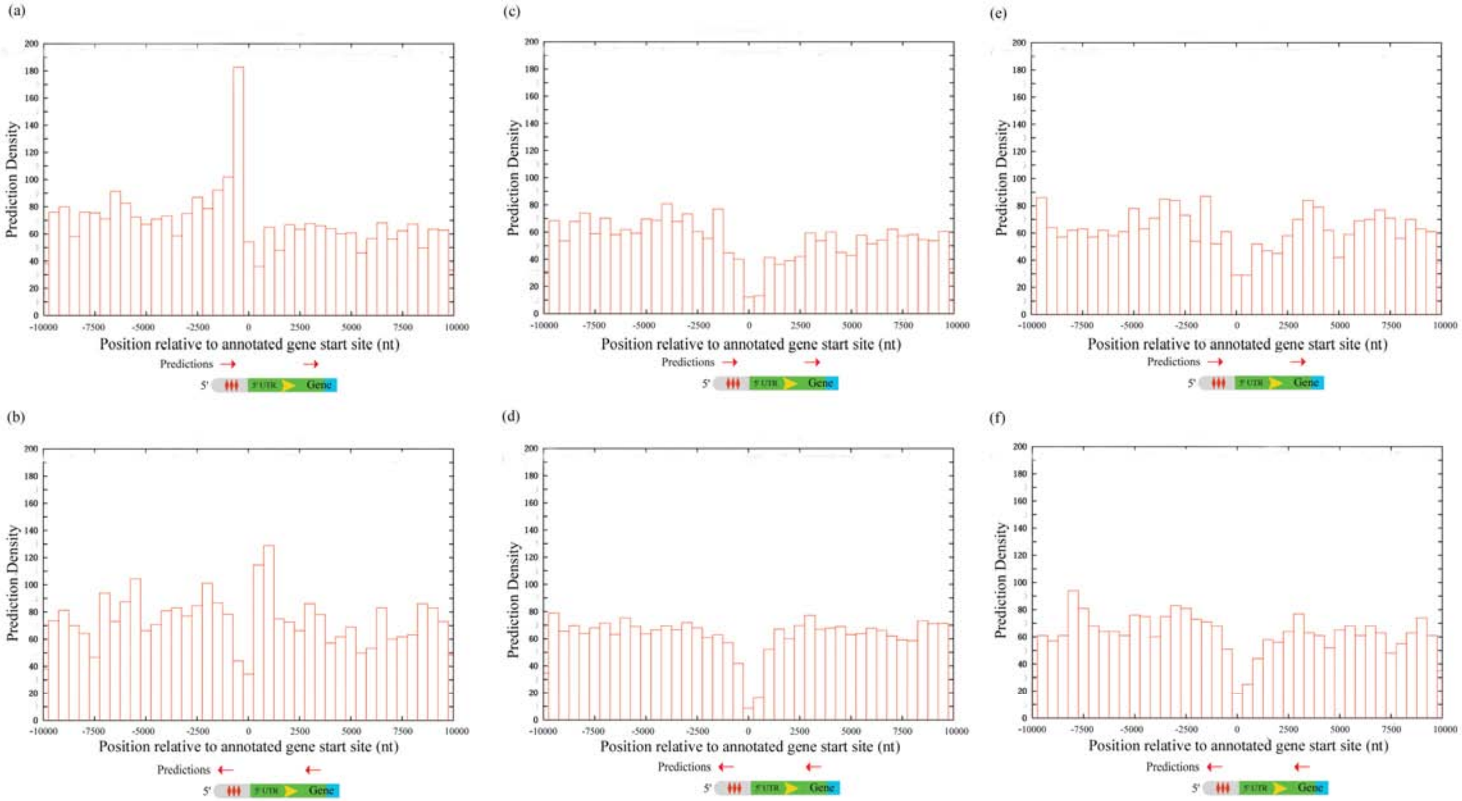


Figure 35. Prediction densities near chromosome 20 and 6 annotated gene start sites. (a) Density of predictions in the same strand as of the gene in chromosome 6 predicted by Eponine (b) Density of predictions in the reverse strand as of the gene in chromosome 6 predicted by Eponine (c) Density of predictions in the same strand as of the gene in chromosome 20 predicted by ERPIN (d) Density of predictions in the reverse strand as of the gene in chromosome 20 predicted by ERPIN (e) Density of predictions in the same strand as of the gene in chromosome 20 predicted by Polyadq (f) Density of predictions in the reverse strand as of the gene in chromosome 20 predicted by Polyadq.

Thus the plots show a higher than expected number of predictions just upstream of annotated gene start sites and on the assumption that this effect is real it is interesting to speculate on possible biological functions.

3.11 Hypotheses

Here, I have attempted to explain this unusual set of predictions by proposing a hypothesis assigning biological function based on previous knowledge from experiments.

Hypothesis I: As promoter detection by the RNA polymerase complex depends on a scanning mechanism, a terminator-like sequence positioned just upstream of a gene start site (referred hereafter as FP-TSS) might help the complex to prevent long regions of scanning and recruit the factors to the promoter elements and localize them.

The FP-TSS in the promoter region just 500 bases upstream of the annotated gene start site in the same strand as that of the gene might act as a guiding signal for the transcription factors to bind to its corresponding DNA binding domains. This will reduce the range of nucleotides need for scanning and help the factors to identify its binding site. If any case, if the factors gets recruited far upstream of the start site, then they are likely to get dissociated and the scanning terminated due to the FP-TSS signals. Thus the FP-TSS might help in positioning the initiation complex close to promoters.

Experimental evidences published recently support this view. Joseph Martens (Martens, 2003) reported a novel transcription interference assisted gene regulation process in the ‘Mechanisms of Eukaryotic Transcription’ conference held at CSHL, New York. In this case, a short transcript initiated from an upstream TATA box overlaps a downstream TATA site that is responsible for transcription of the downstream gene, SER3 in *Saccharomyces cerevisiae*. The upstream TATA site and transcript lies within the promoter elements of the SER3 gene and transcription of this short transcript is dependent on Snf2 chromatin-remodeling complex. The short transcript interferes with SER3 transcription by masking the overlapping activator elements of SER3. The interference was confirmed with the derepression of SER3 transcription when the upstream TATA site was mutated. Interestingly, apart from having a fully functional TATA box, the upstream short transcript has a poly(A) signal just like a normal gene. The positional occurrence of this poly(A) signal correlates with the prediction identified earlier by the Eponine model near gene start sites. Thus this experiment confirms the poly(A)/terminator-like signals found upstream of the gene start site are likely to have biological functions.

Evidence for the hypothesis also stems from more unexpected discoveries of interaction between RNA polymerase II and processing machineries. As described in chapter 1, the interaction between polyadenylation machinery factors (CPSF, CstF) and CTD of RNA polymerase II has now been well established (Dichtl *et al.,* 2002b; Osheim *et al.,* 2002). Interestingly this interaction begins right from the promoter and apart from CTD, even general transcription factors like TFIID associate with CPSF (Dantonel *et al.*, 1997) making it a component of the transcription initiation complex. This study adds support to the presence of termination like sequences in the upstream region of gene start sites as predicted by the EAS model. Although the role of CPSF in the promoter region is not known, the presence of it in the initiation complex emphasizes that the FP-TSS are likely to have biological functions.

In a recent review, Calvo and Manley have discussed the interaction between 3' processing factors and initiation complex do not stop with CPSF but includes other factors like symplekin (Pta1 in yeast), Ssu72 and PC4 (Calvo and Manley, 2003). There is no definite answer for such growing presence of polyadenylation factors near to the promoter region although the conditions are explained by linking the factors to different roles when present in promoter and termination regions. However such strategies have created only confusing explanations. For example, Ssu72, a phosphatase that interacts with TFIIB and present in elongating polymerase (Sun and Hampsey, 1996) is expected to have anti-terminator activity. However in other experiments it was found Ssu72 is necessary for 3'-end formation and/or termination (Dichtl *et al.,* 2002a; Gavin *et al.,* 2002; He *et al.,* 2003). Similarly PC4, a co-activator protein that binds single and double stranded DNA displays anti-terminator activity (Aranda and Proudfoot, 2001; Ge and Roeder, 1994) but interacts with CPSF and CstF at the 3' processing signals (Calvo and Manley, 2001). Both PC4 and Ssu72, which are unrelated in primary structure, share a common function of helping transcriptional machinery to identify the gene start site by interacting with the general transcription factor, TFIIB (Sun and Hampsey, 1996; Woychik and Hampsey, 2002). Also the factors interact with symplekin, a component of CPSF, mutually exclusive with PC4 binding at the 5' end of the gene and Ssu72 at the 3'-end (He *et al.,* 2003). However recent studies show Ssu72 possesses protein phosphatase activity and may be required for both ends for the gene (Ganem *et al.,* 2003; Meinhart *et al.,* 2003). Like other factors, symplekin also facilitates interaction of transcription initiation factors with CTD of polymerase at the time of transcription initiation (Rodriguez *et al.,* 2000). Thus the experiments show the functions of the protein molecules are conflicting and have different roles within the transcription machinery.

However the above studies confirm the unexpected discoveries of huge number of polyadenylation factors near promoter elements and that the poly(A) signal near to the promoter might be functional. This forms the basis for my hypothetical model that recruitment of polyadenylation factors at the promoter regions, might in turn, recruit other general transcription factors and thus help in localizing the initiation complex just upstream of gene start sites. Also, the predicted signals may help in avoiding the unnecessary scanning required to find the start site and take part in gene regulation as described earlier (Martens, 2003). A related puzzling question which remains to be tested is, if FP-TSS is functional, are they similar in activity to the 3'-processing signals? This can be verified by cloning the FP-TSS and surrounding sequences to the end of the gene and expect for termination of RNA polymerase II to occur. If polymerase terminates it will unambiguously confirm the upstream predictions are similar to 3'-processing signals found at the end of genes.

The hypothetical model also fits the transcription machinery model (Cook, 1999) wherein transcription factors and RNA polymerase associate together to form a machinery through which the DNA passes when a particular gene needs to be transcribed. Given the model is true, it is not a surprise to find so many polyadenylation factors in the promoter elements as all the factors are likely to participate in the machinery. Presence of FP-TSS at the promoter region in this machinery model will define the start of a gene by terminating any transcription starts initiated far upstream from promoter.

Hypothesis II: The unusual condition of significantly high number of predictions in the first intron compared to other introns is explained in this hypothesis. The polymerase queuing model (Heinemann and Wagner, 1997; Wagner, 2000) explained in prokaryotes forms the basis for this. Prokaryotic genes are transcribed by multiple polymerases at any given time, and this leads to a trail of polymerase transcribing a single gene. If the rate of transcription is slow when compared to entry of new polymerase complexes at the promoter region, the transcription complexes are likely to queue all along the gene. I contemplate the same mechanism might act in eukaryotic gene transcription as well. In cases of genes with high expression levels and strong promoter motifs, there is a high probability of more transcription initiation complexes getting assembled and initiating transcription. However if the rate of transcription is likely to be less than the rate of assemblage and initiation then the complexes are likely to be queued. Having a terminator like sequences at the first intron might act as a strong pause signal and induce weak or incompetent complexes to terminate initiation and dissociate from the DNA. However if the complex is competent enough then its likely to continue the transcription process past the pause signals in the first intron and complete the whole gene transcription. Presence of predictions at the first intron compared to other internal introns might save energy from unnecessary transcription as the cell can abandon it just after initiation.

This hypothesis depends on the assumption that transcription of a gene is carried out by multiple polymerases at a given time. However there are split views on this. Supporters of the transcription machinery model argue gene transcription is done by a single polymerase at a given time. However there is no consensus so far. Even if the transcription machinery model holds true, the terminator like signals at the first intron might act as check point to evaluate the processivity of the machinery complex in transcribing the gene.

Experimental evidences support this hypothesis and reports premature termination occurs in the 5’ region of many viral and cellular genes (reviewed in Spencer and Groudine, 1990). Such intragenic termination occurs efficiently near gene start sites. This was shown in c-myc gene where terminator like sequence present 310 bases from the start site when moved to 600 bases resulted in more than five fold decrease in termination efficiency. This emphasizes that the evaluation of processivity of complexes occur early in the transcription process and influenced by the distance from start site (Roberts and Bentley, 1992). Similar observation in c-myc, c-myb, c-fos, β-globin, adenosine deaminase and porphobilinogen deaminase genes show that all intragenic terminations occur in the 5’ region of the genes usually within 1 kb from the start site (Beaumont *et al.,* 1989; Bentley and Groudine, 1986; Chinsky *et al.,* 1989; Lois *et al.,* 1990; Mechti *et al.,* 1991; Watson, 1988). These experimental results show the predictions in the first intron might have biological function and help in classifying the transcription complexes that initiate from promoters into two heterogeneous sets based on their processivity. Thus the predictions might act as an attenuator and thereby allow only read-through complexes to complete transcription.

The read-through of RNA polymerase II can be assisted by various complexes. One such complex extensively studied in phage, called N and Q anti-termination system involves at least six proteins (Das, 1993; Friedman and Court, 1995; Greenblatt *et al.,* 1993). A similar mechanism is present in eukaryotes as well and so far 5 factors were reported. The first factor, S-II was originally discovered in mouse and found to suppress pausing of polymerase and activate reinitiation (Reines, 1994). The second factor, TFIIF (factor 5) is required for initiation and stimulation of elongation rate of polymerase (Wiest *et al.,* 1992). The third factor, TFIIX, identified in HeLa cell extract stimulates elongation of polymerase II (Bengal *et al.,* 1991). The fourth factor, a yeast protein YES stimulates the elongation rate of polymerase II (Chafin *et al.,* 1991). Finally, a factor, P-TEF stimulates elongation by forming part of productive elongation complexes and restores initiation of paused polymerase (Orphanides and Reinberg, 2002). DmS-II and Factor 5 forms part of the late elongation complex while P-TEF plays role in the early elongation complex (Marshall and Price, 1992). P-TEF phosphorylates DSIF and CTD of polymerase and thus increase the processivity of the elongation complexes (Renner *et al.,* 2001).

Thus the predictions near to the gene start site and first intron are likely to have biological functions and experimental evidence evaluating it will add new knowledge to the understanding of the transcription machinery.

3.12 Concluding remarks

With Eponine transcription termination models, I identified few multiplex pause elements present in the sequences downstream of the cleavage site. Occurrence of these signals repetitively indicates they might complement each other in pausing polymerase before release. The signals are similar to the sequences found in yeast ura4, α-globin, C2, factor B and nmt2 genes (Aranda and Proudfoot, 1999; Birse *et al.,* 1997; Yonaha and Proudfoot, 2000). The A-richness found in human β and α2 globin genes are represented as negative positional constraint (TTTT motif) in the model (Dye and Proudfoot, 2001; Enriquez-Harris *et al.,* 1991). Likewise the G-richness found in the pause elements (Table 1) agree with experimental results from human C2 and factor B genes (Ashfield *et al.,* 1991). MAZ and Sp1 transcription factors bind to these G-rich elements and interestingly in a recent experiment by Affymetrix (Cawley *et al.,* 2004), 34% Sp1 TFBS are found internal or proximal to the 3’ end of the gene. These TFBS show a possible correlation with internal predictions identified by Eponine model. Detailed analyses of internal predictions indicate they are not randomly distributed and significantly present in longer genes and shorter introns (less than 1000 bp).

Earlier computational analyses by Nussinov (Nussinov, 1987, 1990) indicated the presence of TATAAA, AGGG and GGGC motifs in the sequences upstream of the transcription initiation site. These motifs resemble the AATAAA and pause elements of the model and thus correlate with significant number of predictions found proximal to the gene start sites.

Thus identification of transcription termination signals in the first intron and proximal to gene start sites encourages future mechanistic investigations and discussions concerning the transcriptional machinery and the possible reconsideration of current concepts of gene regulation in the eukaryotic genome.

MODELLING DONOR AND ACCEPTOR SITES

4.1 Introduction

Recent *in vivo* experiments show splicing or RNA processing events are found to be temporally and spatially related to the transcription process and that the CTD of the RNA polymerase II itself initiates such interactions (refer chapter 1, Cramer *et al.,* 2001; Manley, 2002). Hence while studying transcription termination, it is natural to consider splice sites and investigate whether modelling them is required to create an *ab initio* gene prediction system based on regulatory signals. Combining transcription termination models with splice site models may remove some of the internal predictions by the termination model and thereby help in predicting the correct gene structure.

In existing *ab initio* gene prediction systems splice site determination has formed a major role as they define the exons and introns of a gene. The first exon differs from internal exons as it lacks an acceptor site as the 5' end of the mRNA is capped with 7-methyl guanosine. Likewise, the last exon is different from other internal exons as it lacks a donor site as the mRNA is terminated with a poly(A) tail attached to the cleavage site. Each internal exon has an acceptor and donor site at its 5' and 3' ends.

These sites along with other regulatory elements recruit an array of protein and RNA factors depending on the splicing signals and remove the intervening sequences or introns from the nascent RNA and stitch the exons together. This process is referred as *Splicing* and details of this process are explained in chapter 1. Recognition of donor sites is relatively easy as the donor signals are more conserved than acceptor signals (for details refer reviews, Black, 2003; Jurica and Moore, 2003; Reed, 2000).

Several programs are available in the public domain that can detect donor and acceptor sites in the mRNA sequences. They function either as a stand-alone splice site finder or as part of gene prediction programs. The performance of most of the gene finding systems is greatly influenced by their accuracy at determining splice sites. In theory, a program that could identify all splice sites would do a nearly perfect job of *ab initio* gene finding as it would determine all protein coding regions correctly given the transcription start site (Brendel and

Kleffe, 1998; Burge and Karlin, 1997; Solovyev and Salamov, 1997). However identifying all the potential splice sites and gene structures is difficult, in particular because each gene may be spliced in a number of different ways. Recent experimental data suggests that in human at least one-third of genes are alternatively spliced (Ashurst and Collins, 2003). This increases the complexity of predicting donor and acceptor splice sites in the RNA sequences and the identification of gene structure by *ab initio* programs.

Thus in order to meet the objective of developing an *ab initio* gene prediction program based on regulatory signals, here I attempt to create a donor and acceptor site model using the EAS system explained in chapter 2.

4.2 Datasets

For the purpose of training and testing the model, I used annotated splice sites from human genomic sequences from the database, *SpliceDB* (Burset *et al.,* 2001). From this database, 28468 canonical and non-canonical human splice pairs were extracted. After removing splice site sequences with undetermined base pairs (denoted as ‘N’), 24808 donor and 24894 acceptor splice sites were dumped to derive a positive dataset. From the 24808 donor sites, 500 sequences of 82 bases (40 bases on either direction of the consensus site + the 2 consensus bases at the donor site) each were extracted randomly to form a training set for donor sites. Likewise, 500 sequences of 82 bases (40 bases on either direction of the consensus site + the 2 consensus bases at the acceptor site) each picked randomly from 24894 acceptor sites formed the training set. GT and AG of donor and acceptor site respectively formed the anchor point and both the sets of sequences are collectively referred as *positive datasets*.

Eponine classifier requires another set of sequences where donor and acceptor signals are unlikely to occur. As this is the critical step in deriving the model, I tried different set of negative sequences – random, exonic and intronic sequences. A set of 946 random sequences of 82 bases each were dumped from chromosome 20 to form a negative set. Using BLASTN, an all-against-all search was done and sequences were removed such that none had more than 90% sequence identity with any of the others. This left a set of 561 sequences, which was referred as *randneg* negative dataset.

A list of 781 exons of at least 500 bases was dumped from chromosome 20 and 22 to derive the exonic negative dataset. After removing 52 redundant exons (with same identifiers), 82 base sequences from the centre of each exon were extracted. Then I did an all against all search on these 729 (781-52) sequences using BLASTN and again sequences were removed such that none had more than 90% sequence identity to any other. This formed a set of 500 sequences, *exonneg*. By extracting sequences from the middle of the exon, any sequence elements near exon-intron and intron-exon boundaries are excluded from the negative dataset.

Likewise, 1000 introns from chromosome 20 of at least 500 bases were used to form the intronic negative dataset. Redundant introns with same identifiers were removed leaving 891 introns. From this set, 82 base sequences from the centre of each intron were extracted. After removing any sequence with more than 90% sequence identity to any other as detected using BLASTN, a set of 507 sequences were dumped to form the *intronneg* dataset.

Another set of 500 sequences were created from randneg, exonneg and intronneg datasets by picking random sequences. This set was referred to as *combneg*.

4.3 Training the splice site models

With the availability of positive and negative datasets for training and testing, I used 900 sequences (450 positive + 450 negative) for training and the remaining 100 sequences (50 positive + 50 negative) for testing the models. The test sequences are unseen while training and used only for initial testing of the models. I trained an Eponine donor site model by allowing the trainer to run for 6000 cycles. Each cycle took a few seconds in a 256MB RAM PIII Pentium laptop. The anchor point for the training is fixed at the first base of the donor consensus sequence - *i.e.* G in GT consensus signal. The window size for training the model was restricted to 35 bases on either side of the anchor point as any constraints selected near to the edges of the sequence are likely to cause the trainer to trip, leading to problems in determining the Gaussian distribution for the constraint. As we are interested in capturing only the donor consensus signals rather than all regulatory elements conserved in exon or intron, the window size of 35 bases was found sufficient. During the training, models were dumped at various checkpoints to analyse the performance of the trainer and determine the convergence of the model. Figure 36 shows a typical donor site model learnt by the EAS system. The model seems to be complex with positive and negative overlapping constraints. Different training cycles with modified parameters and negative datasets showed similar results and the model did not converge even at varied numbers of training cycles.

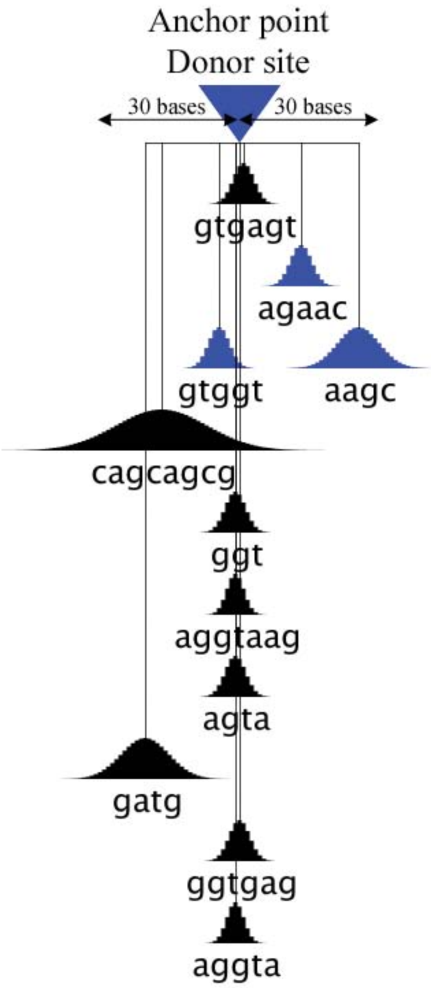


Figure 36. Donor site model trained from SpliceDB sequences

Likewise, the acceptor splice site model was also found to be complex with more negative constraints (Figure 37). Here, A of AG in the consensus acceptor site was used as the anchor point with a window size of 35 bases.

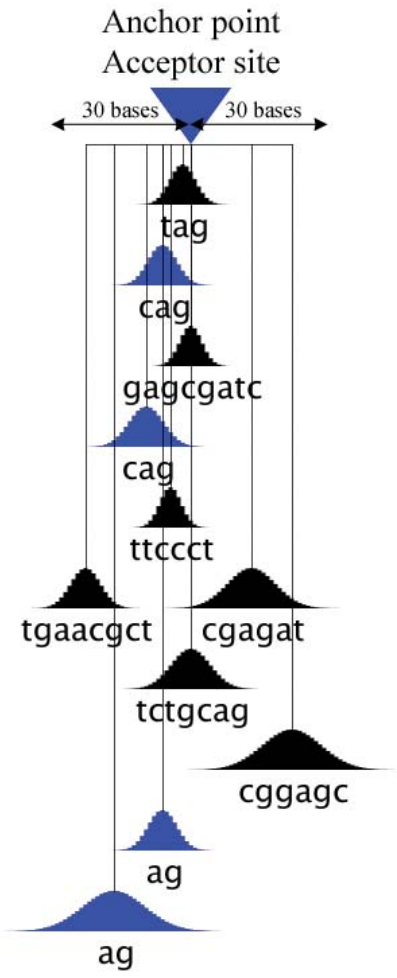
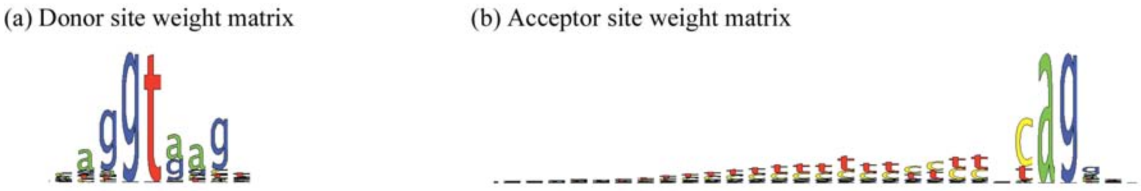


Figure 37. Acceptor site model trained from SpliceDB sequences

4.4 Refining the models

So to refine the models explained earlier, I adopted two strategies –

Eponine models are created by linking positional weight matrices scanned while training from the positive sequences. As explained in chapter 2, an initial set of constraints or weight matrices are sourced from the training set and on training, informative constraints are kept, removing the uninformative ones. This process continues until the datasets can be modelled using a set of sequence motifs. Thus in the EAS model, the complex natural data are simplified to a sparse model by projecting the data into feature space. However in some instances, selecting those few constraints that can effectively define the feature space of the datasets might be difficult and hence the models are unlikely to converge. Here donor and acceptor site models have reached this point and hence to facilitate the learning process, I used weight matrices calculated from donor and acceptor sites of chromosome 22 as an input along with the DNA sequences. This reduced the difficulty in learning an appropriate set of constraints that can optimally classify positive from negative sequences. From chromosome 22, experimentally annotated 2348 donor and acceptor sites were dumped from coding genes and weight matrices showing the probability distribution of the nucleotides at each position of the sequence was constructed. Figure 38 shows the probability distribution for 30 nucleotides around donor and acceptor sites. The donor weight matrix has captured the canonical consensus sequences reported earlier. Likewise, the acceptor site matrix has captured the consensus sequence along with the polypyrimidine motif preceding the signal. These weight matrices are used as input along with DNA sequences to learn a sparse EAS model by including the following lines in the parameter file.



(a) Donor site weight matrix

(b) Acceptor site weight matrix

Figure 38. Nucleotide Distribution at (a) Donor and (b) Acceptor site from chromosome 22 sequences

```
<child jclass="eponine.model.WMFileBasisSource">
    <string name="fileName" value="./donorORacceptorWMfile.xml" />
    <int name="position" value="-25" />
     <double name="minDistWidth" value="2.5" />
    <double name="maxDistWidth" value="50.0" />
    <boolean name="reversible" value="false" />
</child>
```

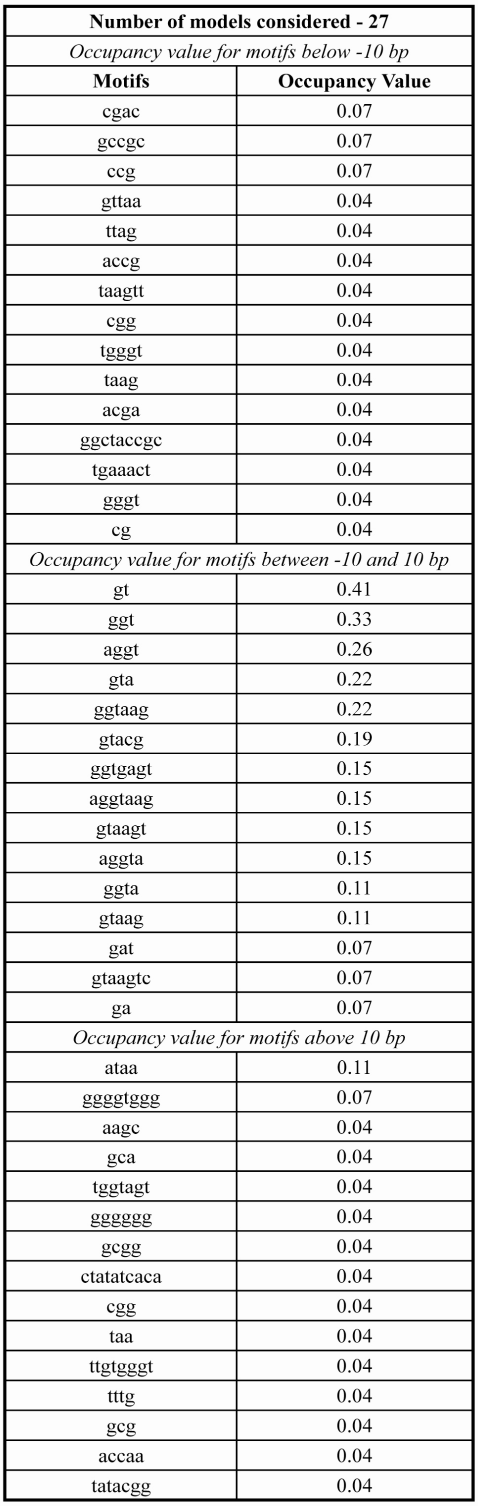
Positional distribution of the constraints learnt by Eponine is usually captured as a Gaussian distribution. However, the system allows for various other distributions to be used depending on the conditions of the problem. Here as the consensus sequence of the donor and acceptor sites in the training sequences are less likely to vary in their position, a Delta distribution (instead of Gaussian distribution) is appropriate in modelling the positional variations. Thus for learning splice site models I implemented a Delta distribution along with the Gaussian for capturing the offset values of the constraints relative to the anchor point. I added the following lines to the parameter file –

```
<child jclass="eponine.model.MakeDeltaBasisSource" />
<child jclass="eponine.model.BreakDeltaBasisSource" />
```

With these changes I retrained the model with a new set of data derived from chromosome 22. From chromosome 22, 550 donor site sequences of 350 bases each (50 bases upstream and 300 downstream of GT signal. 300 bases downstream of GT is used in the aim to find out any unknown signals in this region) were dumped to form the positive dataset. Likewise, 550 acceptor site sequences of 350 bases each (300 bases upstream and 50 bases downstream of AG signal. 300 bases upstream of AG signal will include the well known Branch Point region) formed the acceptor site positive dataset. An equal number of random sequences from chromosome 22 formed the negative set. From the positive and negative datasets, 1000 sequences (500 positive + 500 negative) were used for training and the remaining 100 sequences (50 positive + 50 negative) for initial testing of the model. G of GT and A of AG in donor and acceptor signals respectively were used as anchor points. For donor site mode, the window size limits are set to 42 bases upstream and 290 bases downstream of the anchor point whereas for the acceptor model, the limits are 290 bases upstream and 40 bases downstream of the anchor point. Although the window size spans to 290 bases, the positional constraints learnt by both models are within 30 bases from anchor points, emphasising the fact that regulatory elements determining splice sites are closely linked with donor and acceptor signals.

Figure 39 and Figure 40 show the refined models with new parameters and datasets for donor and acceptor sites respectively. Figure 41 shows the position, constraint weight and Gaussian width of each constraint learnt by the donor model. The consensus signal at the donor sites are captured by 3 positive constraints. One of the constraints had a Delta distribution meaning no positional variation in the motif. All the constraints emphasize the importance of the GT bases in the consensus motif. From the intronic sequences, a constraint rich in G nucleotides was learnt and positioned at 28 bases downstream of the anchor point. The biological importance of the motif is not known. Table 6 shows the occupancy value of the top 15 motifs represented in various donor site models.

Table 6. Occupancy value for motifs detected in the donor site models.



Number of models considered - 27	
Occupancy value for motifs below -10 bp	
Motifs	**Occupancy Value**
cgac	0.07
gccgc	0.07
ccg	0.07
gttaa	0.04
ttag	0.04
accg	0.04
taagtt	0.04
cgg	0.04
tgggt	0.04
taag	0.04
acga	0.04
ggctaccgc	0.04
tgaaact	0.04
gggt	0.04
cg	0.04
Occupancy value for motifs between -10 and 10 bp	
gt	0.41
ggt	0.33
aggt	0.26
gta	0.22
ggtaag	0.22
gtacg	0.19
ggtgagt	0.15
aggtaag	0.15
gtaagt	0.15
aggta	0.15
ggta	0.11
gtaag	0.11
gat	0.07
gtaagtc	0.07
ga	0.07
Occupancy value for motifs above 10 bp	
ataa	0.11
ggggtggg	0.07
aagc	0.04
gca	0.04
tggtagt	0.04
gggggg	0.04
gcgg	0.04
ctatatcaca	0.04
cgg	0.04
taa	0.04
ttgtgggt	0.04
tttg	0.04
gcg	0.04
accaa	0.04
tatacgg	0.04

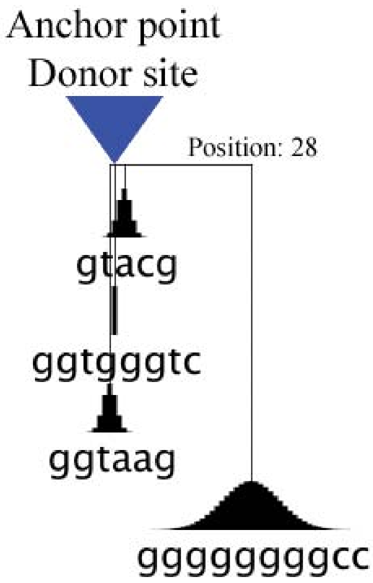


Figure 39. Donor site model trained from chromosome 22 sequences and donor site weight matrix

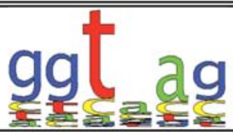
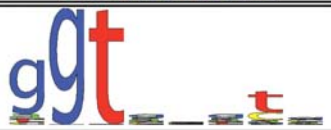
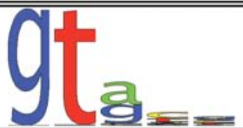
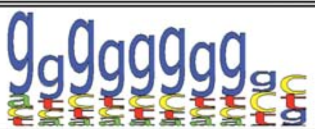
MOTIFS	POSITION	CONSTRAINT WEIGHT	GAUSSIAN WIDTH
	-1	9.54	1.33
	0	4.66	-
	2	2.81	1.33
	28	5.32	5.83

Figure 40. Position constraints of donor site model learnt while training chromosome 22 sequences

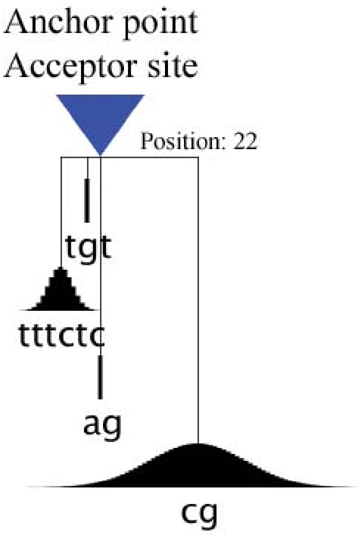
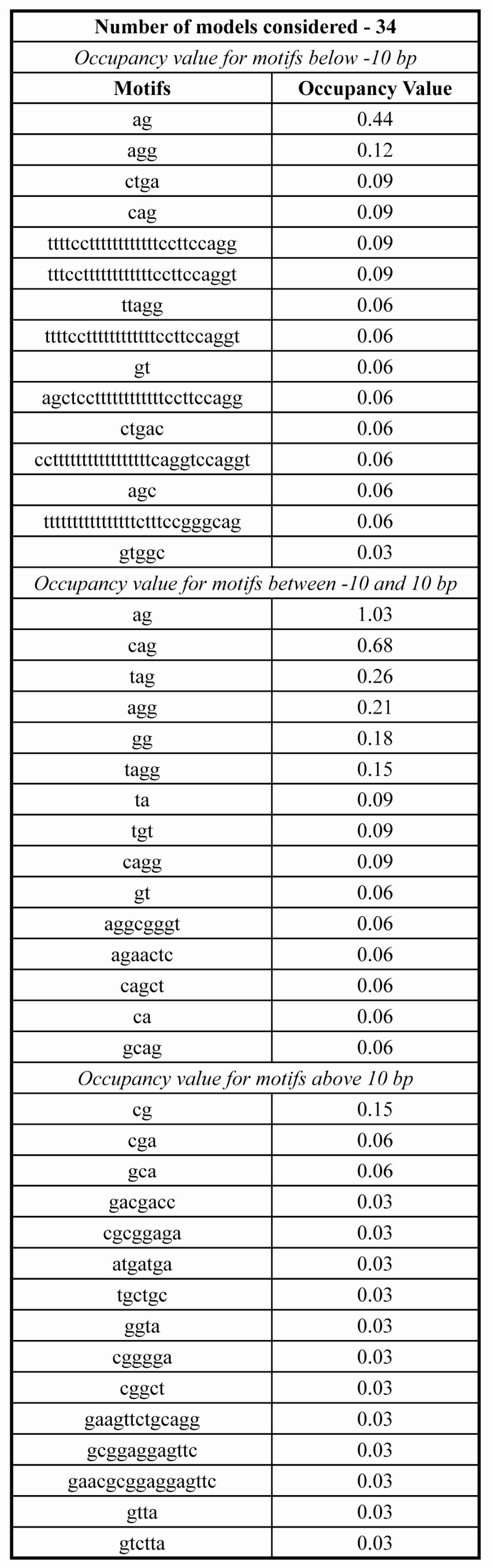


Figure 41. Acceptor site model trained from chromosome 22 sequences and acceptor site weight matrix

Figure 42 shows the properties of positional constraints learnt by the acceptor model. The consensus AG signal is well captured along with the polypyrimidine motif known earlier. A CG rich motif was found 22 bases downstream of the acceptor site (in exonic sequence) with a Gaussian distribution width of 10.96 and constraint weight of 9.42. The values emphasise the signal is important but the role of the motif is not known. The frequency of distribution of these motifs and other top 15 motifs in different regions of the model are shown in Table 7. A value of more than 1 indicates the motif is represented more than once in few models.

Thus, this second training approach, seeding the training with a position distribution and making use of a Delta function, was able to generate simple models containing only positive weights (Figure 39 and Figure 41) compared to the earlier training approach (Figure 36 and Figure 37).

Table 7. Occupancy value for motifs detected in the acceptor site models.



Number of models considered - 34	
Occupancy value for motifs below -10 bp	
Motifs	**Occupancy Value**
ag	0.44
agg	0.12
ctga	0.09
cag	0.09
ttttcctttttttttttttccttccagg	0.09
tttcctttttttttttttccttccaggt	0.09
ttagg	0.06
ttttcctttttttttttttccttccaggt	0.06
gt	0.06
agctcctttttttttttttccttccagg	0.06
ctgac	0.06
cctttttttttttttttttcaggtccaggt	0.06
agc	0.06
ttttttttttttttttctttccgggcag	0.06
gtggc	0.03
Occupancy value for motifs between -10 and 10 bp	
ag	1.03
cag	0.68
tag	0.26
agg	0.21
gg	0.18
tagg	0.15
ta	0.09
tgt	0.09
cagg	0.09
gt	0.06
aggcgggt	0.06
agaactc	0.06
cagct	0.06
ca	0.06
gcag	0.06
Occupancy value for motifs above 10 bp	
cg	0.15
cga	0.06
gca	0.06
gacgacc	0.03
cgcggaga	0.03
atgatga	0.03
tgctgc	0.03
ggta	0.03
cgggga	0.03
cggct	0.03
gaagttctgcagg	0.03
gcggaggagttc	0.03
gaacgcggaggagttc	0.03
gtta	0.03
gtctta	0.03

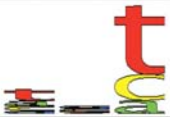
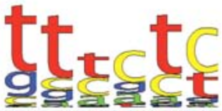
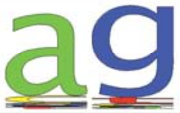
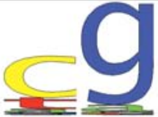
MOTIFS	POSITION	CONSTRAINT WEIGHT	GAUSSIAN WIDTH
	-3	3.34	-
	-9	7.99	2.71
	0	4.65	-
	22	9.42	10.96

Figure 42. Position constraints of acceptor site model learnt while training chromosome 22 sequences

4.5 Validating and testing the models

I tested the performance of these refined models of donor and acceptor sites using datasets derived from chromosome 20. From the VEGA annotation of chromosome 20 (on build NCBI 33) (Ashurst, 2002), I extracted 614 genes. These genes are defined as ‘Known’ or ‘Novel_CDS’ and ‘Novel_transcript’ in the database. ‘Putative’ and ‘Pseudogene’ categories are not considered. Constitutive and alternative exons known in these 614 genes totalled to 8771 and they were dumped to extract donor and acceptor sites. Out of 614, 42 genes are single exon genes and are thus omitted in this study. From the remaining 572 genes, 8037 and 8141 constitutive and alternative donor and acceptor sites are extracted respectively. After removing the redundancy present in the set of donor and acceptor sites, 5835 and 6166 unique donor and acceptor sites are found and I used this set to test the performance of the models.

Coverage is defined as the set of genes with at least one prediction within it. The value is calculated from the number of the genes with a prediction over the total number of genes (572). Accuracy is defined as the set of predictions that matches the annotated sites over total predictions within the gene. Any predictions (in the same strand matching the annotation, prediction on the opposite strand are considered as false positives) present within 10 bases of the centre point of annotated sites are considered as true predictions. Any other predictions, within transcription unit or intragenic region, are considered as false positives.

Coverage was calculated at exon level as well and in this case, exons with predictions are counted as predicted.

Figure 43 and Figure 44 show the coverage and accuracy of the donor and acceptor site model respectively as a ROC curve. The donor site model registers higher accuracy at low coverage rate. Comparatively the performance of the acceptor site model is less than the donor site model. This is expected as the acceptor sites are less conserved and they are relatively more difficult to identify than donor sites.

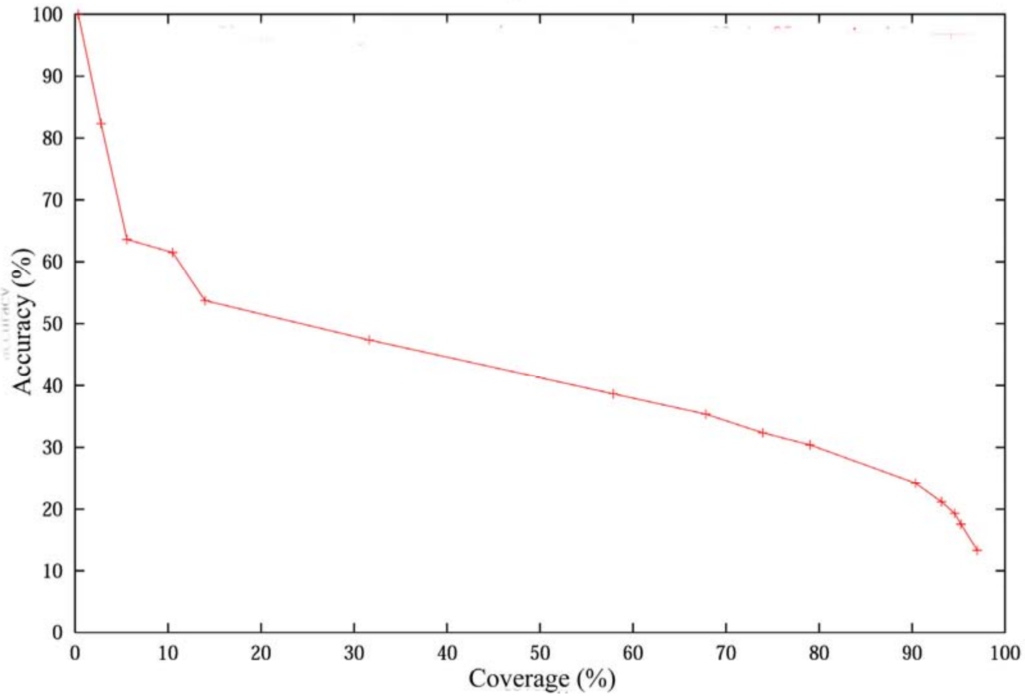


Figure 43. ROC curve for Eponine donor site model on chromosome 20 dataset

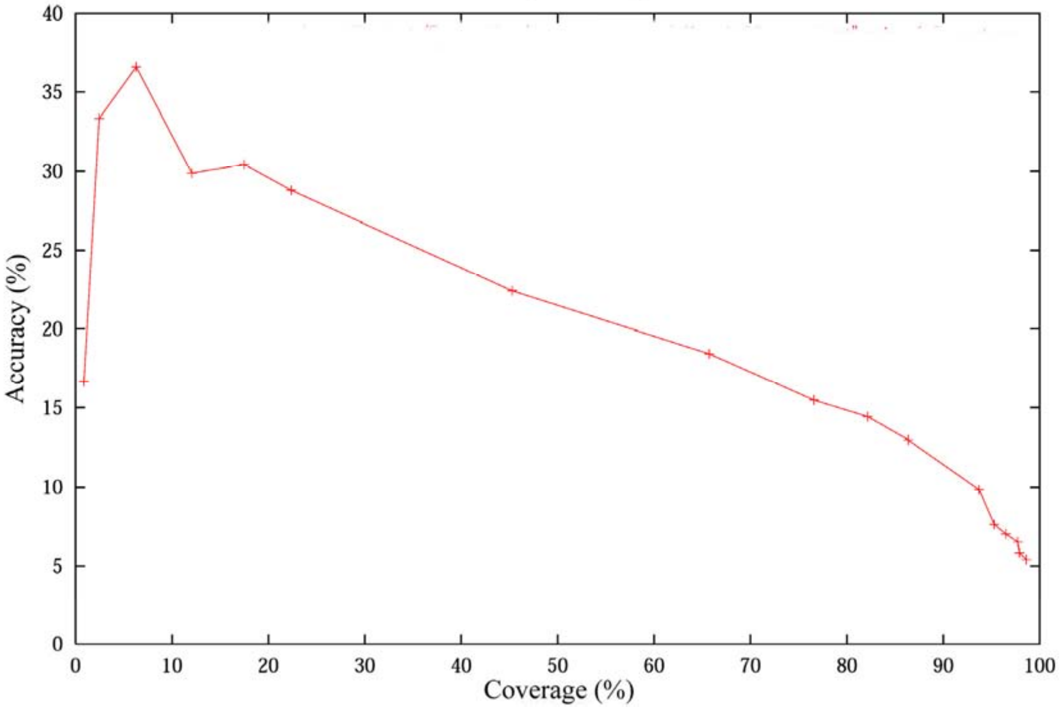


Figure 44. ROC curve for Eponine acceptor site model on chromosome 20 dataset

4.6 Position accuracy of the models

I calculated the densities of predictions of the donor and acceptor site model in relation to the annotated donor and acceptor sites in chromosome 20. The histograms of the densities calculated are shown in Figure 45 and Figure 46. The X-axis represents the position of the sequence relative to the annotated sites and the Y-axis represents the density of predictions at each position. In both figures the densities are drawn for 100 bases upstream and downstream of the annotated site.

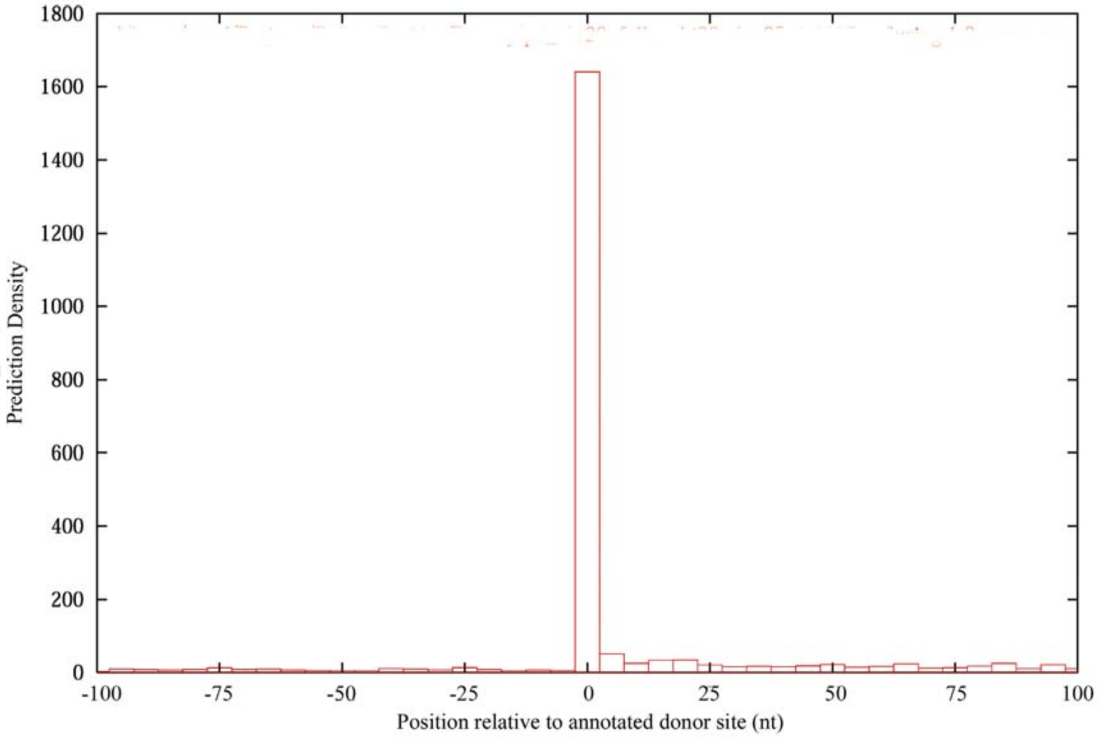


Figure 45. Prediction density for donor site model relative to annotated sites

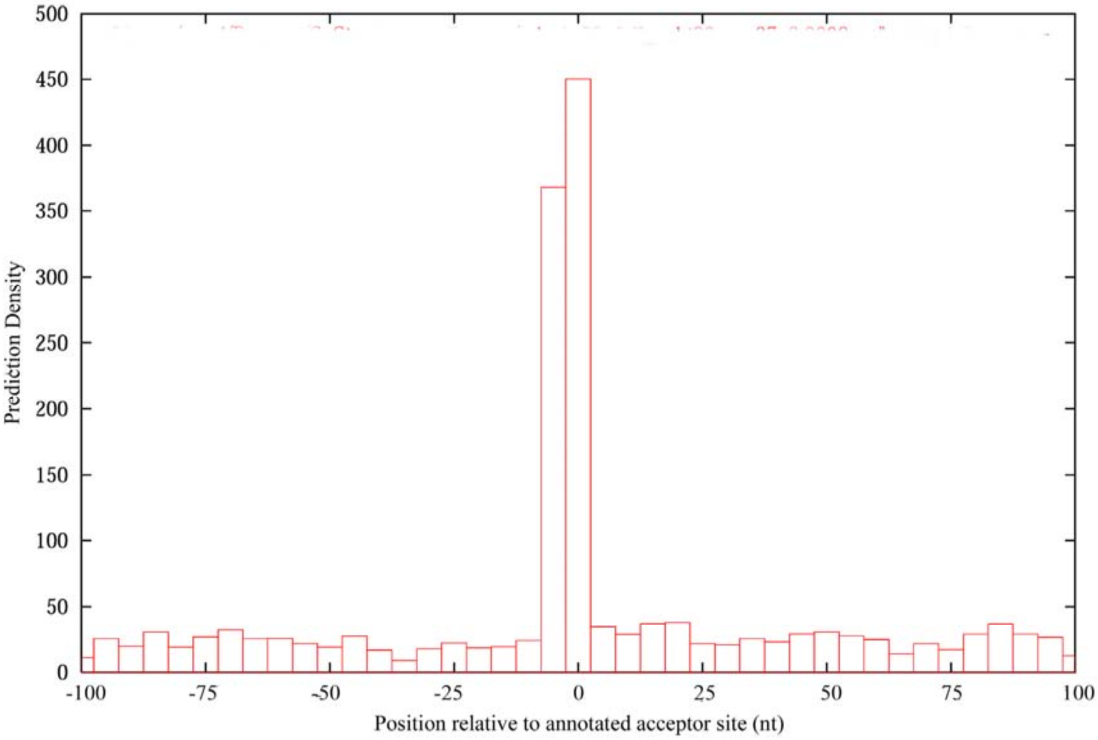


Figure 46. Prediction density for acceptor site model relative to annotated sites

The results show a clear peak exactly on the site of annotated donor and acceptor sites. The accuracy of the predictions by the donor site model corresponds to within 5 bases from the annotated site whereas acceptor site model predictions are within 10 bases. Figure 46 shows few predictions on either side of the peak: this might correspond to the false positives of the model but some of them might be due to alternative acceptor sites in the sequence.

Both the models are good at detecting the directionality of the sites and the figures shown here are for predictions in the same strand as the annotation. Density histograms of the predictions on the reverse strand show no peak at the annotated site.

4.7 Comparison with other models

I compared the Eponine splice site models with two other splice site programs available in the public domain. They are –

StrataSplice (Levine, 2001a) – This program uses a new splice site prediction model that combines the local GC content (80 bases upstream and downstream of the splice site) with a standard probabilistic pattern recognition technique. The method predicts both canonical (GT-AG) and minor variant (GC-AG) splice sites and is designed to integrate easily into a variety of gene prediction and annotation systems. The performance of the model is better in gene-rich high GC regions.

GeneSplicer (Pertea *et al.*, 2001) – This program uses a decision tree method called maximal dependence decomposition, first developed by Burge and Karlin (Burge and Karlin, 1997), enhanced with markov models that capture additional dependencies (16 bases around donor site and 29 bases around acceptor sites) surrounding the splice sites. This method considers only a small window around the splice junctions, which contains most of the information recognised by the spliceosome. It also takes into account the coding and non-coding sequence switch at the splice junctions and the local score optimality feature developed by Brendel and Kleffe. (Brendel and Kleffe, 1998).

I used the test set described earlier – 5835 donor and 6166 acceptor splice sites from 572 genes from chromosome 20 to compare the performance of the methods. As both programs are available in the public domain (Levine, 2001b; Pertea, 2001), I downloaded them and scanned chromosome 20 sequence locally in a 1GB Compaq Tru64 UNIX machine. Donor sites are predicted with higher accuracy by StrataSplice than acceptor sites. I collected the StrataSplice predictions for donor sites at posterior probabilities: 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10, 0.12, 0.14, 0.18, 0.20, 0.24, 0.26, 0.30, 0.34, 0.38, 0.42, 0.44, 0.48, 0.50, 0.54 and 0.58 while for acceptor sites at 0.02, 0.03, 0.04, 0.05 and 0.06. No acceptor site predictions are made for probabilities above 0.06. Likewise donor and acceptor predictions of GeneSplicer are extracted at score thresholds: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 and 22. Again donor site predictions by GeneSplicer had higher score values than acceptor site predictions.

Figure 47 and Figure 48 shows the performance of Eponine, StrataSplice and GeneSplicer on chromosome 20. Predictions within 10 bases from the annotated donor and acceptor site predictions are considered as true predictions. Coverage and accuracy are measured as described above. Figure 47 shows the performance of Eponine is comparable with StrataSplice although less than GeneSplicer. However coverage and accuracy of Eponine acceptor site model and StrataSplice are significantly less than GeneSplicer (Figure 48). To analyse the coverage of exons by the three programs, I did a ROC curve taking only exons having a prediction within 10 bases from donor or acceptor sites as true predictions while calculating coverage. 5835 exons from 572 transcripts are used for this analysis. Figure 49 and Figure 50 shows the exon coverage and accuracy for donor and acceptor sites respectively. GeneSplicer again performs better in detecting the exons boundaries better than Eponine and StrataSplice.

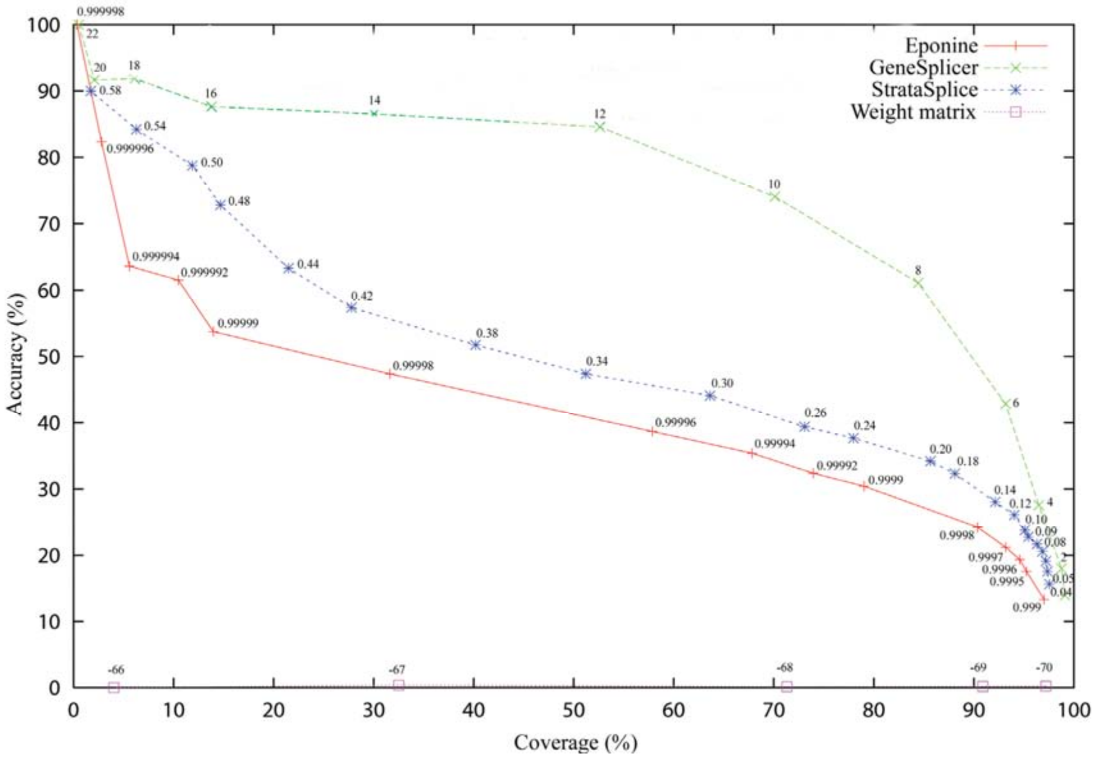


Figure 47. ROC curves on donor sites in chromosome 20 for Eponine, GeneSplicer, StrataSplice and Weight matrix.

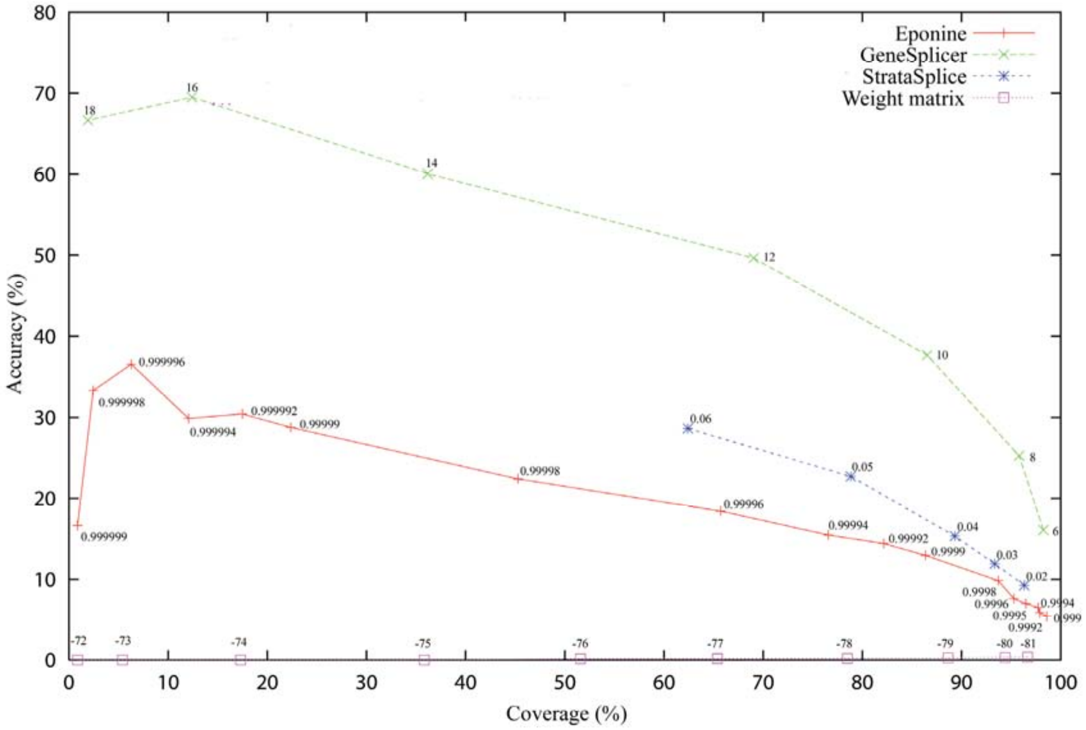


Figure 48. ROC curves on acceptor sites in chromosome 20 for Eponine, GeneSplicer, StrataSplice and Weight matrix.

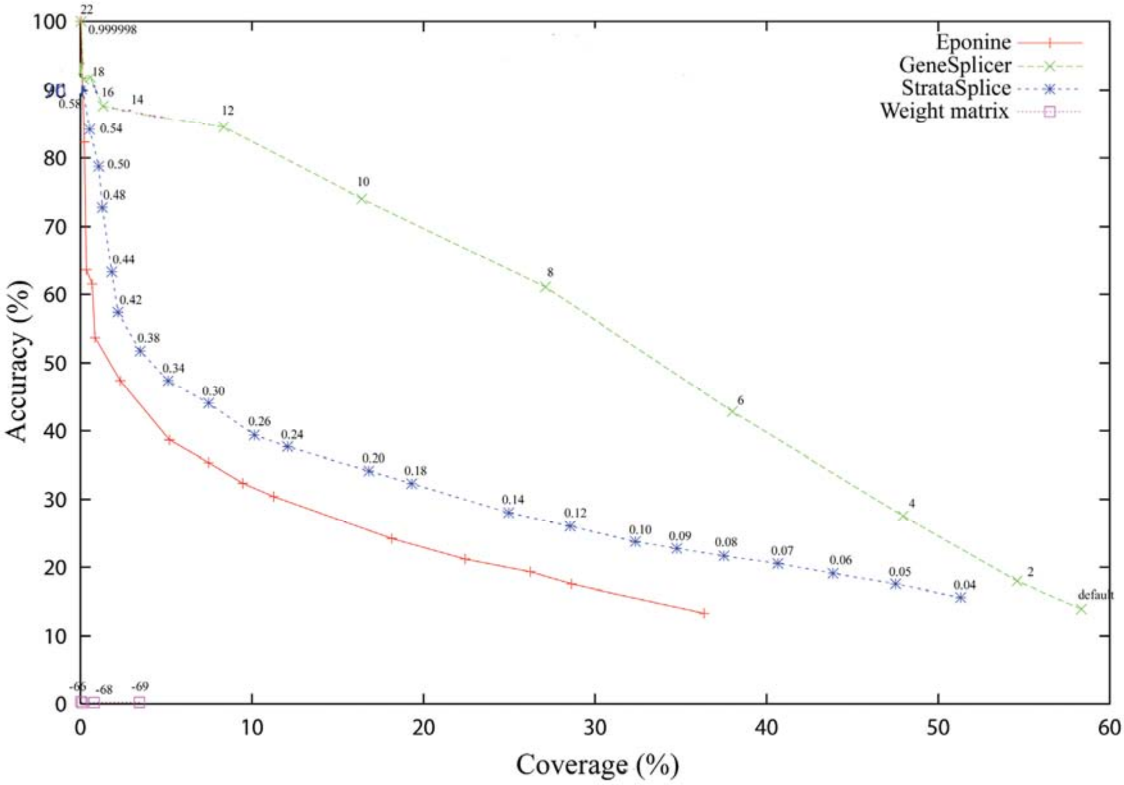


Figure 49. Exon coverage and accuracy on donor sites in chromosome 20 for Eponine, GeneSplicer, StrataSplice and Weight matrix.

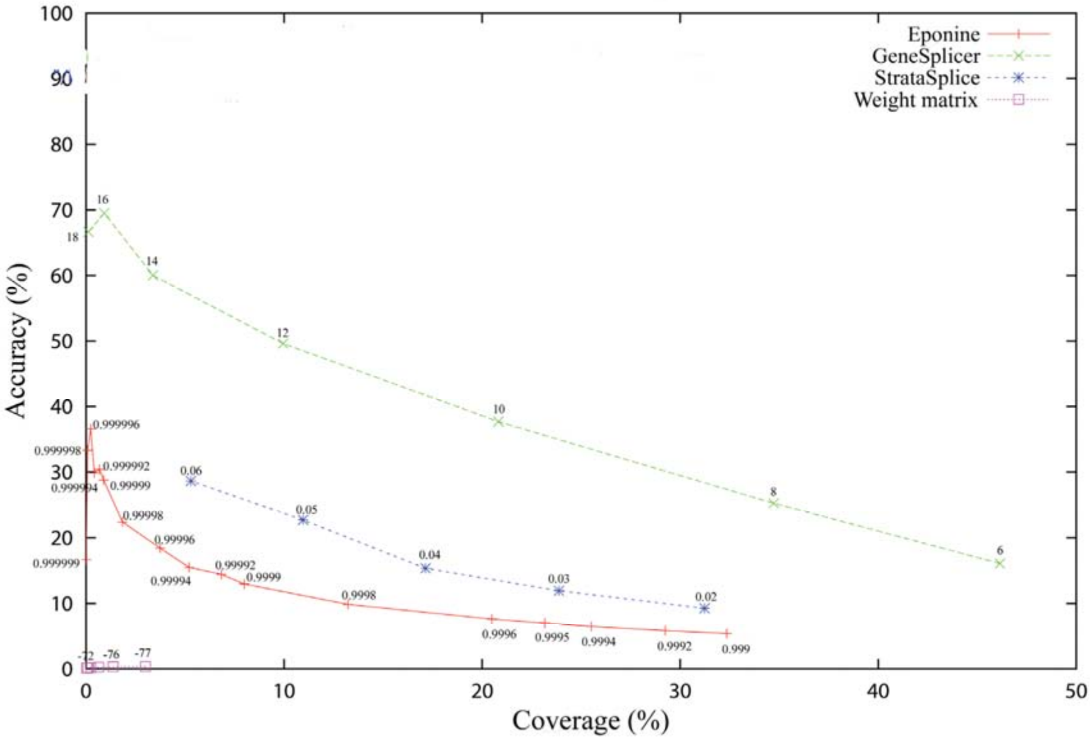


Figure 50. Exon coverage and accuracy on acceptor sites in chromosome 20 for Eponine, GeneSplicer, StrataSplice and Weight matrix.

The results are not particularly surprising, as Eponine model is based only on positional weight matrices, whereas, StrataSplice uses local variation in GC content to differentiate true and false signals. The performance of StrataSplice has been shown to be higher at gene-rich regions of chromosomes (Levine, 2001a). GeneSplicer, apart from modelling sequence elements present near splice sites, uses coding/non-coding potential present in exons/introns near splice sites. A significant number of false positives by GeneSplicer are removed by choosing a splice site in a favourable sequence context. The context includes first, the availability of an appropriately spaced complementary splice site such that this pair of sites defines a potential intron and second, absence of nearby sites of the same type with higher score which could favourably compete with the given site for splicing factors (for details refer, Brendel and Kleffe, 1998). Employing both the favourable sequence context strategy and the coding potential is against the objective of developing an *ab initio* gene prediction system purely based on gene regulatory signals. Hence, although informative these strategies are not included in the Eponine models.

However, Eponine splice site models are shown to have significantly better performance than using donor and acceptor site weight matrices only (Figure 47 and Figure 48). These matrices are derived from chromosome 22 splice sites and are described in Figure 38. I scanned chromosome 20 sequence with the donor and acceptor site weight matrices and compared the predictions with the test set of 5835 donor and 6166 acceptor sites described earlier. Predictions extracted at different thresholds in both donor and acceptor site scan indicates weight matrices alone are not informative in predicting the splice sites. Similar results are found in predicting exon boundaries as well (Figure 49 and Figure 50). Thus Eponine splice site models although perform less well than GeneSplicer, they are found better than weight matrices.

Figure 51 shows the positional accuracy of Eponine models (Figure 45 and Figure 46) are equivalent to the predictions of StrataSplice and GeneSplicer.

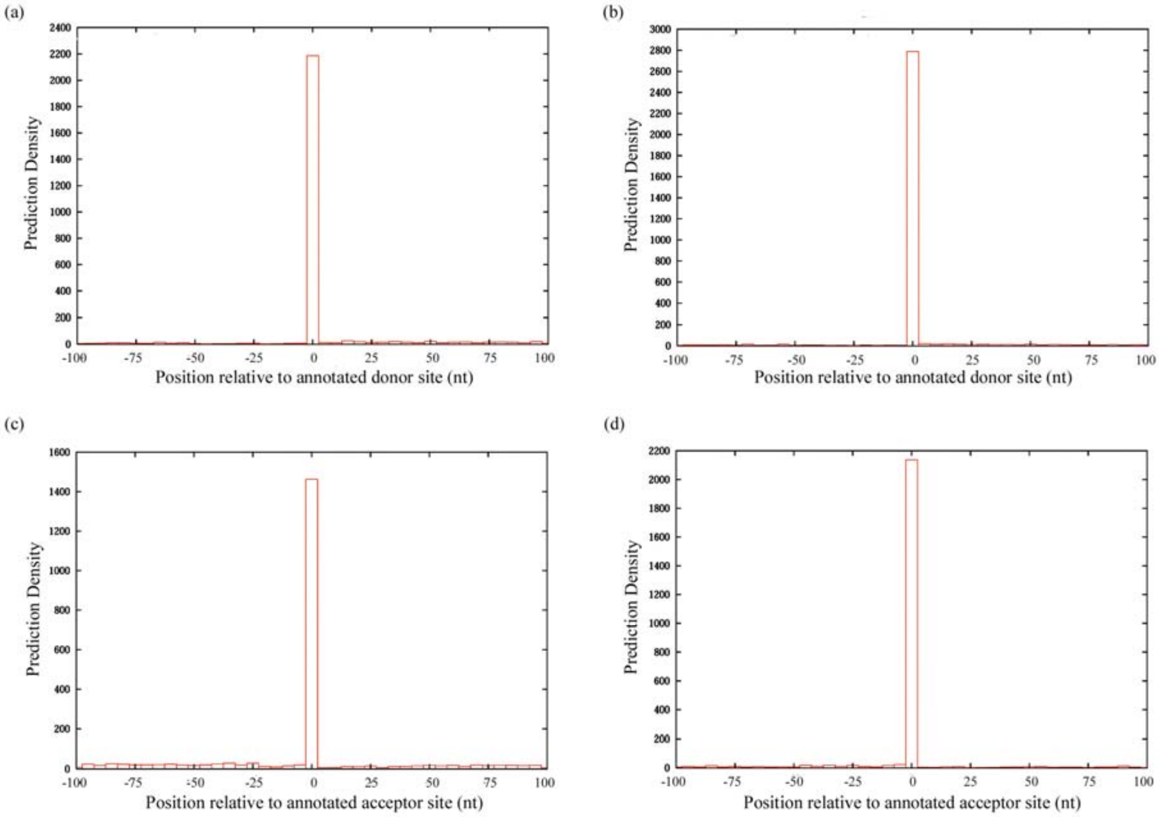


Figure 51. Prediction densities for StrataSplice and GeneSplicer donor and acceptor site predictions (a), (b) Densities for StrataSplice and GeneSplicer donor site predictions relative to the annotated site respectively (c), (d) Densities for StrataSplice and GeneSplicer acceptor site predictions relative to the annotated site respectively

Thus the comparison shows that the performance of Eponine is comparable with StrataSplice and GeneSplicer given the amount of information used to process DNA sequences in identifying splice sites. Also, it reveals the scope for improvement of Eponine predictions using local GC content in the gene rich regions.

4.8 Concluding remarks

With the requirement of splice site models to meet the objective of developing an *ab initio* model based on regulatory elements, I created donor and acceptor site models. Initial training cycles did not yield a sparse model and hence I used weight matrices derived from chromosome 22 as an input to the EAS system along with DNA sequences. A Delta distribution was used to capture the positional variation as it suits the problem better than a

Gaussian distribution. The models learnt the known consensus signals and they are used to predict splice sites in chromosome 20. On comparison, the performance of the Eponine model was found better than using weight matrices alone. Availability of these models facilitates construction of a gene prediction program and to determine the structure of the predicted genes.

MODELLING TRANSLATION START AND STOP SITES

5.1 Introduction

As explained in chapter 1, transcription and translation, understood to be coupled together in prokaryotic organisms, have now been found to be interlinked in eukaryotes as well. Non-sense Mediated Decay (NMD) and protein synthetic capability in the nucleus add support for this view (for review, Cook, 1999). NMD is triggered due to the encounter of a pre-termination codon by the translating ribosome machinery (for details refer, Hillman *et al.*, 2004; Iborra *et al.*, 2004). So just like transcription, the translation mechanism is also under the control of regulatory elements on the RNA (transcribed from DNA). Translation start and stop signals are important regulatory signals and so far various methodologies have been used to study them.

With detailed knowledge of translation regulatory elements and machinery, computational detection of translation start and stop codons and their auxiliary sequences has been relatively easy and techniques from simple positional weight matrices to artificial neural networks to support vector machine have been used for this purpose. Almost all the translation start models are based on the important *Kozak* consensus sequence (Kozak, 1987) at the translation initiation site in detecting the start codon. However later algorithms, in an effort to improve prediction coverage and accuracy, used other features as well. Among them, detecting the coding potential of the sequence following the start codon, open reading frame length (the distance between the predicted start and stop codon) and distance of first ATG from the start of the sequence took a serious role. In addition, a few techniques even analysed the density of trimers, tetramers and pentamers in the sequences before the start codon and its property for non-coding potential. These properties, although improving the prediction system, giving it better accuracy and coverage, have limited it to be used successfully in cDNA and EST sequences or incorporated into an *ab initio* gene prediction system. However as standalone programs for prediction on genomic sequences they are likely to make numerous errors. So to address this issue, here I attempted to use the Eponine model trainer to learn translation start and stop signals. Also, the translation models can be used with other Eponine models in devising an *ab initio* gene prediction system like splice site models explained in the previous chapter.

In the remainder of this chapter, I will explain the datasets and parameters used to derive translation models and compare their performance with existing programs.

5.2 Datasets

5.2.1 Translation start model

Deriving a reasonable dataset with better annotation is one of the key factors in deriving any prediction models. Screening for such data from a massive amount of unannotated and incomplete cDNAs and ESTs is a formidable task. With annotated data the issue of upstream ATGs has to be addressed. Translation as explained earlier is known to occur by a cap dependent scanning mechanism or cap independent internal initiation process. The common cap dependent process helps the ribosome to start translation from the first ATG it encounters from the 5’ end. However it is not always the case and at times ATGs further downstream can be used. The presence of multiple ATGs at the 5’ end may confuse annotators leading to the identification of wrong ATGs as translation initiation sites. One estimate shows about 37% of human and 36% of mouse sequences in the 5’ UTR database (Pesole *et al.,* 1996) have upstream ATGs with reference to annotated translation start sites (Rogozin *et al.,* 2001). Thus deriving a dataset with correctly annotated translation start sites is one of the most difficult steps.

As explained earlier, Eponine trainer requires two kinds of dataset – A *positive dataset* having DNA sequences that are likely to have translation start sites and a *negative dataset* with sequences of no such sites. Here I used two sets of positive sequences – one from genomic and another from cDNA for training an EAS model.

(A) Genomic sequences – Using the annotation of coding sequences in human chromosome 22 by (Collins *et al.*), 330 sequences with translation start sites and at least 200 bases of 5’ UTR were extracted. Two hundred nucleotides upstream and downstream of the ATG codon formed the positive set, *pos-1*.

Likewise, 200 bases on either side of the start codon from 506 transcripts of 327 genes in chromosome 20 were dumped from VEGA database (Ashurst, 2002). This dataset formed another positive set, *pos-2*. Only transcripts from the ‘known’ gene category from the VEGA database were used here.

Combining both *pos-1* and *pos-2* datasets, a set of 836 transcripts with annotated translation start sites was formed *pos-3*.

Random and intergenic sequences of 400 nucleotides from chromosome 20 and chromosome 22 were dumped to form negative datasets for training the EAS model from *pos-1*, *pos-2* and *pos-3*. Equal numbers of positive and negative sequences were used for each training cycle.

(B) cDNA sequences – From the Reference Sequence database (RefSeq, Pruitt and Maglott, 2001), 14038 cDNA sequences were dumped in EMBL format. Out of this 5693 sequences were categorised as ‘provisional’, 2350 as ‘predicted’, 2523 as ‘curated’ and 3472 as ‘genome annotation’ based on the types of evidences and annotation done on these sequences. Reviewed RefSeq records represent full length cDNA sequences with manual curation of gene features. Hence I took the 2523 sequences and screened for the ones with at least 200 bases of 5’ UTR and resulted in a subset of 676 sequences. Out of these 676 sequences, only 563 sequences have annotations in the ENSEMBL database (Birney *et al.*, 2004) and thus were used for training purposes. The positive dataset, *mpos-1* was derived by extracting 200 bases on either side of the ATG codon present in these 563 sequences. An against all BLAST (Altschul *et al.*, 1990) search was carried out on the *mpos-1* set to make sure no identical sequences were present in the positive dataset. The remaining 113 (676-563) sequences were used for testing the models.

As training on cDNA sequences will tend to be biased towards learning coding potential of the sequences downstream of the ATG codon, two types of negative datasets were synthesized to tackle it. Two hundred nucleotides of noncoding or intronic sequence (from intergenic or intron regions of chromosome 22) and 200 nucleotides of coding or exonic sequence (from exon regions of chromosome 22) were concatenated together to form a 400 base pair negative sequence. This way, both the positive and negative set had exonic sequences downstream of ATG and hence the trainer is less likely to model the coding potential in the positive set. A set of 563 such sequences (equal to the positive set) formed the negative set, *mneg-1*.

In another negative set (563 sequences), *mneg-2*, the intronic (196 bases) and exonic (197 bases) sequences are concatenated together with AXXATGG sandwiched between them. AXXATGG resembles the consensus sequence near translation initiation sites and by incorporating it in the negative set; the trainer is restricted to learn other position constraints that can meaningfully classify the sequences.

With these different positive and negative datasets I trained the EAS translation start model.

5.2.2 Translation stop model

I used the sequences from the 'PolyA site' database (Tabaska and Zhang, 1999) to derive a positive set for training the translation stop model. The database was formed by aligning ESTs of a UniGene cluster (Wheeler *et al.*, 2004) with all of its DNA and non-EST RNA sequences (for details, read Tabaska and Zhang, 1999). A hundred bases upstream and downstream of the stop codon were dumped from 124 sequences from the database to form a positive set. For a negative set of sequences where translation is unlikely to terminate, I extracted 124 sequences of 200 bases each from random regions of chromosome 20. Thus the randomly picked sequences are equal in length to the positive sequences. Out of 248 sequences (124 positive and 124 negative), I kept apart 28 sequences (14 positive and 14 negative) for initial testing of the model. These 28 sequences are randomly picked during different training runs. The remaining 220 sequences were used for training the model. The 113 human RefSeq cDNA sequences (explained earlier) set apart for testing the translation start model were also used for determining the performance of the translation stop model. The first base in the termination codon was used as the anchor point. Models were also trained with a training set derived from 200 bases upstream and downstream of this anchor point.

5.3 Training the translation models

5.3.1 Translation start model

With the datasets available, I initially used *pos-1*, *pos-2*, *pos-3* positive datasets and random negative sequences to train the EAS translation start model. The nucleotide A in the first codon, ATG was set as the anchor point. As explained earlier each positive and negative sequence is of 400 bases length, spanning 200 nucleotides upstream and downstream of this anchor point. During training the trainer is likely to fish out informative positional constraints from these sequences to identify positive from negative sequences. However selection of any constraints near to the edges of the sequence is likely to cause the trainer to cross the boundary, as it will be difficult to estimate a Gaussian distribution for such motifs. So to avoid such cases, I have limited the window size for screening for constraints to 160 bases either side of the anchor point. This appears to be sufficient to capture any regulatory motifs that determine the translation start site as training done with increased window sizes did not find any new constraints. However reducing the window size from 160 bases to 50 bases (-50 to +50 bases from anchor point) and 20 bases (-20 to +20 bases from anchor point) produced models with only ATG and *Kozak* motifs and thus had less predictive power than previous models.

The trainer with these datasets and default parameters was allowed to run for a maximum of 6000 cycles to learn a simplistic model that can significantly classify translation start site from other sequences. Typically each training run took nearly 1 hour in a personal computer with 1GHz Pentium CPU and 256 MB RAM.

A typical model learnt from the *pos-1* positive dataset and random sequences as the negative dataset was shown in Figure 52a. Different training cycles showed that positional constraints, especially those present downstream of the ATG codon, are not converging and the trainer tended to learn negative constraints. Likewise, models trained from the *pos-2* dataset also showed similar results (Figure 52b). The models from the *pos-2* dataset are even more complex with more negative constraints. Intergenic sequences as the negative dataset did not improve the model.

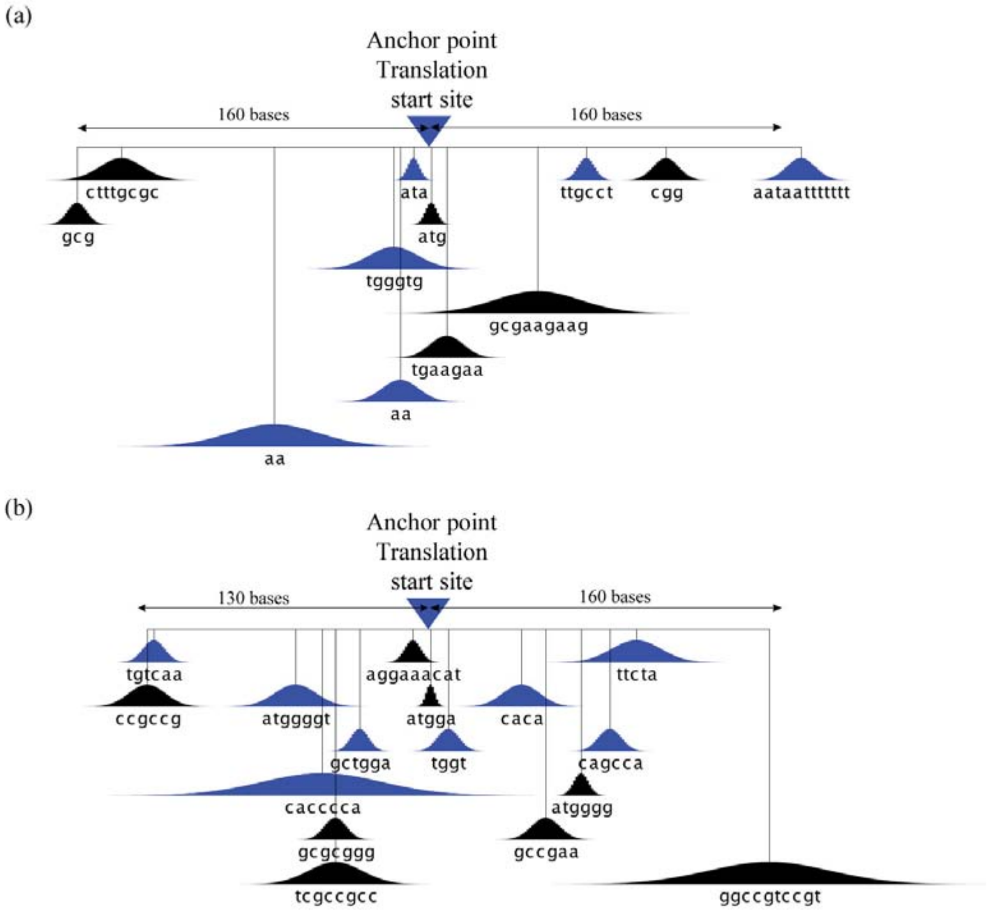


Figure 52. Translation start model trained from (a) chromosome 22 (b) chromosome 20 genomic sequences

Non-convergence of positional constraints might be due to existence of intronic sequences in the positive dataset. While extracting 200 bases downstream of the ATG codon to form the positive set, in cases where sequences followed by the start codon are less than 200 bases, nucleotides from introns are likely to be dumped and added to the positive set. Thus the variation present in the sequences downstream of ATG might be the cause for non-convergence of the model.

Hence to avoid this problem, I switched to training models from cDNA sequences using datasets *mpos-1* and *mneg-1*. A of ATG is again set as the anchor point with the trainer allowed for scanning positional constraints within 160 bases from it. The trainer ran between 5000 and 8000 cycles during various training trials. Examining different trained models showed constraints that are positioned between 140 bases upstream and 120 bases downstream of the anchor point. This suggests that motifs that can identify translation start sites are closely associated with the start codon. Figure 53 shows a typical model trained from cDNA sequences.

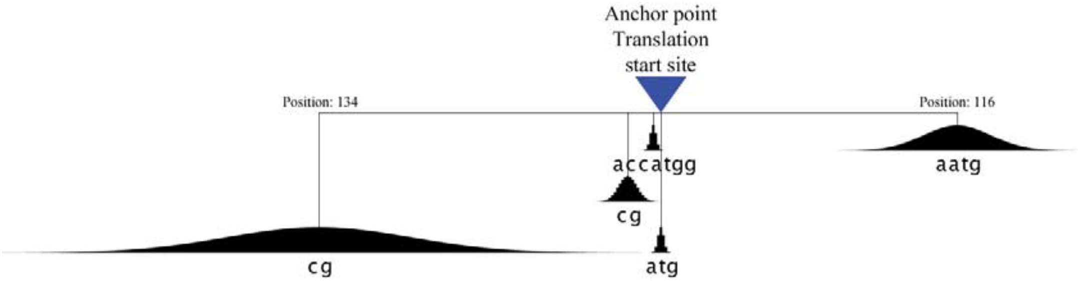


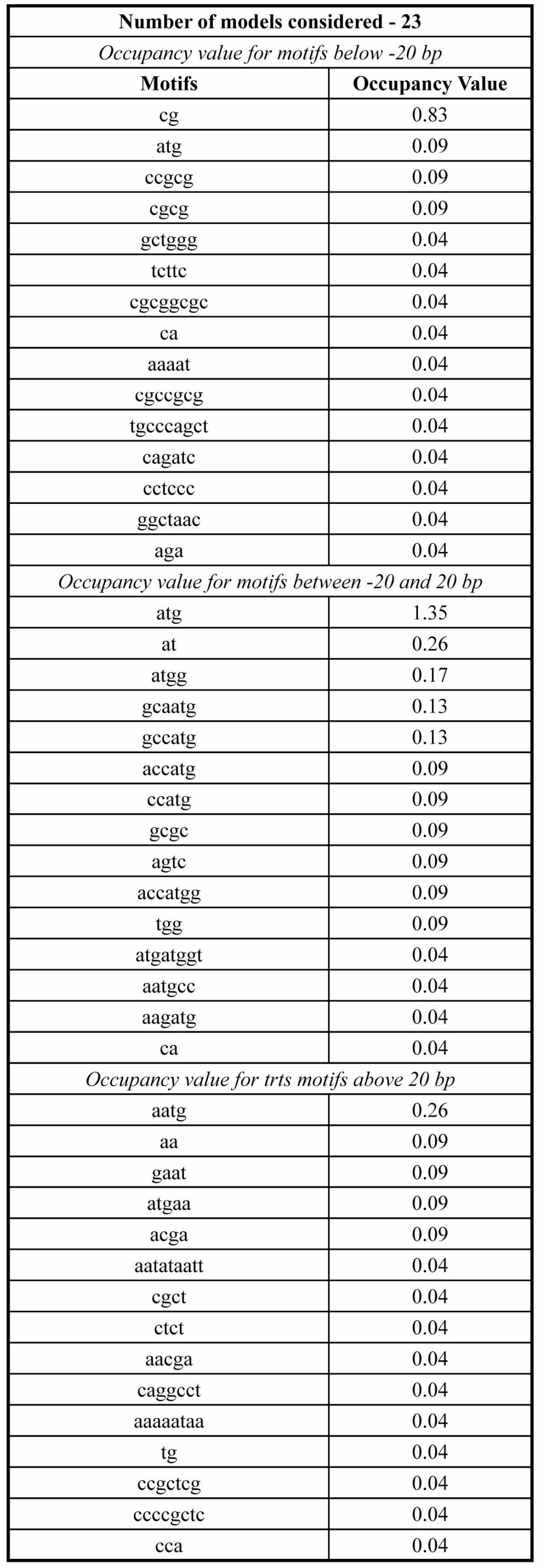
Figure 53. Translation start model trained from RefSeq cDNA sequences

Similar to the results found for genomic sequences, training Eponine models on cDNA datasets with window sizes - -20:+20, -50:+50, -75:+50, -75:+75, -100:+100, -150:+150, -170:+170 and -180:+180 did not yield better models. Models trained from window sizes less than 160 bases produced complex models and those with increased window sizes do not learn any new constraints.

A list of motifs found while training cDNA sequences along with their frequency of distribution is given in Table 8. ATG codon is represented in all the models and in few they are found more than once as indicated by the occupancy score.

The position, constraint weight and the Gaussian distribution width of the motifs represented in the above model is given in Table 9. The model obtained a strong signal for the ATG codon at position 0 with a narrow Gaussian distribution. A distribution width of 1.10 means most of the positional variation of the ATG constraint is within 3 bases from the point given in the table. This signal also has a bigger weight than the other constraints meaning the first codon is the strongest signal to determine the translation start site. Another strong constraint with a narrow Gaussian width is the *Kozak* motif positioned 3 bases upstream of the anchor point. The motif agrees with the previously reported consensus sequence (Kozak, 1987).

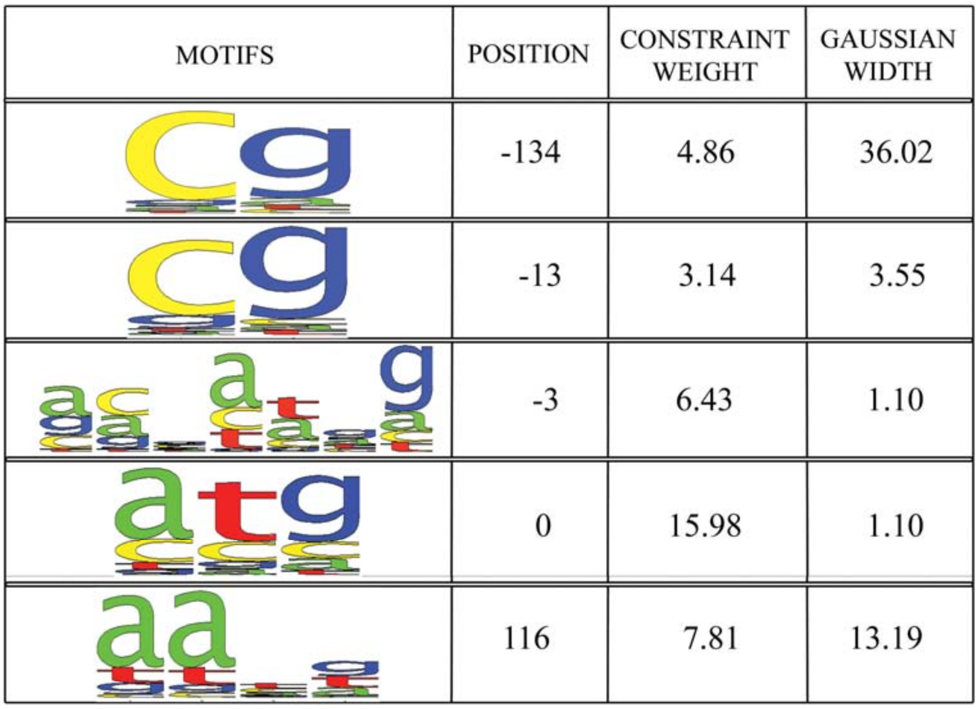
Table 8. Occupancy value for motifs detected in the translation start site models.



Number of models considered - 23	
Occupancy value for motifs below -20 bp	
Motifs	**Occupancy Value**
cg	0.83
atg	0.09
ccgcg	0.09
cgcg	0.09
gctggg	0.04
tcttc	0.04
cgcggcgc	0.04
ca	0.04
aaaat	0.04
cgccgcg	0.04
tgcccagct	0.04
cagatc	0.04
cctccc	0.04
ggctaac	0.04
aga	0.04
Occupancy value for motifs between -20 and 20 bp	
atg	1.35
at	0.26
atgg	0.17
gcaatg	0.13
gccatg	0.13
accatg	0.09
ccatg	0.09
gcgc	0.09
agtc	0.09
accatgg	0.09
tgg	0.09
atgatggt	0.04
aatgcc	0.04
aagatg	0.04
ca	0.04
Occupancy value for trts motifs above 20 bp	
aatg	0.26
aa	0.09
gaat	0.09
atgaa	0.09
acga	0.09
aatataatt	0.04
cgct	0.04
ctct	0.04
aacga	0.04
caggcct	0.04
aaaaataa	0.04
tg	0.04
ccgctcg	0.04
ccccgctc	0.04
cca	0.04

The sequence logo of the *Kozak* motif shows the interesting distribution of nucleotides at each position. The 1st and 7th position in the motif has higher distribution for A and G nucleotides respectively and this agrees with the importance previous computational methods have given for those positions in identifying translation start sites (Cavener and Ray, 1991; Hatzigeorgiou, 2002; Zeng *et al.*, 2002). The two other motifs found in the region upstream of the anchor point may capture the CG richness in the sequence between transcription start site and translation initiation site. The CG motif at position 134 bases upstream of the anchor point notably has a broad Gaussian distribution and may represent so called CpG islands, known to be associated with the 5' end of genes. An interesting constraint in the model is the AATG motif centred at 116 bases downstream of the anchor point. This motif has not been reported previously by other machine learning algorithms. It is not clear if this motif can act as another ATG codon and serve as an alternative translation start site. The Gaussian width for this motif is 13.19, meaning that most motifs would occur between 75 and 150 bases downstream of the start codon. It will be interesting to test if this constraint is involved in the leaky scanning mechanism of the translation machinery.

Table 9. Position constraints of translation start model learnt while training RefSeq cDNA sequences



MOTIFS	POSITION	CONSTRAINT WEIGHT	GAUSSIAN WIDTH
Cg	-134	4.86	36.02
cg	-13	3.14	3.55
ac a t g	-3	6.43	1.10
atg	0	15.98	1.10
aa g	116	7.81	13.19

Overall the model given in Figure 53 appears to have captured both previously known regulatory motifs and the additional interesting AATG motif positioned downstream of the anchor point. Unlike other methods the model does not rely on constraints based on the distance between the 5' end of a cDNA sequence and the ATG codon. This means the model can be used effectively on the genomic sequences as a standalone program to predict translation start sites. Also, the model should be less constrained upon the coding potential of the sequence following the start codon.

5.3.2 Translation stop model

The translation stop model was trained for nearly 5000 cycles and it took less than 1 hour in a PIII laptop. A typical model is shown in Figure 54 along with the sequence logo of motifs, position, constraint weight and Gaussian width in Table 10. Like the translation start model, the stop model is also sparse and informative. The model learnt the stop codon along with a few other sequence motifs. Two position constraints with positive weights were found upstream of the anchor point. The signal positioned at 57 bases upstream of the stop codon has relatively higher constraint weight (20.97) indicating the importance of the signal in determining the stop codon. The role of these signals and others shown with their occupancy score in Table 11 in determining the translation stop mechanism is not known.

Interestingly in this model, there is a position constraint with negative weight (-2.74) just upstream of the stop codon. This constraint – 'TTT' motif represented in blue, simply means, the motif is expected not to be present near the stop codon. However the stretch of U residues is likely to behave as positive signal downstream of the stop codon in the 3' UTR region. This can be inferred from the CCTTT motif positioned 63 bases downstream of the anchor point. Thus poly U residues are less likely to be seen in the upstream than in the downstream of a functional stop codon.

The genetic code table has 3 stop codons – UAG, UGA and UAA. However not all the 3 stop codons are used equally and most organisms have a preference for one of them. In humans, UAA stop codon is the most commonly used. The sequence logo of the TAACC motifs shows, the model has learnt all the three stop codons with preference for UAA. The A and G nucleotide in UAG and UGA stop codons are also modelled separately with a distribution of AG motif at the anchor point. The nucleotide distribution of two bases following the stop codon is almost equal and hence warrants no emphasis. However the biological implications of the two bases are not known.

Models trained with 200 bases upstream and downstream of the anchor point did not show any improvement over this model. This emphasizes the constraints near to the stop codon are more informative, making the model compact.

Table 10. Position constraints of translation stop model learnt while training RefSeq cDNA sequences

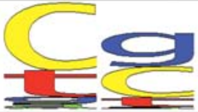
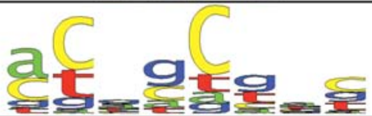
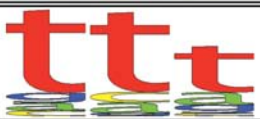
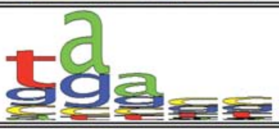
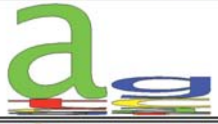
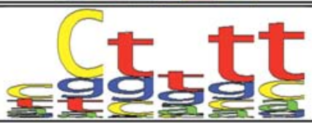
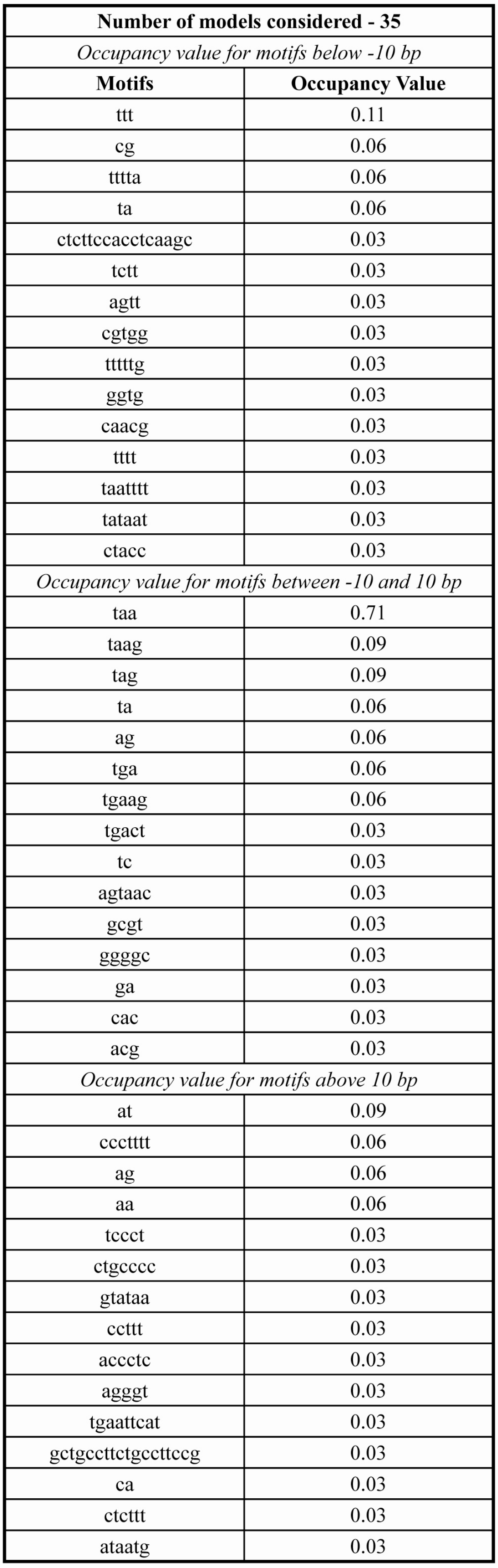
MOTIFS	POSITION	CONSTRAINT WEIGHT	GAUSSIAN WIDTH
	-84	5.15	4.01
	-57	20.97	4.21
	-17	-2.74	5.09
	-2	21.42	2.91
	0	6.95	2.91
	63	12.74	7.66

Table 11. Occupancy value for motifs detected in the translation stop site models.



Number of models considered - 35	
Occupancy value for motifs below -10 bp	
Motifs	**Occupancy Value**
ttt	0.11
cg	0.06
tttta	0.06
ta	0.06
ctcttccacctcaagc	0.03
tctt	0.03
agtt	0.03
cgtgg	0.03
tttttg	0.03
ggtg	0.03
caacg	0.03
tttt	0.03
taatttt	0.03
tataat	0.03
ctacc	0.03
Occupancy value for motifs between -10 and 10 bp	
taa	0.71
taag	0.09
tag	0.09
ta	0.06
ag	0.06
tga	0.06
tgaag	0.06
tgact	0.03
tc	0.03
agtaac	0.03
gcgt	0.03
ggggc	0.03
ga	0.03
cac	0.03
acg	0.03
Occupancy value for motifs above 10 bp	
at	0.09
ccctttt	0.06
ag	0.06
aa	0.06
tccct	0.03
ctgcccc	0.03
gtataa	0.03
ccttt	0.03
accctc	0.03
agggt	0.03
tgaattcat	0.03
gctgccttctgccttccg	0.03
ca	0.03
ctcttt	0.03
ataatg	0.03

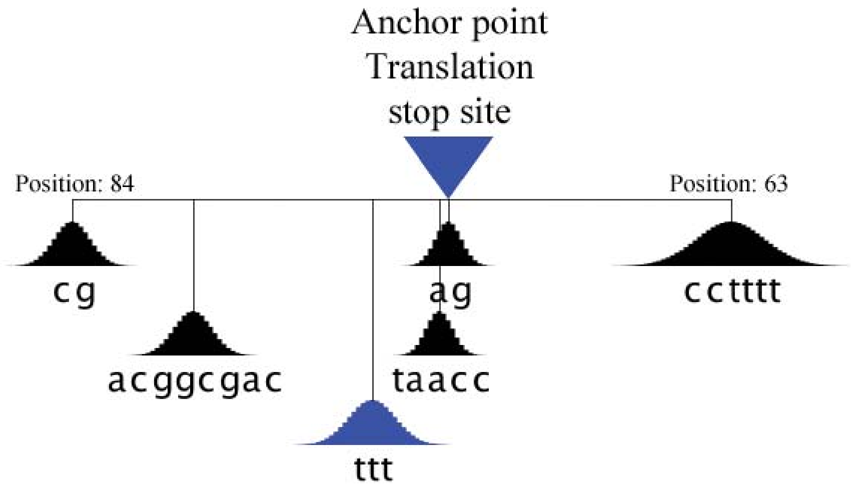


Figure 54. Translation stop model trained from chromosome 22 cDNA sequences

5.4 Validating and testing the models

I tested the performance of both the translation start and stop models in different datasets as explained below –

For quantification purposes, I defined a prediction as accurate if it is positioned within 200 bases upstream or downstream of the annotated start codon. Accuracy is calculated as the number of annotated start sites predicted over total number of predictions. Whereas coverage is number of annotated start sites predicted over total number of annotated start sites. Initial testing of the models was done on the set of sequences set apart while training. As explained before during each run, 226 (113 positive + 113 negative) sequences were kept apart from the trainer to be unseen while training. These 226 sequences were randomly picked up from the positive and negative set and thus vary for each run. So these test sets are fairly representative of the sequences available in the database and the model was tested on it. The performance of the model on this set was found to have good coverage and accuracy. However in this case, testing was done by scanning only the few bases around the anchor point to determine whether the sequence is a positive or negative. Hence, I used three independent test sets to analyse the performance of the models by allowing them to scan the whole cDNA sequence.

I took human reviewed RefSeq human and mouse and Riken mouse cDNA with at least 200 bases upstream to test the models. RefSeq database has cDNAs with different levels of annotation and thus they are not of equal degree. Among the different levels, manually reviewed cDNA are of high quality and I limited my test sets to these sequences alone. The 113 sequences used here were human cDNAs of this quality with at least 200 bases upstream. Predictions are made by scanning the sequence moving from left to right and evaluating the probability of the fit of the sequence motifs in the model in the cDNA sequence. The model found 3169 predictions (same strand) covering 87% (99 start codons out of 113, at threshold of 0.99) of annotated translation initiation sites. Figure 55 shows the ROC curve for the translation start model for predictions in the same (prediction in the same direction as of the gene), opposite (prediction in the reverse direction compared to annotation) and both (strand details are ignored) strands. Although the model predicted the strand of the annotated site correctly in most cases, few predictions are found in the opposite strand compared to the start codon. Combining predictions in both strands shows better coverage and accuracy than strand specific predictions.

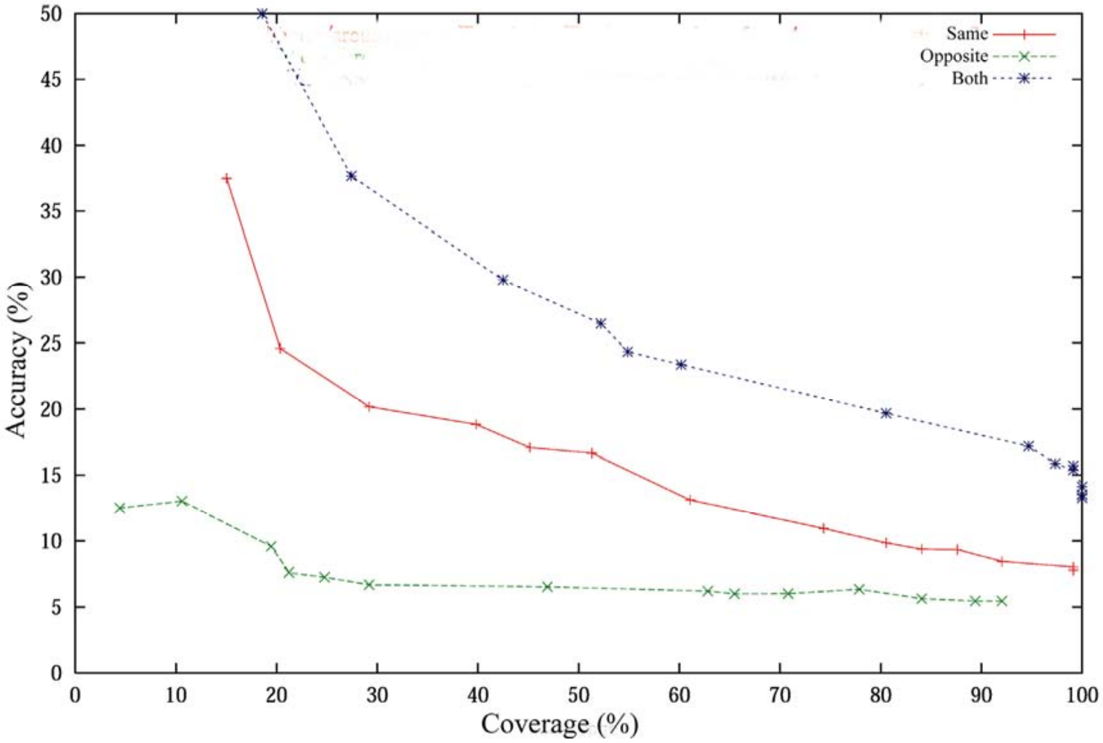


Figure 55. ROC curve on human RefSeq cDNA dataset for Eponine translation start site predictions in same, opposite and both strands

A set of similar quality entries from RefSeq was extracted for mouse cDNAs. This set resulted only in 37 sequences. The translation start model predicted 65% of the annotated start codons with 1537 predictions in total at a relatively less stringent threshold score of 0.95. The low coverage may be due to the requirement of 200 bases upstream of the translation start site by the model for scanning and only few sequences in the dataset met this criterion. Also some of the predictions might be true start sites and annotations are not available at present to validate them. In cases where the annotated sites are identified correctly, the positions of the predictions are limited to -2 to +2 bases from the annotated site. An ROC plot of the performance of model for this dataset is given in Figure 56a and Figure 56b for same and both strand predictions respectively.

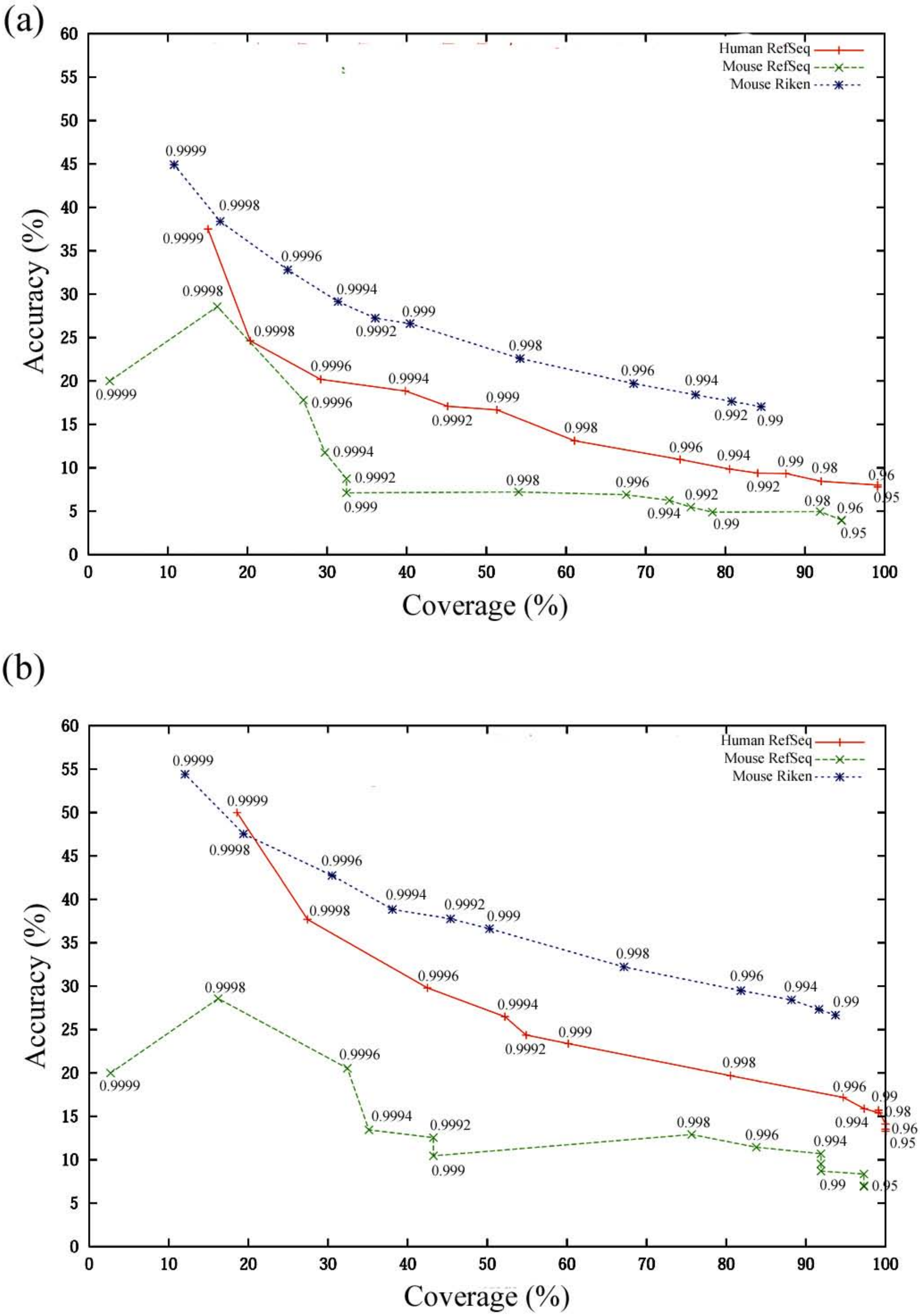


Figure 56. ROC curve for translation start model on human RefSeq, mouse RefSeq and mouse Riken cDNA datasets (a) Predictions in the same strand (b) Predictions in both strands

I extracted another set of mouse cDNA sequences of comparable quality from the RIKEN database. This set had 1593 sequences with each sequence having at least 200 bases upstream of the annotated start codon. The scanning of all these sequences using the model shown in Figure 53 gave 20400 predictions with a threshold value of 0.992. The predictions (same strand) covered 1287 translation initiation sites (80% coverage). ROC plots calculated from predictions in the same and both strands are given in Figure 56a and Figure 56b respectively.

Like the start model, the translation stop model was first tested on the test sequences set apart while training. As the test sequences are randomly selected from positive and negative sequences and they are different with each run of training, the trainer has less chance to 'overfit' the positive dataset.

Accuracy and coverage was calculated as explained above with 200 base tolerance in the prediction position relative to the annotated stop codon. Figure 57 shows the ROC curve for translation stop sites in human RefSeq dataset (113 sequences). The performance of the stop model is worse compared to the start model.

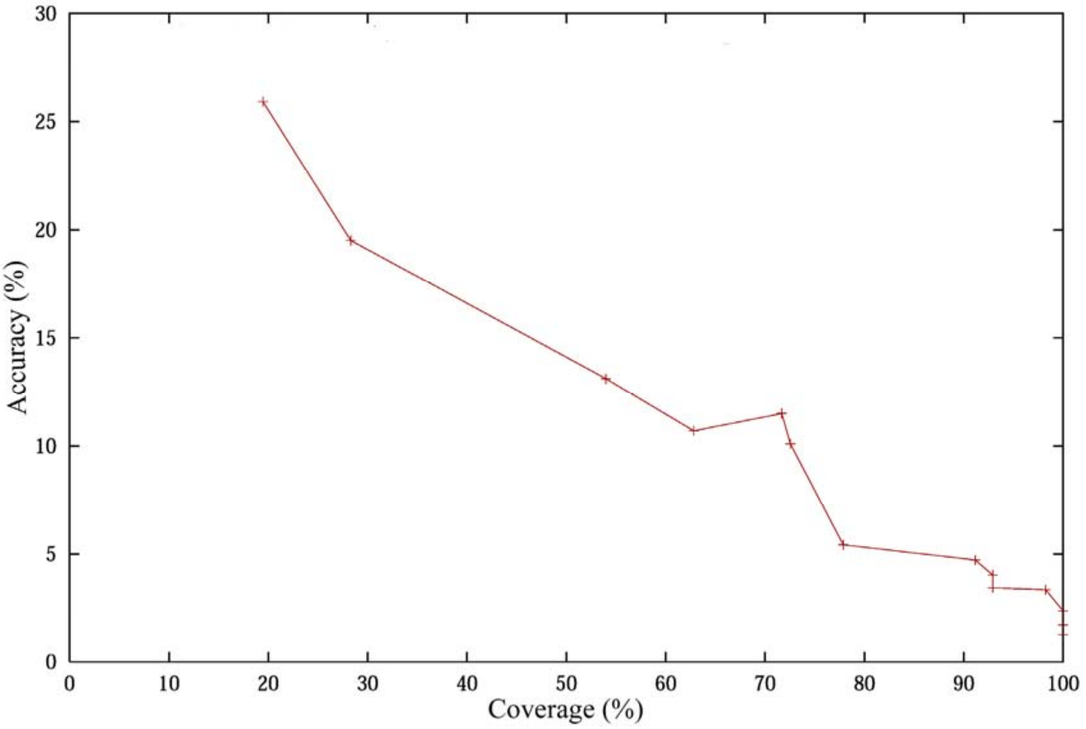


Figure 57. ROC curve on translation stop sites in human RefSeq cDNAs for Eponine model

5.5 Position accuracy of the models

5.5.1 Translation start model

As well as predicting most of the annotated start codon, the translation start model showed reasonable performance in determining the exact position of the start codon. The model is anchored on the A in ATG codon and any point in the sequence predicted by the model correlates with this nucleotide. I calculated the density of the predictions relative to the start codon and plotted the histogram for human RefSeq and mouse RIKEN datasets (Figure 58).

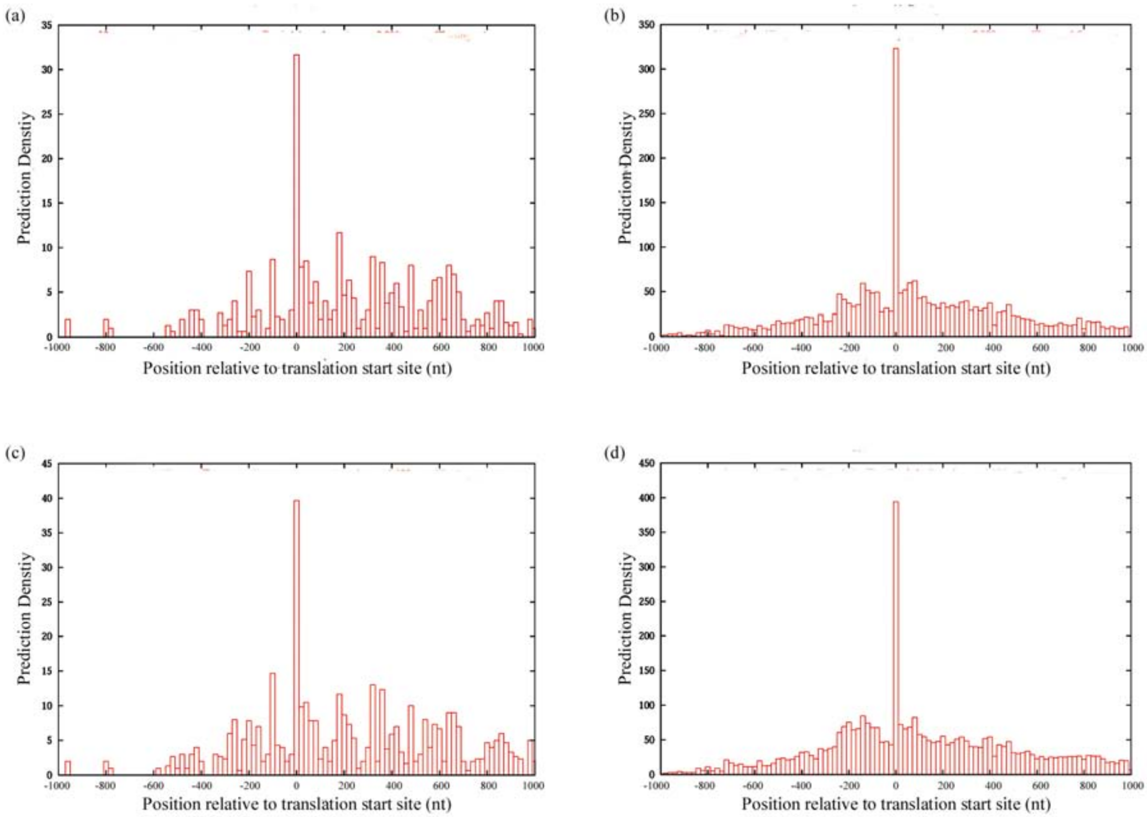


Figure 58. Prediction densities for translation start site model relative to annotated initiation sites (a), (b) Density of predictions in the same strand for human RefSeq and mouse RIKEN data respectively (c), (d) Density of predictions in both strands for human RefSeq and mouse RIKEN data respectively

The results show a clear peak, with many of the predictions centred within 20 bases from the annotated start codon. In some cases, the prediction positions are highly accurate and anchored at +1/-1 bases relative to the initiation site. However the majority of predictions are between -5 and +7 nucleotides relative to the anchor point. Figure 58 shows most of the predictions are in the same strand as of the annotation although few predictions lie in the opposite direction. Even in cases, where the predictions are in the opposite strand, the predictions are concentrated within 20 bases from the start codon.

5.5.2 Translation stop model

The density of predictions made by this model on the human RefSeq set of sequences is shown in Figure 59. This model cannot predict the position of the stop codon correctly and there is no significant peak near annotated sites. I was surprised to note this result, as the model seems sparse and informative and captured known consensus signals. Further analysis might yield better results in predicting translation stop sites. Nevertheless, the results here simply indicate that the end of translation is determined solely by the stop codon itself and not by any motifs in the surrounding sequence.

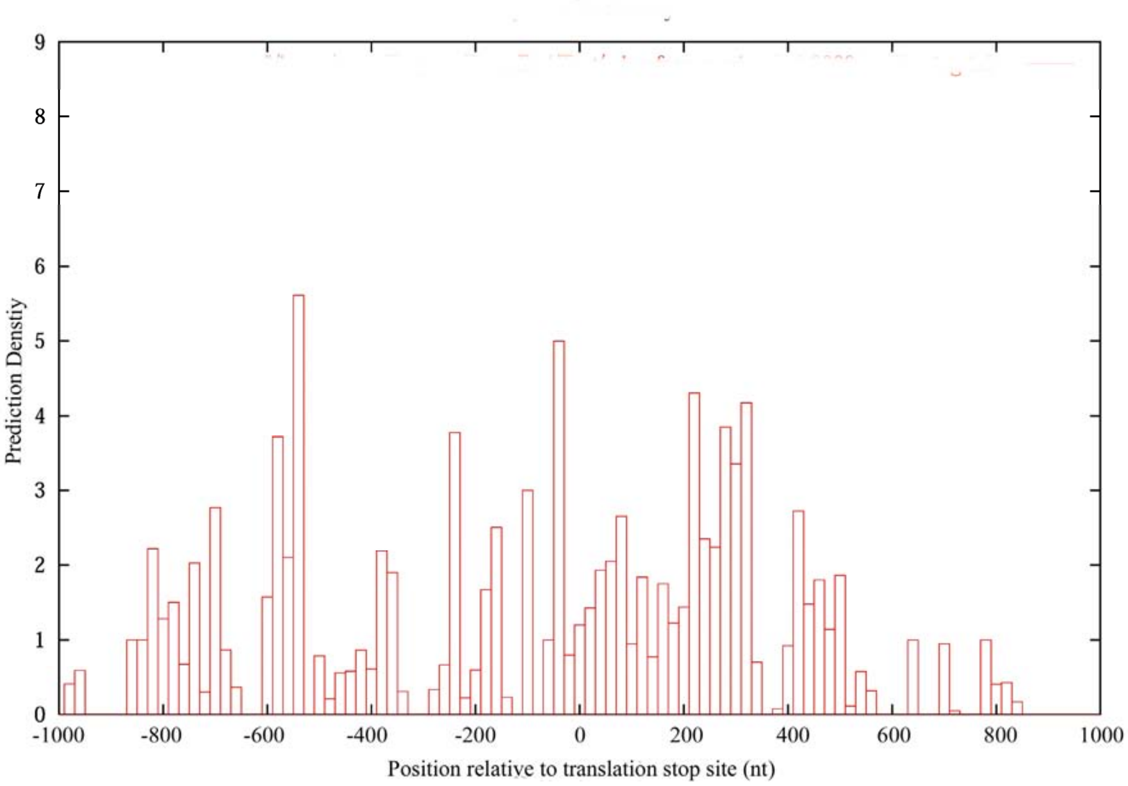


Figure 59. Prediction density for translation stop model relative to annotated stop sites

5.6 Comparison with other models

I compared the performance of Eponine predictions with two other translation initiation prediction programs, NetStart and ATGpr available in the public domain.

NetStart (Pedersen and Nielsen, 1997b) – This program was created as an improvement over using weight matrices to determine translation initiation sites. The program uses Artificial Neural Network (ANN) and was trained on 100 bases upstream and downstream of the start codon. The information surrounding the AUG codon was used primarily for prediction.

ATGpr (Salamov *et al.,* 1998a) – This program along with sequence context used six other characteristics to identify putative start sites. These characteristics are –

(a) Positional weight matrix around an ATG.

(b) Hexanucleotide difference between sequences upstream and downstream of the ATG codon.

(c) Preference for longer reading frames downstream of ATG

(d) Signal peptide characteristic

(e) Presence of another upstream in-frame ATG

(f) Upstream cytosine nucleotide characteristic

Linear discriminate analysis was used to generate a single score from the combination of these properties.

As neither of the programs is available for download, I used their web interfaces to scan human RefSeq test sequences. The NetStart (Pedersen and Nielsen, 1997a) web interface has a restrictions on the number of sequences submitted to the server (at most 50 sequences) and hence I split the dataset (113 sequences) into 3 sets and scanned them separately. The results from the three sets are then combined together for comparison. I used default parameters for vertebrate sequence given in the web-based predictor.

The ATGpr web interface (Salamov *et al.,* 1998b) has even more severe restrictions and it cannot take any sequence longer than 1300 bases and hence I split each sequence into 1150 base chunks with overlapping window size of 10 bases. These chunks were submitted to the server and predictions for a cDNA sequence were obtained by merging the predictions of each chunk. Default parameters for human sequence were used for prediction.

Figure 60 shows the performance of Eponine model compared to these two programs. The ROC curve shows, Eponine performs better than NetStart although less well than ATGpr. This was expected, as ATGpr uses additional information apart from regulatory elements to screen out false positives. NetStart which uses only sequence elements performs less well than Eponine.

5.7 Concluding remarks

Machine learning techniques assume the ribosomes operate in a linear fashion. NetStart developed by Pedersen and Nielsen (Pedersen and Nielsen, 1997a, b) based on a ANN was trained on a 203 nucleotide window centred on the AUG codon. The same dataset was used to train a Support Vector Machine model by Zien *et al.* and an improvement was obtained by using a kernel function to detect the codon bias in the downstream sequence of AUG. Likewise, Salzberg used a conditional positional probability kernel function to improve the ANN model using SVMs (Salzberg, 1997). More recently, Hatzigeorgiou reported a prediction program called DIANA-TIS based on a ANN trained on human sequences. This program combined a consensus ANN with a coding ANN together with the ribosome scanning model. Zeng *et al.*. used similar techniques by combining various informative features generated by different machine learning techniques. They found the following features useful: -3 and -1 position in the sequence relative to AUG; upstream k-grams for k = 3, 4 and 5; stop-codon frequency; downstream in-frame 3-gram; and the distance of AUG to the beginning of the sequence. The k-grams count the frequency of occurrence of a particular pattern in a window of length k that slides upstream and downstream of the AUG codon. Downstream in-frame 3 gram gives measure of the coding potential of the downstream sequences.

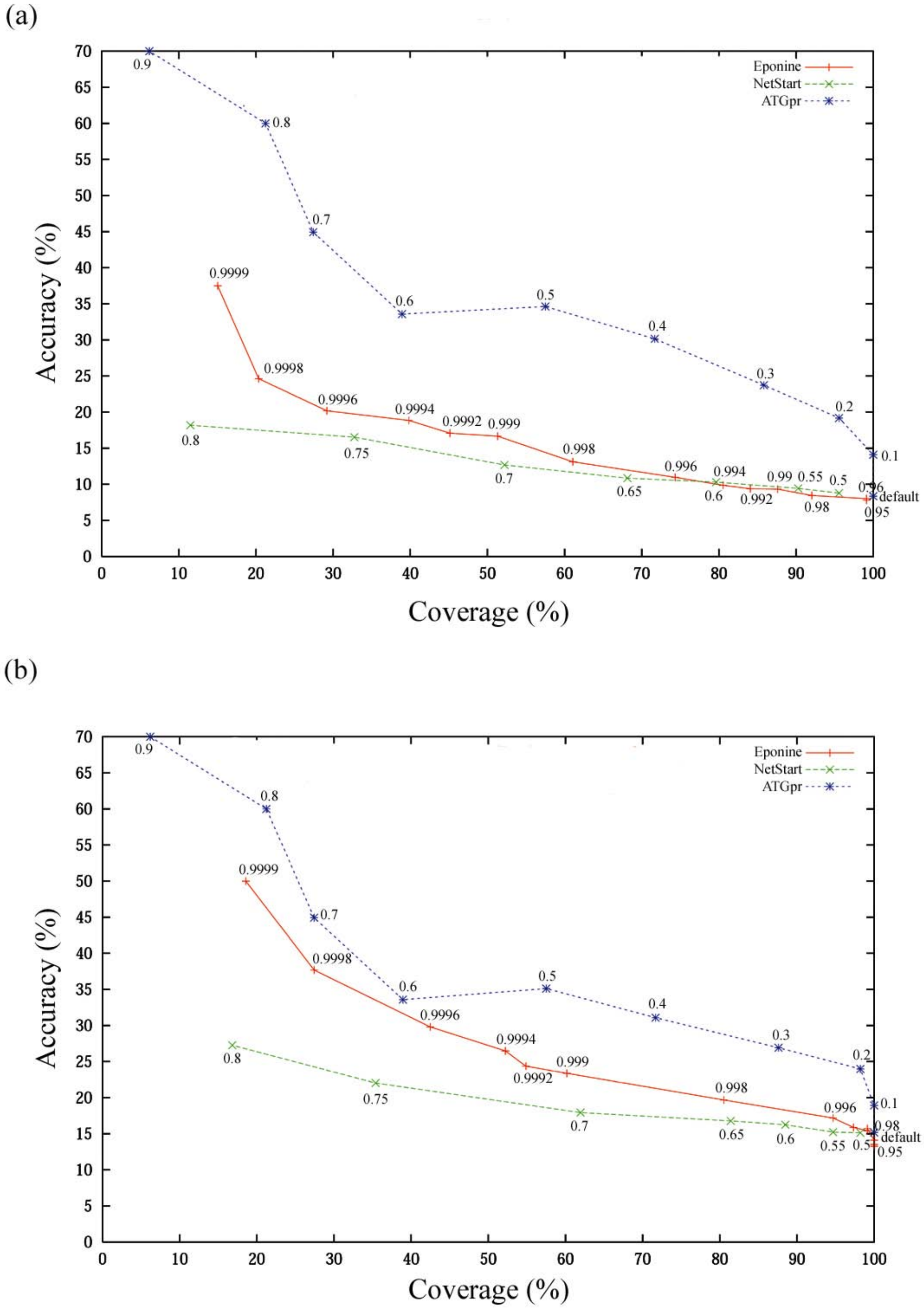


Figure 60. ROC curves for Eponine, NetStart and ATGpr on human translation start sites in RefSeq cDNA sequences without (a) and with (b) strand information

Thus these programs use a significant amount of ‘content’ information from the cDNA sequences to predict translation start codons. Here I attempted to make a translation stop and stop model that can scan genomic sequences and predict start and stop codons respectively purely based on regulatory signals. Despite using only signal information, the Eponine translation start model performed better than NetStart. The positional accuracy of the start model in cDNA sequences is good and few of the predictions are in the opposite stand relative to the annotated site.

With transcription, splicing and translation models learnt so far I show the advantage of making an *ab initio* gene prediction program combining them using GAZE in the next chapter.

GENEPRED – AN *AB INITIO* GENE PREDICTOR

6.1 Introduction

With the availability of models for transcription, splice site and translation, here I introduce an *ab initio* gene prediction system, *GenePred*, created using the regulatory signals identified by Eponine. As explained previously, almost all gene prediction programs use 'content' information, such as codon bias and ORF length. The gene prediction system explained in this chapter is different in this respect as the system uses only 'signal' information. Such a gene prediction system has an advantage over existing gene prediction algorithms in that it has the potential to identify non protein coding RNAs as well as coding RNAs. Recent analyses (Cawley *et al.*, 2004; Mattick, 2001) indicate that a huge amount of non coding transcription occurs within the cell and most of these RNAs are regulated in a similar way to protein coding genes. Various functions are attributed to these RNAs such as RNA interference, co-suppression, transgene silencing, imprinting and methylation. Few attempts (di Bernardo *et al.*, 2003; Rivas and Eddy, 2001; Rivas *et al.*, 2001) have been made to identify these RNAs computationally and so far with only limited success. A gene prediction program based on 'signal' information alone, and thus not biased due to 'content' information, should more closely mimic the biological system than existing gene prediction methods, as the *in vivo* transcriptional machinery does not use 'content' information while transcribing a genomic region. Content information has historically been used to assist computational detection of genes since signal based prediction alone has been insufficient (Guigo, 1997).

The gene prediction model explained in this chapter was constructed using a dynamic programming framework called GAZE (Howe *et al.*, 2002), which can combine features identified by predictive models, such as those described in the previous chapters. GAZE allows evidence for individual gene components to be assembled in order to predict entire gene structures. As explained in chapter 2, the method uses a dynamic programming algorithm to obtain (i) the highest scoring gene structure with the supplied features and (ii) posterior probabilities that each input feature is part of a gene.

In this chapter I explain the details of the features and gene models used in deriving various versions of the gene prediction system. Following this, I compare the performance of the system with the well established gene prediction program called GENSCAN (Burge and Karlin, 1997), as this program is assessed to be one of the best *ab initio* programs available in the public domain (Guigo *et al.*, 2000; Parra *et al.*, 2003). Towards the end of this chapter I revisit the performance of the transcription termination model given the context of splice site model predictions.

6.2 GAZE gene structure models

Many gene prediction programs have two common features –

(i) signal and content measures are used to detect components and regions belonging to genes
(ii) assemblage of these components into complete gene structure prediction for the sequence and scored against some measures

For the first of these steps, different measures, say weight matrices, codon bias, pentamer and hexamer frequencies and splice site predictions can be used to distinguish the components of gene structure from the sequence. For the second of these steps, a choice must be made as to the *model* of gene structure over which the assembly is to be performed. One of the advantages of GAZE is that it decouples these two steps of assembly of signal and content data into gene structure predictions from the generation of the data itself. The inputs for both these steps are provided externally and GAZE does not work directly with genomic DNA. In this project, for the first step, I used Eponine predictions as signal features, which I explain in the next section. For the second step, I used the following models (Figure 61) to validate the assembled components of the gene signal features.

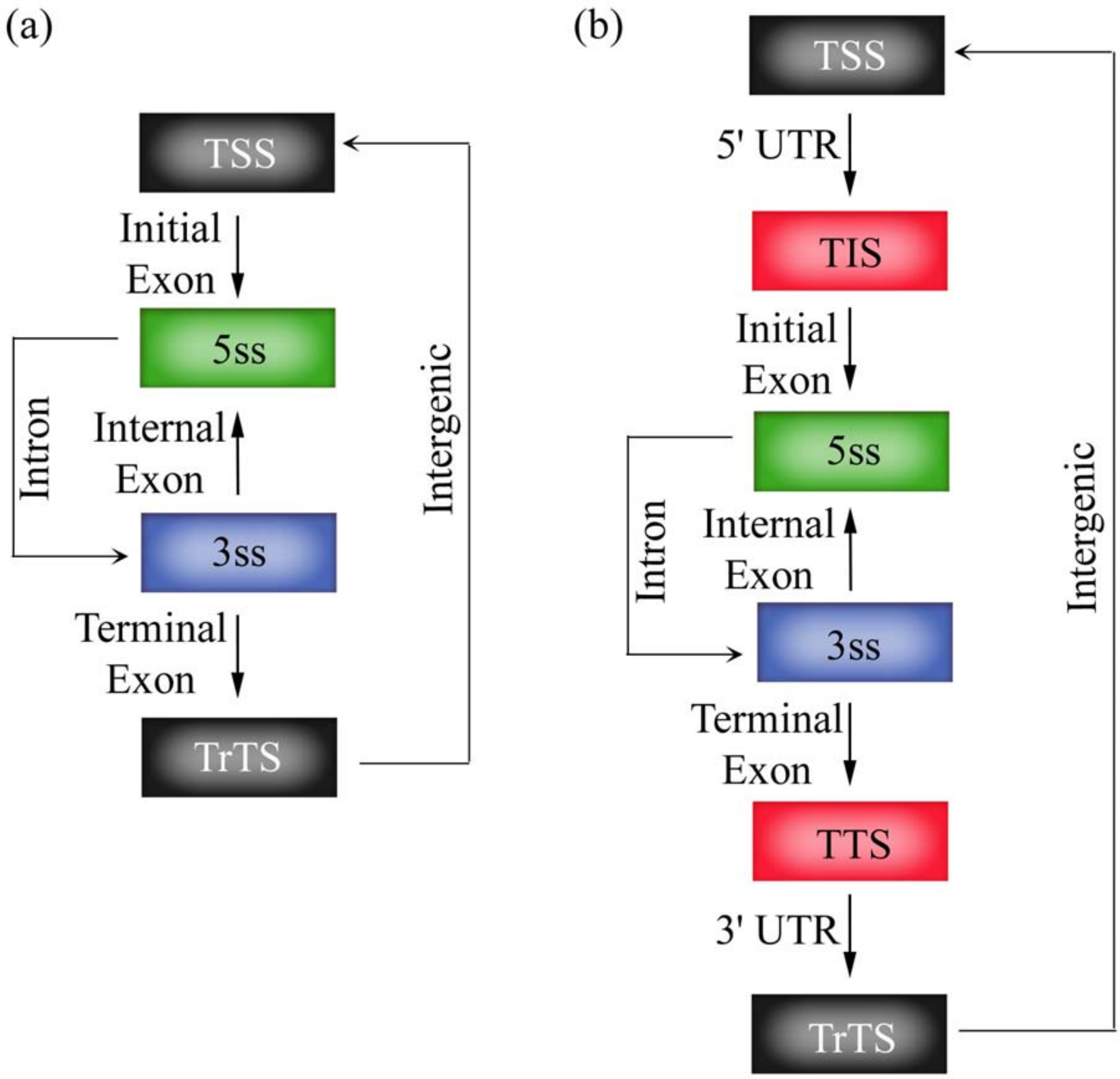


Figure 61. Schematic representation of the gene models used for predicting genes from features in the forward strand. Reverse complementation of the forward strand rules are used for reverse strand gene predictions. (a) Simple gene model without translation models and thus no protein information. (b) Gene model with translation features. Any introns within 5' UTR region are not modeled. Based on these gene structures, candidate genes are predicted on both strands at the same time.

In Appendix C, I have given the configuration files where the gene structure models used are presented in GAZE-XML format. A pictorial representation of these gene structures is given in Figure 61. The configuration file has five sections –

(i) *declarations* – declares the Eponine features that GAZE is going to work with

(ii) *gff2gaze* – dictates how the input files are used to obtain a list of features

(iii) *dna2gaze* – allows for the creation of features from simple sequence motifs observed in the input DNA sequence

(iv) *model* – contains the gene structure rules

(v) *lengthfunctions* – this section describes the length penalties used in defining exons, introns and intergenic regions in the model

The gene structure rule I used here (Figure 61a) is simple and it starts with a transcription start site followed by the donor site. The region between these two predictions defines the *initial exon* segment. The *introns* that interrupt the coding region of the gene are modelled by allowing a transition from donor to acceptor site. Introns might occur between two codons or in the middle of a codon, either between first and second position or between second and third positions. However, since the aim is not to consider any coding information in constructing the gene model, the phase associated with intron interruption is not considered. The donor and acceptor site features are represented as *5ss* and *3ss*. The sequences between a *3ss* and a *5ss* feature forms the *internal exon* of the gene structure. The *terminal exon* is defined as that part of the sequences between an acceptor site and a transcription termination site. Transcription termination site defines the end of the candidate gene. Thus a gene structure is defined with the features from transcription and splice site signals. To form the next gene another list of features are sampled and analysed to fit the rules explained above. That part of the sequence between two genes defined between transcription termination and start features is referred to as *intergenic*. To predict genes in the reverse strand, reverse complementation of the above rules are employed. Single exon genes are not modeled in this case. This is due to the fact that a simple single exon gene model will use only transcription start and termination site without any splice site model predictions. Allowing this simple single exon gene transition will bias the gene structures to terminate just after the start site because of the unusual presence of termination signals near transcription initiation site (refer to chapter 3).

Thus, this gene model without translation components more realistically mimics the biological transcriptome and spliceosome machinery that transcribes the DNA and processes the newly synthesized RNA respectively.

All the features are derived from Eponine models for predicting genes and I did not use the *dna2gaze* section to create a set of features from the DNA sequences. Similarly no constraints on the maximum length of exons and introns are placed in the gene model and thus no length penalty functions are used.

Figure 61b shows a pictorial representation of a different gene model used in predicting genes. In this structure, I used Eponine translation start and stop models as well. The transcription start site is now allowed to transit to the translation start codon, thus defining a new segment called the *5' UTR*. The region between the translation start codon and donor site now defines the *initial exon* segment. Similarly, the translation stop model is incorporated after the acceptor site of the last exon before transition to the transcription termination site. This change will make the GenePred system emit the *3' UTR* segment. By adding translation models and thus start and stop codon signal information, some protein coding information is attached to the gene prediction system. This is done to analyse the influence of the translation models in the GenePred system.

6.3 Eponine prediction models

As explained earlier, given a candidate set of gene features, GAZE predicts genes by deriving a subset of features that according to the given gene structure is the most likely candidate. The gene structure scoring the highest value with the list of features is predicted as a candidate gene. In order to provide the list of features to GAZE, I used Eponine model predictions. The following models are used along with their respective thresholds (given in brackets) to obtain predictions from signals in the DNA sequences.

(i) Transcription Start Site model (0.99)
(ii) Translation Start model (0.99)
(iii) Donor Site model (0.999)
(iv) Acceptor Site model (0.9998)
(v) Translation Termination model (0.999)
(vi) Transcription Termination model (0.99)

Apart from Eponine models, I also used GeneSplicer predictions while testing the performance of the GenePred system. GeneSplicer was used with default options to predict splice site features from the DNA sequence.

All the predictions were dumped in the General Feature Format (GFF, WTSI), a widely used standard for the exchange of gene prediction information.

Here I used chromosome 20 for scanning features and predicting genes as all the Eponine models discussed in previous chapters are trained from chromosome 22.

6.4 Gene prediction with Eponine features

With the availability of features from chromosome 20, I combined them to create a gene prediction system by inputting the features and the gene model structure (Figure 61a) into GAZE.

Figure 62 and Figure 63 show the genes predicted using GenePred as red tracks (the first red track in Figure 62 and the last red track in Figure 63) for a 1 mega base region (57.35 to 58.35 bases) of chromosome 20. For ease of comparison, in Figure 64, I removed all the annotation tracks and kept only VEGA, ENSEMBL annotations and GENSCAN (Burge and Karlin, 1997) predictions.

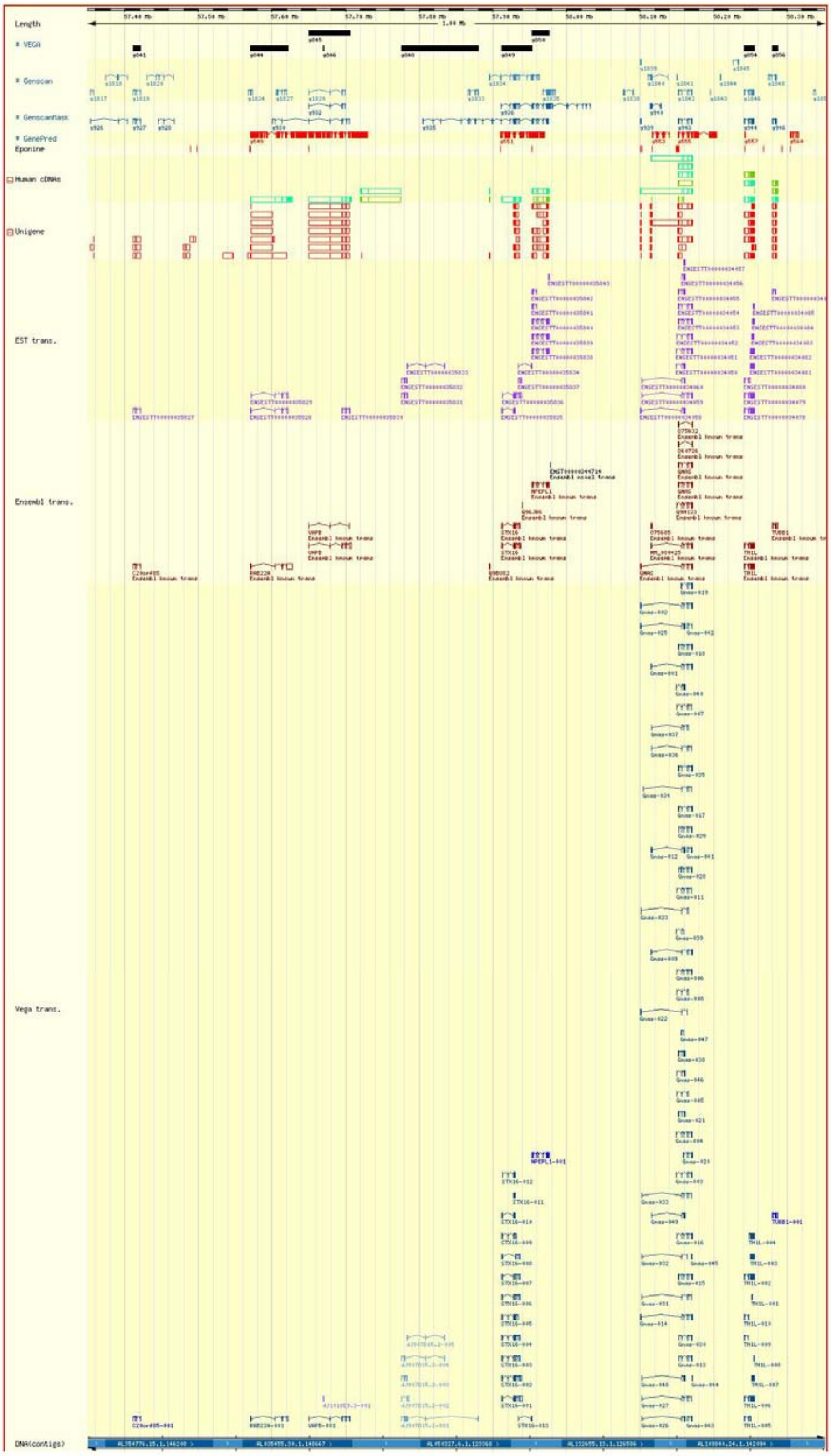


Figure 62. Genes predicted by linking Eponine models using GenePred compared with annotations available in the forward stand. Annotations from VEGA, ENSEMBL, EST transcripts, UNIGENE and Human cDNAs are shown as tracks along with GENSCAN predictions (both on masked and unmasked sequence). The comparison is possible with the ENSEMBL ContigView which can load predictions from external source as DAS tracks.

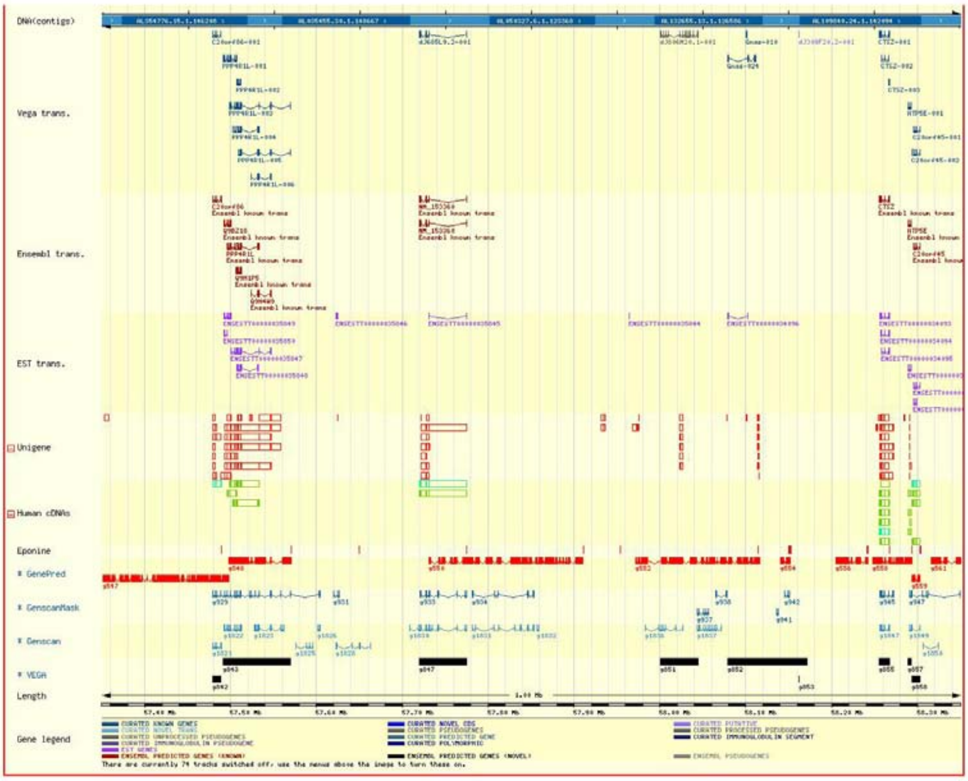


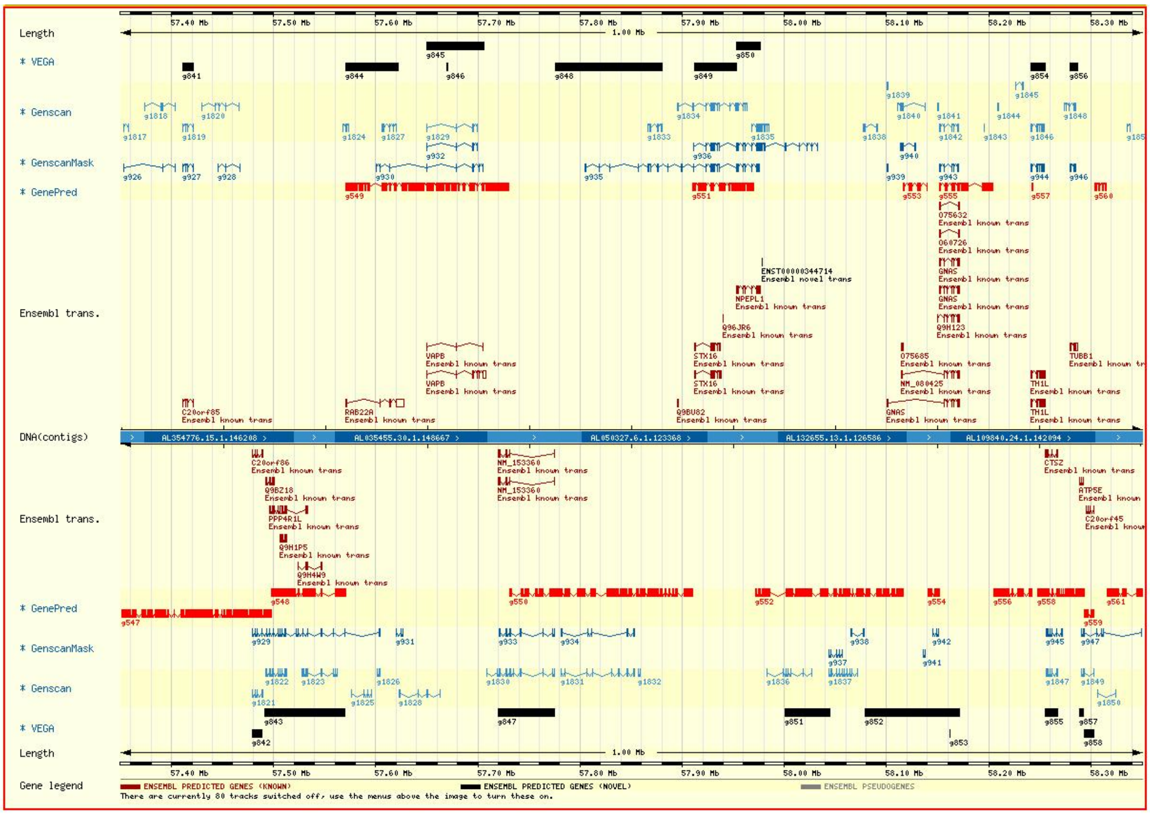
Figure 63. Genes predicted by linking Eponine models using GenePred compared with annotations available in the reverse stand. Annotations from VEGA, ENSEMBL, EST transcripts, UNIGENE and Human cDNAs are shown as tracks along with GENSCAN predictions (both on masked and unmasked sequence). This figure is reproduced from ENSEMBL ContigView viewer.

GENSCAN predictions are derived by scanning repeat masked chromosome 20 sequence. This is done by splitting the chromosome sequence into 200 kb overlapping blocks and GENSCAN predictions on each block are then merged together using a merging algorithm (Hubbard, T., personal communication) to derive the final list of predictions.

I compared the performance of GenePred with that of GENSCAN using the following definition of coverage and accuracy –

(i) Coverage is defined as the number of genes identified over the total number of annotated genes.

(ii) Accuracy is calculated as the number of predictions matching the annotation over the total number of predictions. Predictions that fuse or split the gene are considered as false positives (Figure 65). This included a few predictions matching genes that have an internal gene in the same strand.



There are currently 80 tracks switched off, use the menus above the image to turn these on.

Figure 64. Genes predicted by linking Eponine models using GenePred compared with annotations available in both strands. VEGA annotations are shown as black bars. The region covered by a bar includes all the alternative transcripts of a gene. GenePred predictions are given in red color. The figure also shows GENSCAN predictions and ENSEMBL annotations in different tracks.

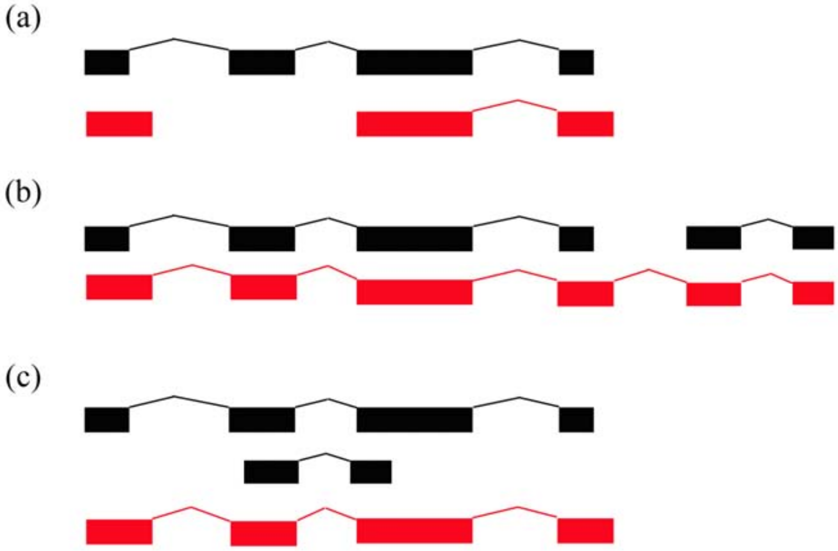
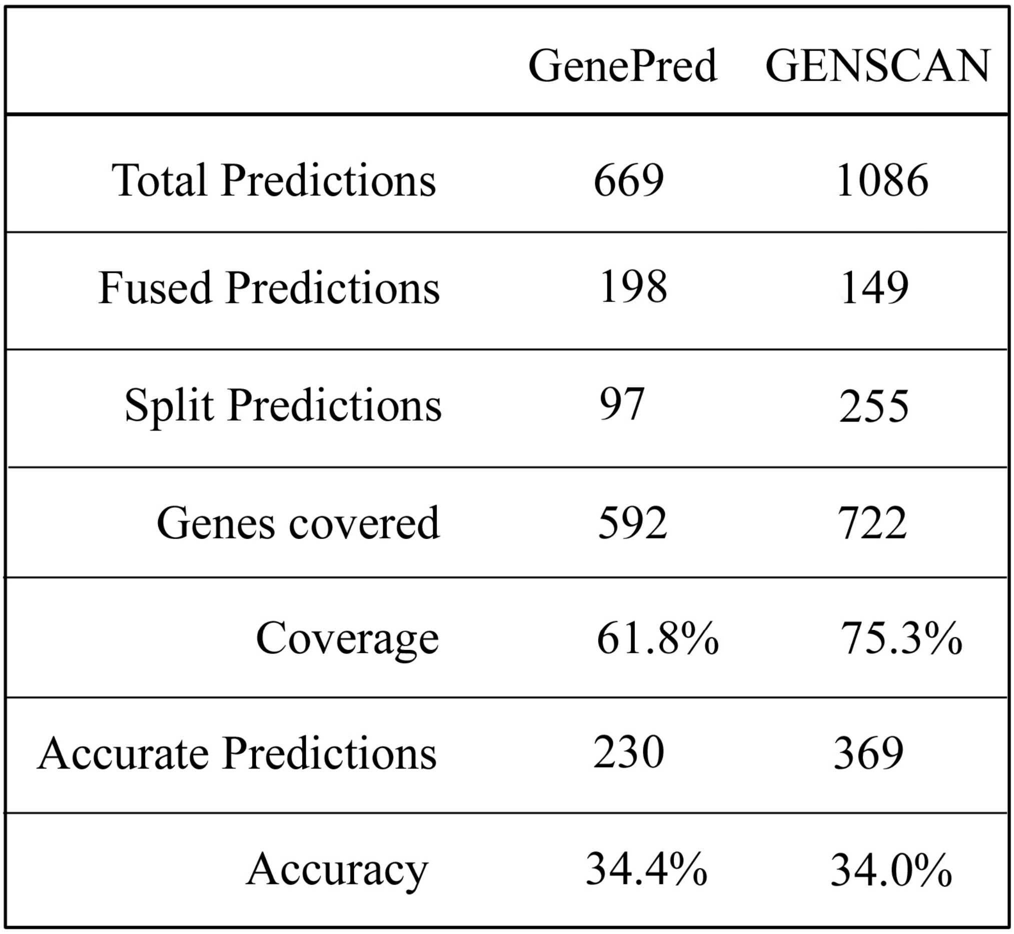


Figure 65. Pictorial representation of (a) split and (b) fused predictions in comparison with annotation. (c) Few annotated genes have internal genes in the same strand. Predictions matching these genes are ignored while calculating accuracy. Annotations are given in black while predictions are drawn in red.

I extracted annotations from the VEGA database (Ashurst, 2002) and found that chromosome 20 had 959 annotated genes (includes, Known, Novel CDS, Novel Transcripts, Pseudogene, Processed pseudogene, Unprocessed pseudogene and Putative categories). GenePred predicted 669 genes while GENSCAN made 1086 predictions after scanning the chromosome 20 sequence (Table 12). GenePred covered 592 genes (61.8%) of the total annotated genes while GENSCAN coverage was roughly 13% higher, identifying 722 genes (75.3%). Accuracy of GenePred and GENSCAN was found to be similar. GenePred made 230 correct predictions (34.4%) while GENSCAN predicted 369 (34.0%). However, GENSCAN made relatively higher number of split predictions (255 predictions). In contrast, GenePred made relatively more fused predictions (198 predictions compared to 149 by GENSCAN) and a smaller number of split predictions (97 predictions). However, GenePred had difficulty in identifying the annotated exon and intron boundaries when compared to GENSCAN (Table 13). Any prediction that overlaps an annotated exon is included in calculating coverage and accuracy. However, split and merge predictions are counted as false positives while deriving accuracy. Out of 6441 exons annotated by VEGA, GenePred predictions overlapped with 2869 (44.5%) while GENSCAN predicted 4132 (64.2%). GENSCAN’s coverage is achieved from fewer predictions than GenePred and hence accuracy of GENSCAN (46.3%) is significantly higher than GenePred (12.7%). GenePred is not suited for predicting exact exon-intron boundaries (5512 annotated splice sites) as the donor and acceptor site coverage and accuracy is significantly less than GENSCAN (refer to Table 13). These results are expected as GenePred does not use any ‘content’ information like other *ab initio* gene prediction systems. Thus, GenePred is good for identifying gene blocks in the DNA sequences, which could be later annotated for exon-intron structure using other algorithms. The high number of fused predictions by GenePred indicates the potential to improve the model by tweaking the parameters and the feature models used to predict genes.

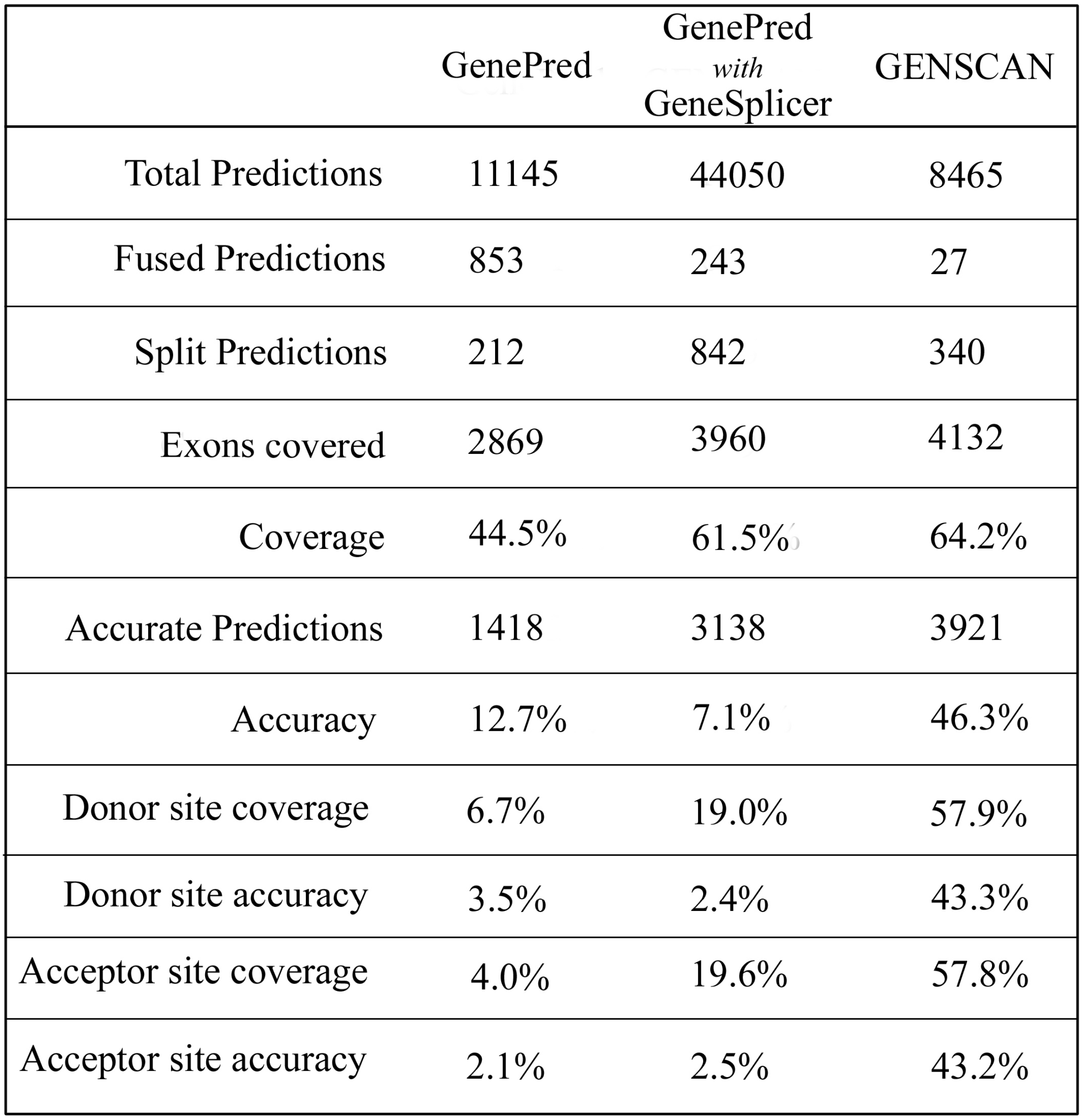
For the above comparison, I used GENSCAN predictions on repeat masked sequence since GENSCAN was known to perform better in masked than unmasked sequence. GENSCAN predictions on unmasked chromosome 20 (2108 predictions) shows significantly less accuracy (19.7%, compared to 34.0% reported earlier) although the coverage remains similar (masked: 75.3%, unmasked: 77.3%). This might be due to the difficulty of GENSCAN in ruling out coding regions in repeat sequences. However, such problems are not observed with GenePred, as predictions on both masked and unmasked sequence showed similar coverage (masked: 60.8%, unmasked: 61.8%) and accuracy (masked: 36.6%, unmasked: 34.4%).

Table 12. Performance of GenePred and GENSCAN in predicting VEGA annotated genes.



	GenePred	GENSCAN
Total Predictions	669	1086
Fused Predictions	198	149
Split Predictions	97	255
Genes covered	592	722
Coverage	61.8%	75.3%
Accurate Predictions	230	369
Accuracy	34.4%	34.0%

Table 13. Performance of GenePred and GENSCAN in predicting VEGA annotated exons and splice sites.



	GenePred	GenePred *with* GeneSplicer	GENSCAN
Total Predictions	11145	44050	8465
Fused Predictions	853	243	27
Split Predictions	212	842	340
Exons covered	2869	3960	4132
Coverage	44.5%	61.5%	64.2%
Accurate Predictions	1418	3138	3921
Accuracy	12.7%	7.1%	46.3%
Donor site coverage	6.7%	19.0%	57.9%
Donor site accuracy	3.5%	2.4%	43.3%
Acceptor site coverage	4.0%	19.6%	57.8%
Acceptor site accuracy	2.1%	2.5%	43.2%

Out of total predictions from both GenePred and GENSCAN, nearly 40% of predictions (excluding, 34% correct predictions and approximately 26% fused/split predictions) are not correlated with VEGA annotations. A number of these may turn out to represent real transcripts missing from the existing annotation. As GAZE predictions are based on regulatory signals, some of the predictions that do not match the annotation are likely to be non-coding transcripts. Recent experiments by Affymetrix on chromosome 20 and 22 emphasise this fact (Cawley *et al.*, 2004). They found that a significant number of transcription factor binding sites are correlated with non-coding RNAs and that they are regulated by a mechanism similar to that of protein coding genes. Thus, the excess predictions by GAZE are potential sequence blocks for hunting genes.

6.5 Tweaking GenePred gene prediction system

Having shown that the performance of GenePred is comparable with GENSCAN, I then tweaked Eponine models and configuration files used in making the GenePred prediction system to try to find improvements. I adopted three main approaches, which are explained below.

6.5.1 With Eponine translation models

With the availability of translation start and stop models, I decided to include them in the GenePred system in order to determine if this additional information might help in improving the performance. For this purpose, as explained earlier (Figure 61b), from the transcription start feature the model is allowed to transit to the translation start codon emitting the 5’ UTR segment. Likewise, between the acceptor site of the last exon and the transcription termination site, the translation stop signal features are introduced.

I tested this modified gene prediction system with the annotation from chromosome 20 and found that there is no significant change in coverage and accuracy when compared to GenePred without Eponine translation models (Table 14). However, the number of genes predicted by the system increased (886 predictions compared to 669 predictions reported previously) and because of it the coverage increased by a small proportion (64.9%, 622 annotated genes were correctly identified) and accuracy decreased by a small proportion (32.0%, 284 predictions are accurate). As there is a trade-off between coverage and accuracy, the values are comparable with the GenePred system without translation models. However, adding translation models to GenePred created less fused (155 predictions) and more split predictions (170). Thus, Eponine splice sites bias the gene prediction system to extend the gene rather than terminate the extending prediction. This issue is addressed in case (iii) below.

Table 14. Performance of GenePred constructed with translation start and stop features.

Total Predictions	Fused Predictions	Split Predictions	Genes covered	Coverage	Accurate Predictions	Accuracy
886	155	170	622	64.9%	284	32.0%

Thus, adding translation models to the GenePred did not affect the performance in identifying annotated genes from the genomic DNA but modified the number of fused and split predictions.

6.5.2 Eponine Splice site predictions replaced with GeneSplicer predictions

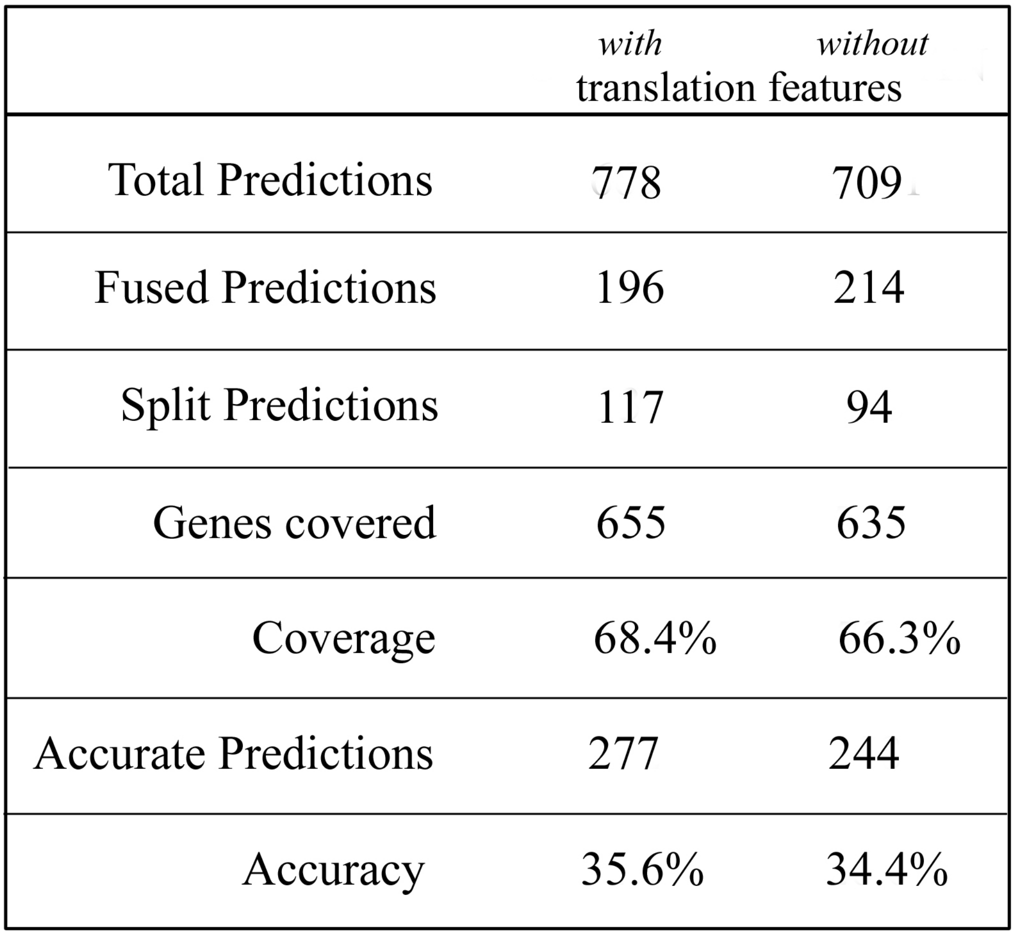
In another attempt, I replaced Eponine splice site model predictions with GeneSplicer predictions while making GenePred. As explained in chapter 4, GeneSplicer performed better than Eponine splice site models by using more information from the DNA sequence. Since splice sites form the essential part in determining the gene structure by any gene predictor, I attempted GeneSplicer predictions with GenePred in predicting genes. Since GeneSplicer predictions are given in bit scores (x), they are first converted to log scores (z) using the expression given below before usage.

$$z = \frac{1}{1+e^{-x}} \qquad (15)$$

GeneSplicer predictions with log scores are combined with both cases – with all Eponine models and with only Eponine transcription models (without translation models) – to derive a gene prediction system.

With the GeneSplicer features (along with transcription and translation features), the coverage (68.4%) and accuracy (35.6%) improved in comparison with GenePred using Eponine splice site features (Table 15). The increase in accuracy is due to the reduced number of predictions (778 predictions compared to 886) by the model. However, the number of fused predictions increased (196 compared to 155 predictions) when GeneSplicer splice site features are used. The results are similar, except that the number of predictions increased (709 predictions compared to 669) when translation model predictions were not used along with GeneSplicer. Including GeneSplicer predictions, however significantly improved exon and splice sites coverage by GenePred (Exon: 61.5%, Donor: 19.0%, Acceptor: 19.6%). This improvement in coverage is due to the increase in the number of predictions (44050 predictions compared to 11145, refer to Table 13) and hence the accuracy decreased by a small proportion.

Table 15. Performance of GenePred constructed with and without translation features along with GeneSplicer features instead of Eponine splice sites.



	with	*without*
	translation features	
Total Predictions	778	709
Fused Predictions	196	214
Split Predictions	117	94
Genes covered	655	635
Coverage	68.4%	66.3%
Accurate Predictions	277	244
Accuracy	35.6%	34.4%

Thus, the increase in coverage using features of GeneSplicer features narrowed the margin between the GenePred and the GENSCAN while keeping the high accuracy of the GenePred system.

6.5.3 Scaled down Eponine feature scores

As noted earlier Eponine donor and acceptor site model features are screened for scores above 0.999 and 0.9998 respectively, for constructing GenePred using GAZE.

On evaluating different gene structures from the DNA sequence based on the given model, GAZE tries to balance between splice sites and transcription termination features in extending or terminating the gene. This might be compared to the *in vivo* competition between transcriptome and spliceosome in transcribing a gene. At least in two cases – IgM heavy chain genes and Calcitonin genes – the competing nature of splicing and transcription is shown experimentally. An internal weak poly(A) signal present within an intron of the IgM heavy chain gene under the low amount of CstF-64 transcription factor, misses the poly(A) signal and hence the transcription continues with the influence of the donor splice site present downstream. In cases where CstF-64 is available in relatively high concentrations, as in plasma cells, the transcriptome has the advantage and terminates the transcription (Takagaki and Manley, 1998; Takagaki *et al.,* 1996). Similarly in Calcitonin gene transcription, a weak internal poly(A) signal is used by the transcriptome, if the SRp 20 protein, a splice regulatory factor, fails to get recruited to the nearby splice sites (Zhao *et al.,* 1999).

A high number of fused gene predictions by GenePred might be due to the higher score of splice sites than transcription termination features predicted by the Eponine models. To test this hypothesis, here I attempt to scale down the values of splice site features. This is done by taking the inverse logit of the Eponine score and multiplying it with a scaling factor and reconverting back to the logit score. Inverse logit of the Eponine score was done using the formula –

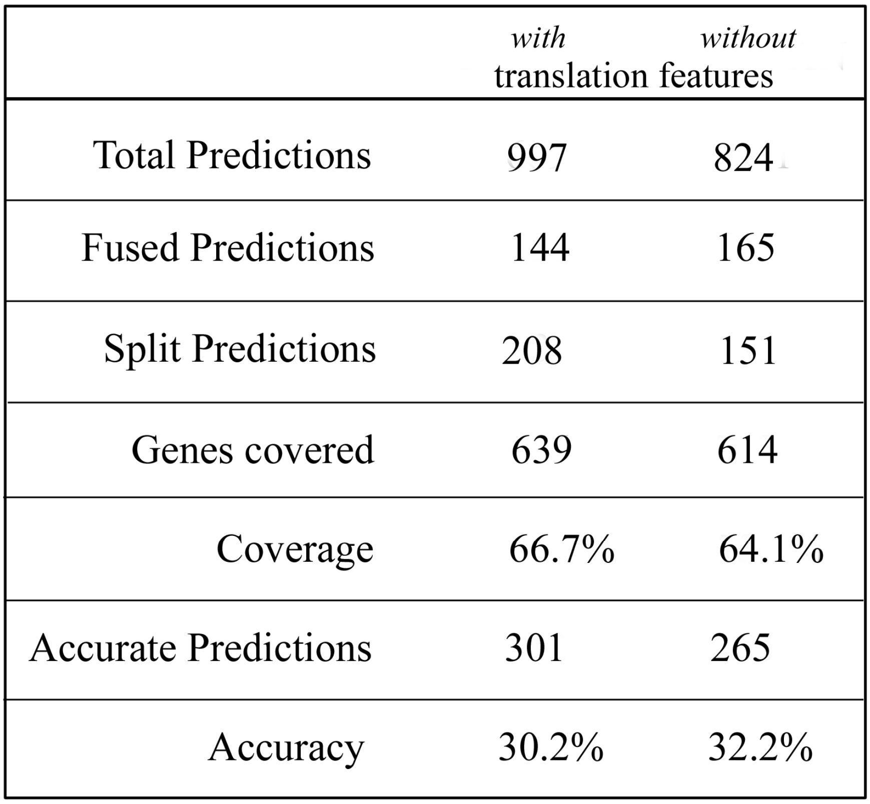
$$x = \log(\frac{z}{1-z}) \qquad (16)$$

The inverse logit score (*x*) for donor and acceptor sites are scaled down by multiplying the values with 0.67 and 0.54 respectively. These values were found to be optimum after different runs and the scaled down scores are more equivalent to the transcription start and termination model scores (0.99). Likewise, the scores for translation stop model features

(0.999) are also scaled down by multiplying the inverse logit scores with a factor of 0.67. Before incorporating the donor and acceptor and translation stop features into GenePred the scores are converted back to logit values using equation 20 explained above.

Table 16 shows the GenePred system with the scaled down feature scores predicted more split predictions (208 and 151 predictions compared to 170 and 97 by GenePred with no scaled down features) and less fused predictions (144 and 165 predictions compared to 155 and 198 predictions by GenePred without scaled down scores). The scenario is similar for exons as well (657 split predictions compared to 212 predictions without scaled down scores). Overall the number of predictions also increased (997 and 824 predictions). Although there is a small increase in coverage (66.7% and 64.1%), it was compensated with a small decrease in accuracy (30.2% and 32.2%) and hence the coverage and accuracy are not significantly different from the above models. However, this tweak showed that the high number of fused predictions by GenePred is due to the splice site score values fed into the gene prediction system.

Table 16. Performance of GenePred system constructed with and without translation after scaling down splice site and translation stop scores.



	with	*without*
	translation features	
Total Predictions	997	824
Fused Predictions	144	165
Split Predictions	208	151
Genes covered	639	614
Coverage	66.7%	64.1%
Accurate Predictions	301	265
Accuracy	30.2%	32.2%

6.6 Revisiting transcription termination predictions

In chapter 3, I showed that the Eponine model works better than existing programs, ERPIN and Polyadq, in predicting transcription termination sites. However, the model made a huge number of false positive predictions and nearly 10% of them lie within the genes. Ruling out these false positive predictions within the gene will increase the accuracy of the model. This is possible by defining the exon-intron structure of a gene and removing any transcription termination predictions lying within exons or introns. The exon-intron structure can be defined using GenePred and thus might help to pin-point the false transcription termination model predictions.

To achieve this objective, I used the GenePred system developed by omitting Eponine translation models (included Eponine transcription start site, donor, acceptor and transcription termination models only) for this purpose. The system predicted genes by including only appropriate transcription termination sites after defining the exon-intron structure using the splice site features given. Transcription termination sites selected by GenePred are then dumped to find the coverage and accuracy of the model by comparing it with the VEGA annotated gene ends in chromosome 20 (Table 17). Out of 98 predictions matching the 213 annotated genes of chromosome 20, 24 predictions lie within 2500 bases from the annotated gene end showing an accuracy of 24.5% with coverage of 40.4%. For a comparable coverage the earlier analysis (refer to chapter 3) showed only 16.6% accuracy for the transcription termination model.

Table 17. Performance of transcription termination model with the support of GenePred prediction system.

Transcription Termination model	Coverage	Accuracy
with GenePred support	40.4%	24.5%
without GenePred support	40%	16.6%

Thus, by defining the exon-intron structure, some of the internal predictions of transcription termination can be removed giving the model better accuracy with no compromise on coverage.

6.7 Concluding remarks

In this chapter, I tried to build a gene prediction system by taking advantage of the sequence features predicted by Eponine models explained in previous chapters and GAZE, a dynamic programming based gene assembler. Various versions of the gene prediction system, GenePred, showed that the coverage and accuracy are comparable with GENSCAN. This is respectable given no protein information is used by GenePred unlike GENSCAN. However, GenePred should be treated as complementary to GENSCAN rather than a replacement, given the following facts: firstly the coverage of the union of predictions of GENSCAN and GenePred is higher than the coverage by the individual programs (Figure 66) and secondly the very poor performance of GenePred in predicting exon-intron structures compared to GENSCAN. Figure 66 shows that out of 959 VEGA annotated genes, 490 genes are predicted both by GenePred and GENSCAN. Twenty percent (102/592) of GenePred predictions and 32% (232/722) of GENSCAN predictions do not overlap with each other. This indicates that by using GenePred and GENSCAN together a better coverage of the annotation can be attained.

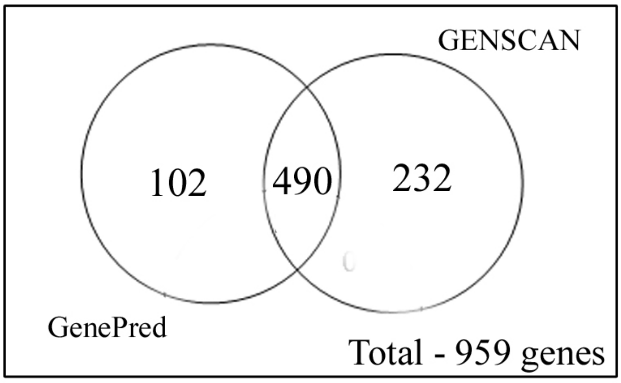
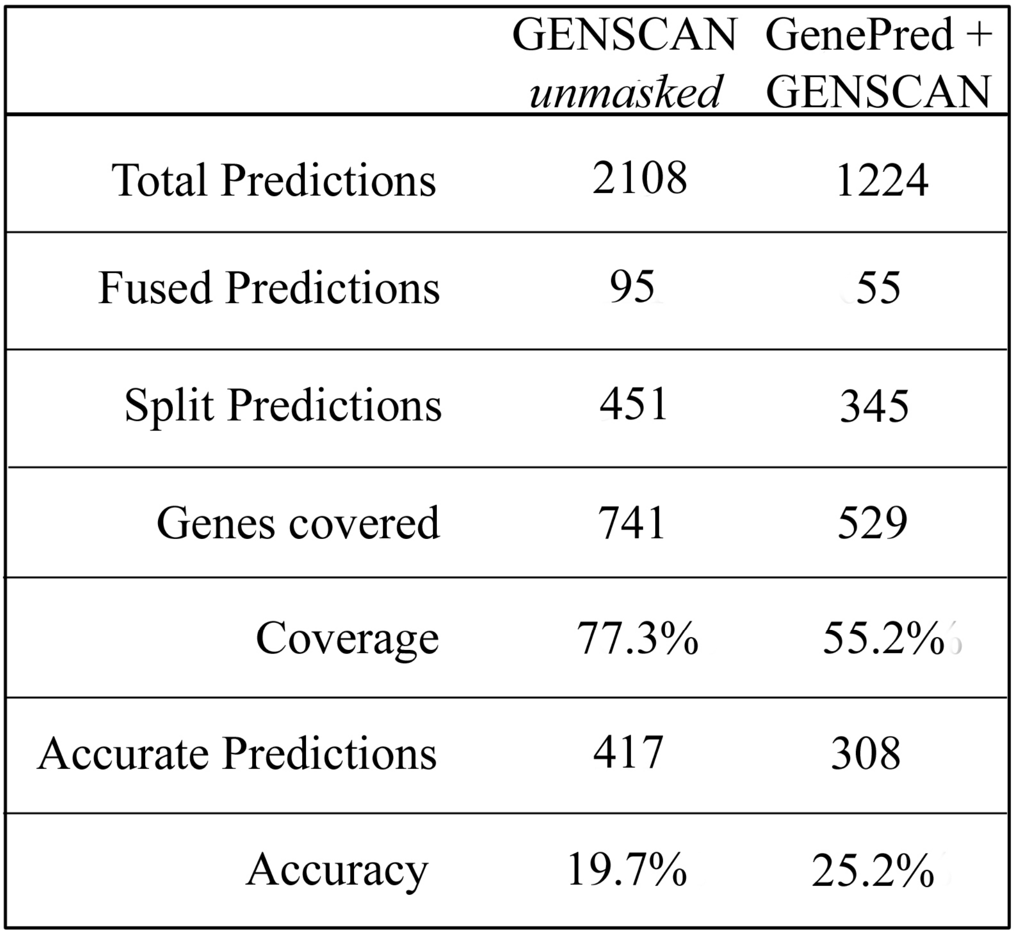


Figure 66. Venn diagram showing the coverage of GenePred and GENSCAN.

The accuracy of GENSCAN can also be improved by supplementing with the predictions of GenePred as indicated below. Table 18 and Table 19 show the accuracy of GENSCAN with and without GenePred in predicting VEGA annotated genes and exons respectively. A GENSCAN scan on the GenePred predicted regions of chromosome 20 improved its accuracy compared to using it alone on the unmasked chromosome sequence. These results again emphasise that GenePred should be treated as a complement to GENSCAN.

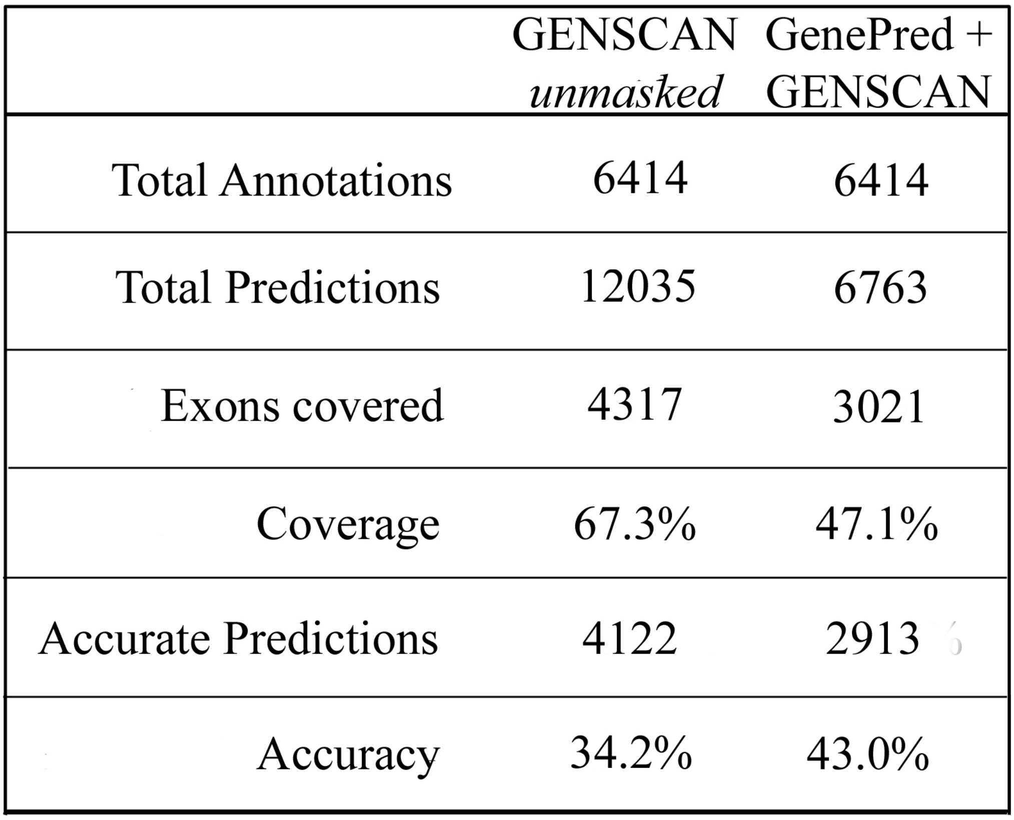
Detailed analysis of the predictions of GenePred as a percentage of nucleotides covered reveal that 97.6% of nucleotides in chromosome 20 are annotated by GenePred (Table 20). This number is very high and significantly higher than the fraction of genome covered by GENSCAN (68.5%, 43654921 bases) or by VEGA annotations (28632433 bases, 44.9%) of chromosome 20.

Table 18. Performance of GENSCAN with and without GenePred in predicting VEGA annotated genes.



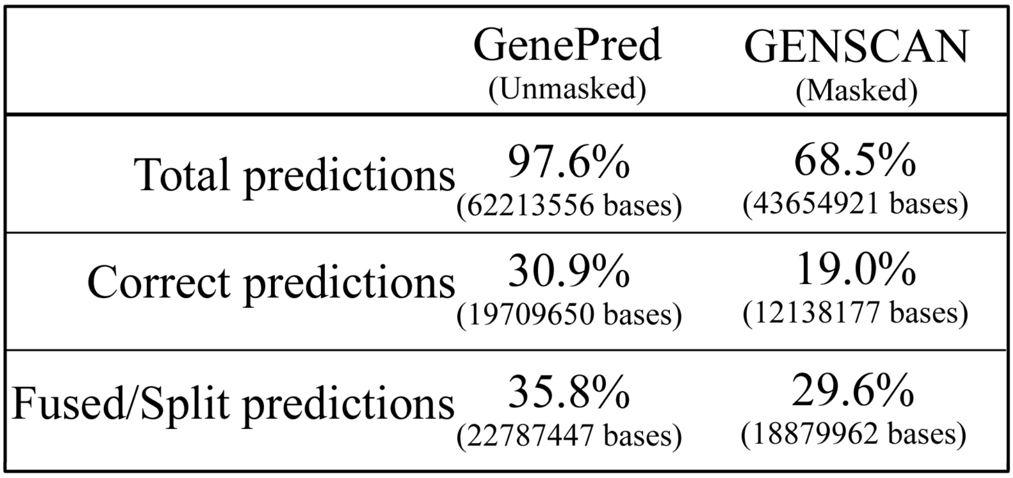
	GENSCAN *unmasked*	GenePred + GENSCAN
Total Predictions	2108	1224
Fused Predictions	95	55
Split Predictions	451	345
Genes covered	741	529
Coverage	77.3%	55.2%
Accurate Predictions	417	308
Accuracy	19.7%	25.2%

Table 19. Performance of GENSCAN with and without GenePred in predicting VEGA annotated exons.



	GENSCAN *unmasked*	GenePred + GENSCAN
Total Annotations	6414	6414
Total Predictions	12035	6763
Exons covered	4317	3021
Coverage	67.3%	47.1%
Accurate Predictions	4122	2913
Accuracy	34.2%	43.0%

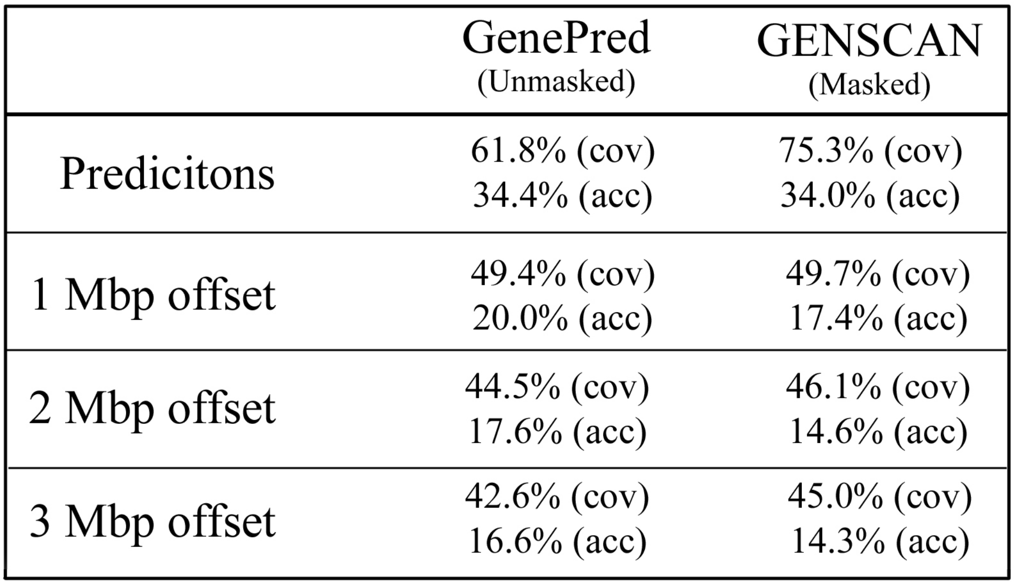
Table 20. Nucleotide coverage by predictions of GenePred and GENSCAN.



	GenePred (Unmasked)	GENSCAN (Masked)
Total predictions	97.6% (62213556 bases)	68.5% (43654921 bases)
Correct predictions	30.9% (19709650 bases)	19.0% (12138177 bases)
Fused/Split predictions	35.8% (22787447 bases)	29.6% (18879962 bases)

These results indicate that GenePred’s prediction accuracy comes mainly by determining the correct strand to transcribe, yet it is performing better than random: Random prediction accuracy was evaluated by offsetting the predictions of GenePred and GENSCAN by 1, 2 and 3 mega bases (predictions exceeding the length of the chromosome are rotated round to the beginning) and recalculating the coverage and accuracy with respect to VEGA annotation (Table 21). GenePred predictions offset by 3 mega bases shows 42.6% coverage and 16.6% accuracy, which is significantly less than for the original predictions (coverage: 61.8%, accuracy: 34.4%). Similar results are found for GENSCAN predictions as well.

Table 21. Coverage and accuracy of GenePred and GENSCAN for predictions offset by 1, 2 and 3 mega bases.



	GenePred (Unmasked)	GENSCAN (Masked)
Predicitons	61.8% (cov) 34.4% (acc)	75.3% (cov) 34.0% (acc)
1 Mbp offset	49.4% (cov) 20.0% (acc)	49.7% (cov) 17.4% (acc)
2 Mbp offset	44.5% (cov) 17.6% (acc)	46.1% (cov) 14.6% (acc)
3 Mbp offset	42.6% (cov) 16.6% (acc)	45.0% (cov) 14.3% (acc)

Although GENSCAN coverage is better than GenePred overall, it is less likely than GenePred to predict VEGA ‘Novel_transcripts’ and ‘Putative’ genes. This may be partly due to GENSCAN’s reliance on protein information. Novel_transcripts are genes annotated from RNA that have weak evidence for being coding transcripts. Likewise, Putative genes are annotated using EST evidence and these genes also have no clear open reading frame. As the protein information content of this set of transcripts is less than for known genes, this may explain GENSCAN predicting few cases than GenePred (Table 22). For Novel_transcripts, all versions of GenePred discussed above show better coverage percentage with twice the accuracy of GENSCAN. Similarly for Putative genes, GenePred predicted at least 15% more genes with twice the accuracy of GENSCAN or more. On the combined dataset (Novel_transcripts + Putative genes), GenePred’s coverage was at least

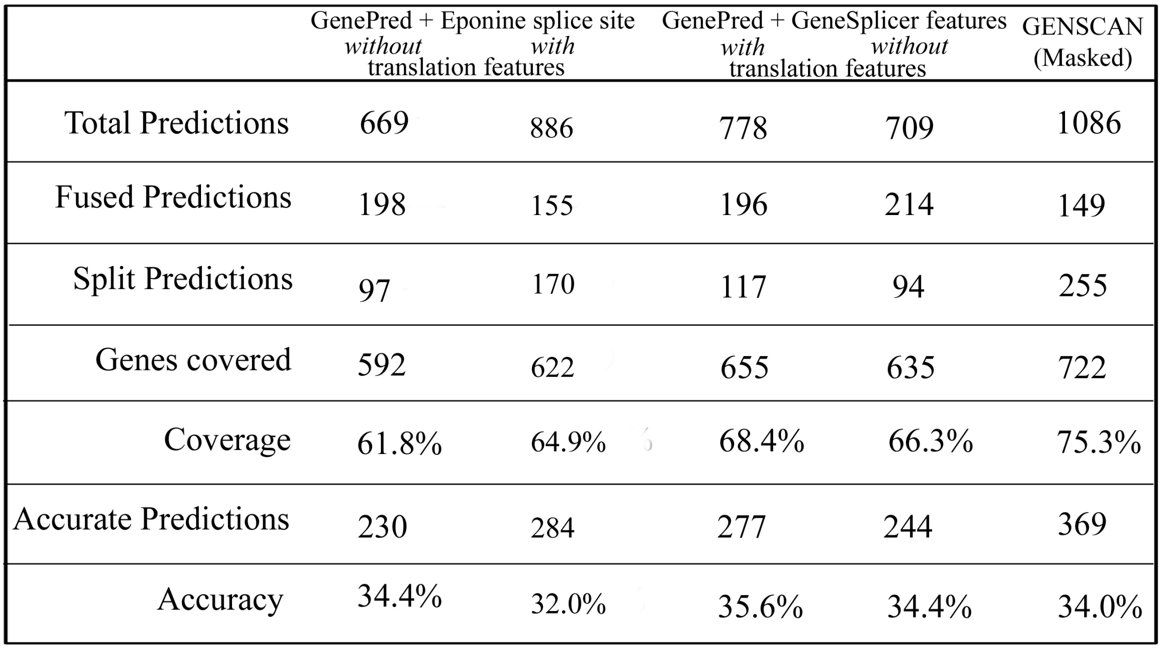
10% more than GENSCAN with twice the accuracy or more. Table 22 details the coverage and accuracy of various versions of GenePred (with and without splice site and translation models) compared to GENSCAN. The low accuracy values are a consequence of considering predictions matching only Novel_transcripts and Putative genes as true and the rest as false predictions.

Table 22. Performance of GenePred and GENSCAN in identifying VEGA Novel_transcripts and Putative genes. Coverage and accuracy for each annotation is given for GenePred with and without translation models. Each of these GenePred systems is combined with either Eponine splice site or GeneSplicer features. Numbers in brackets shows the absolute values.

Annotation	GenePred + Eponine splice site		GenePred + GeneSplicer features		GENSCAN
	without	*with*	*without*	*with*	
	translation features		translation features		
Novel Transcripts	55.5 (50/90) 4.5 (37/824)	57.7 (52/90) 3.8 (38/997)	57.7 (52/90) 5.1 (36/709)	58.8 (53/90) 5.0 (39/778)	50.0 (47/90) 2.9 (33/1086)
Putative genes	48.4 (76/157) 7.0 (58/824)	50.9 (80/157) 6.2 (62/997)	57.9 (91/157) 8.9 (63/709)	60.5 (95/157) 8.5 (66/778)	35.0 (54/157) 4.3 (45/1086)
Novel Transcripts + Putative	51.0 (126/247) 10.6 (87/824)	53.4 (132/247) 9.0 (90/997)	57.9 (143/247) 12.6 (89/709)	59.9 (148/247) 12.2 (95/778)	40.5 (101/247) 6.5 (75/1086)

Thus, in this project I was able to develop a gene prediction system based purely on gene regulatory signals and show that its performance is encouraging considering that it does not rely on protein coding information. In terms of gene coverage it performs similarly for protein coding genes and better for genes with no coding evidence. At present the major problem is the very poor exon prediction accuracy despite including a splicing model. For easy comparison, in the table below (Table 23), I summarise the results of various versions of GenePred compared to GENSCAN in annotating chromosome 20 VEGA annotated genes.

Table 23. Summary of performance of various versions of GenePred and GENSCAN in identifying VEGA annotated genes in human chromosome 20



	GenePred + Eponine splice site		GenePred + GeneSplicer features		GENSCAN (Masked)
	without translation features	*with* translation features	*with* translation features	*without* translation features	
Total Predictions	669	886	778	709	1086
Fused Predictions	198	155	196	214	149
Split Predictions	97	170	117	94	255
Genes covered	592	622	655	635	722
Coverage	61.8%	64.9%	68.4%	66.3%	75.3%
Accurate Predictions	230	284	277	244	369
Accuracy	34.4%	32.0%	35.6%	34.4%	34.0%

CONCLUSIONS

Current genomic revolution has unlocked the potential to understand the gene regulation at molecular, cellular and physiological levels. The first step in this process is to identify the genes present in a genome and study the expression patterns of the gene influenced by regulatory signals. Several programs are available in the public domain that can identify genes from the DNA sequence using ‘signals’ and ‘contents’ of the DNA sequence. Gene prediction programs using ‘contents’ information are limited from identifying only protein coding genes in the genome. So in order to derive an *ab initio* gene prediction system purely based on signals, in this project I attempted to create models for gene regulatory elements.

The start and end of any gene is marked by their promoter and termination signals where from RNA polymerases begin and terminate transcription. The transcription start site was initially identified by Down and Hubbard using generalised linear model based probabilistic algorithm called Eponine (Down and Hubbard, 2004; Down and Hubbard, 2002). In this project, I attempted a number of methods including Eponine to identify transcription termination signals responsible for RNA polymerase stop and release from DNA.

Termination of polymerase does not happen at the cleavage site and the RNA polymerase transcribes DNA even 2 kb downstream before releasing from the DNA. Recent experiments confirm the presence of a pause site downstream of the cleavage site required for transcription termination. In chapter 1, I have detailed the mechanism of transcription termination and compared with other systems known to occur *in vivo*. Attempts to identify the pause elements have so far not been successful in deriving a consensus sequence. However experimental and computational analyses indicate the sequence might be A-rich and G-rich and bind MAZ and Sp1 protein to stop transcription from running-over to the neighbouring genes.

In this project, I first used base compositional analysis to study any significant changes in the nucleotide distribution in the sequences around cleavage site. The differences in the composition were found concentrated within 100 and 50 bases upstream and downstream of cleavage site and these are linked to the poly(A) signal and GT rich region known earlier.

No significant changes were found in the sequences where polymerase is likely to pause. Then, I investigated for the presence of any secondary structures that can potentially stop polymerase as a similar mechanism is found in prokaryotes and histone genes. So to analyse this, I used Nussinov and Zuker algorithms. Base pair maximisation principle-based Nussinov algorithm did not find any stem-loop structures. Free energy minimization based Zuker algorithm, however, predicted the possibility of RNA secondary structure in the sequences 100 to 650 bases downstream of cleavage site. Correlation with GC and GT percentage showed they are unlikely to be caused by sequence artefacts. Confirming these structures using biochemical experiments will help us to understand the mechanism of transcription termination of protein coding genes and correlate them with histone and prokaryotic gene transcription termination.

After analysing for secondary structures in DNA, I used the probabilistic machine learning algorithm based on Bayes theorem and Generalized Linear Models, Eponine, for scanning motifs responsible for transcription termination. The model captured poly(A) signal and auxiliary sequence motifs along with a few multiplex signals that might be responsible for polymerase II pause and termination. An evaluation of this termination model against annotated human chromosomes shows that the model performs better than existing methods. However a significant number of predictions also appear near the annotated start site and first intron of genes. In chapter 3, I have tried to explain these biases and false positives at this region using hypothesis derived from previous knowledge. I propose that a significant number of predictions made by the model that are not correlated with available annotations are not really false predictions and they are likely to have biological functions. It would be interesting to test these hypotheses by devising appropriate molecular and biochemical experiments.

Apart from the bias towards transcription start site and first intron, I found approximately 10% of predictions lie within genes and their density is correlated with gene length and intron size. Interestingly shorter introns were found to have higher prediction density and most of them are likely to be alternative termination or polyadenylation site of the gene. Early experiments show this is possible as at least 22% of mRNAs was recorded to undergo alternative polyadenylation often in a tissue- and time-specific manner (Legendre and Gautheret, 2003). Previous programs developed to find the end of the gene and alternative polyadenylation site are mainly dependent on poly(A) signals and Eponine differs from them by using other downstream signals. A comparison with one such previous program, ERPIN (Legendre and Gautheret, 2003), showed Eponine performed better in identifying transcription termination sites.

I then extended the application of Eponine to develop splice site and translation models to meet the objective of creating an *ab initio* gene prediction system. These models are explained in chapters 4 and 5. Donor and acceptor site models were trained from sequences from chromosome 22 along with appropriate negative datasets. Positional variations in splice site models were captured using a Delta distribution rather than the usual Gaussian distribution. The models picked the known signals near donor and acceptor sites. Acceptor sites, as expected, were difficult to predict relative to the donor site as acceptor sites show variation in the regulatory elements (Lund *et al.*, 2000). Moreover, the Eponine acceptor site model did not capture branch point signal where lariat formation occurs. A comparison of the models with annotated sites of chromosome 20 showed the models have good positional accuracy and performed comparably with GeneSplicer (Pertea *et al.*, 2001) and StrataSplice (Levine, 2001a). I also noticed that there is a scope for improvement of performance of Eponine splice site models by using local GC variation as employed by StrataSplice.

Likewise, I attempted to identify translation start and stop codons and regulatory elements near by that determine translation initiation and termination by the ribosomal machinery. Translation start model learnt the famous Kozak sequence and performed better than NetStart (Pedersen and Nielsen, 1997b) although less well than ATGpr (Salamov *et al.*, 1998a).

After training all the Eponine models, I combined them using the dynamic programming framework based GAZE (Howe *et al.*, 2002) to develop a gene prediction system called GenePred. Various versions of GenePred developed by tweaking the input features and score values showed all the models are comparable with GENSCAN (Burge and Karlin, 1997) in identifying genes from the genomic sequence. In cases of Novel_transcripts and Putative genes, GenePred was found to be better than GENSCAN in identifying these genes. However, GenePred had difficulty in determining the annotated exon-intron structure of the genes. This is expected as the GenePred uses only signal information in predicting the candidate genes.

Thus in this project, I developed various models that influence gene regulatory elements and linked them together to derive an *ab initio* gene prediction system that uses only these gene regulatory signals and not dependent on protein coding information. During this attempt, I found interesting observations like distribution of termination sites near transcription start site, first intron and short introns. Results from experiments confirming these observations will help us to discern the transcriptional machinery and reconsider the current concepts of gene regulation in the eukaryotic genome.

BIBLIOGRAPHY

Adema, G.J., Bovenberg, R.A., Jansz, H.S. and Baas, P.D. (1988) Unusual branch point selection involved in splicing of the alternatively processed Calcitonin/CGRP-I pre-mRNA. *Nucleic Acids Research*, **16**, 9513-9526.

Agarwal, P. and Bafna, V. (1998) The ribosome scanning model for translation initiation: implications for gene prediction and full-length cDNA detection. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, **6**, 2-7.

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990) Basic local alignment search tool. *J Mol Biol*, **215**, 403-410.

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, **25**, 3389-3402.

Apic, G., Gough, J. and Teichmann, S.A. (2001) Domain combinations in archaeal, eubacterial and eukaryotic proteomes. *J Mol Biol*, **310**, 311-325.

Aranda, A. and Proudfoot, N. (2001) Transcriptional termination factors for RNA polymerase II in yeast. *Molecular Cell*, **7**, 1003-1011.

Aranda, A. and Proudfoot, N.J. (1999) Definition of transcriptional pause elements in fission yeast. *Molecular and Cellular Biology*, **19**, 1251-1261.

Ashfield, R., Enriquez-Harris, P. and Proudfoot, N.J. (1991) Transcriptional termination between the closely linked human complement genes C2 and factor B: common termination factor for C2 and c-myc? *EMBO Journal*, **10**, 4197-4207.

Ashurst, J. (2002) http://vega.sanger.ac.uk

Ashurst, J.L. and Collins, J.E. (2003) Gene Annotation: Prediction and Testing. *Annual Review of Genomics and Human Genetics*, **4**, 69-88.

Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L., Studholme, D.J., Yeats, C. and Eddy, S.R. (2004) The Pfam protein families database. *Nucleic Acids Research*, **32**, D138-141.

Beaumont, C., Porcher, C., Picat, C., Nordmann, Y. and Grandchamp, B. (1989) The mouse porphobilinogen deaminase gene. Structural organization, sequence, and transcriptional analysis. *Journal of Biological Chemistry*, **264**, 14829-14834.

Bengal, E., Flores, O., Krauskopf, A., Reinberg, D. and Aloni, Y. (1991) Role of the mammalian transcription factors IIF, IIS, and IIX during elongation by RNA polymerase II. **11**, 1195-1206.

Bentley, D.L. and Groudine, M. (1986) A block to elongation is largely responsible for decreased transcription of c-myc in differentiated HL60 cells. *Nature*, **321**, 702-706.

Berget, S.M. (1995) Exon recognition in vertebrate splicing. *Journal of Biological Chemistry*, **270**, 2411-2414.

Berman, H.M., Battistuz, T., Bhat, T.N., Bluhm, W.F., Bourne, P.E., Burkhardt, K., Feng, Z., Gilliland, G.L., Iype, L., Jain, S., Fagan, P., Marvin, J., Padilla, D., Ravichandran, V., Schneider, B., Thanki, N., Weissig, H., Westbrook, J.D. and Zardecki, C. (2002) The Protein Data Bank. *Acta Crystallogr D Biol Crystallogr*, **58**, 899-907.

Betton, J.M., Jacob, J.P., Hofnung, M. and Broome-Smith, J.K. (1997) Creating a bifunctional protein by insertion of beta-lactamase into the maltodextrin-binding protein. *Nat Biotechnol*, **15**, 1276-1279.

Birney, E., Andrews, D., Bevan, P., Caccamo, M., Cameron, G., Chen, Y., Clarke, L., Coates, G., Cox, T., Cuff, J., Curwen, V., Cutts, T., Down, T., Durbin, R., Eyras, E., Fernandez-Suarez, X.M., Gane, P., Gibbins, B., Gilbert, J., Hammond, M., *et al.* (2004) Ensembl 2004. *Nucleic Acids Research*, **32**, D468-470.

Birney, E. and Durbin, R. (1997) Dynamite: a flexible code generating language for dynamic programming methods used in sequence comparison. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, **5**, 56-64.

Birse, C.E., Lee, B.A., Hansen, K. and Proudfoot, N.J. (1997) Transcriptional termination signals for RNA polymerase II in fission yeast. *EMBO Journal*, **16**, 3633-3643.

Birse, C.E., Minvielle-Sebastia, L., Lee, B.A., Keller, W. and Proudfoot, N.J. (1998) Coupling termination of transcription to messenger RNA maturation in yeast. *Science*, **280**, 298-301.

Black, D.L. (2003) Mechanisms of alternative pre-messenger RNA splicing. *Annual Review of Biochemistry*, **72**, 291-336.

Bogenhagen, D.F. and Brown, D.D. (1981) Nucleotide sequences in Xenopus 5S DNA required for transcription termination. *Cell*, **24**, 261-270.

Bork, P., Downing, A.K., Kieffer, B. and Campbell, I.D. (1996) Structure and distribution of modules in extracellular proteins. *Q Rev Biophys*, **29**, 119-167.

Bossemeyer, D. (1994) The glycine-rich sequence of protein kinases: a multifunctional element. *Trends Biochem Sci*, **19**, 201-205.

Braddock, M., Muckenthaler, M., White, M.R., Thorburn, A.M., Sommerville, J., Kingsman, A.J. and Kingsman, S.M. (1994) Intron-less RNA injected into the nucleus of Xenopus oocytes accesses a regulated translation control pathway. *Nucleic Acids Research*, **22**, 5255-5264.

Brendel, V. and Kleffe, J. (1998) Prediction of locally optimal splice sites in plant pre-mRNA with applications to gene identification in Arabidopsis thaliana genomic DNA. *Nucleic Acids Research*, **26**, 4748-4757.

Brenner, S.E., Chothia, C. and Hubbard, T.J. (1998) Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proc Natl Acad Sci U S A*, **95**, 6073-6078.

Briggs, D., Jackson, D., Whitelaw, E. and Proudfoot, N.J. (1989) Direct demonstration of termination signals for RNA polymerase II from the sea urchin H2A histone gene. *Nucleic Acids Research*, **17**, 8061-8071.

Brinster, R.L., Allen, J.M., Behringer, R.R., Gelinas, R.E. and Palmiter, R.D. (1988) Introns increase transcriptional efficiency in transgenic mice. *Proceedings of the National Academy of Sciences of the United States of America*, **85**, 836-840.

Burge, C. (1998) *Modeling dependencies in pre-mRNA splicing signals*. Elsevier, Amsterdam, Netherlands.

Burge, C. and Karlin, S. (1997) Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, **268**, 78-94.

Burset, M., Seledtsov, I.A. and Solovyev, V.V. (2000) Analysis of canonical and non-canonical splice sites in mammalian genomes. *Nucleic Acids Research*, **28**, 4364-4375.

Burset, M., Seledtsov, I.A. and Solovyev, V.V. (2001) SpliceDB: database of canonical and non-canonical mammalian splice sites. *Nucleic Acids Research*, **29**, 255-259.

Calvo, O. and Manley, J.L. (2001) Evolutionarily conserved interaction between CstF-64 and PC4 links transcription, polyadenylation, and termination. *Molecular Cell*, **7**, 1013-1023.

Calvo, O. and Manley, J.L. (2003) Strange bedfellows: polyadenylation factors at the promoter. *Genes and Development*, **17**, 1321-1327.

Cavener, D.R. and Ray, S.C. (1991) Eukaryotic start and stop translation sites. *Nucleic Acids Research*, **19**, 3185-3192.

Cawley, S., Bekiranov, S., Ng, H.H., Kapranov, P., Sekinger, E.A., Kampa, D., Piccolboni, A., Sementchenko, V., Cheng, J., Williams, A.J., Wheeler, R., Wong, B., Drenkow, J., Yamanaka, M., Patel, S., Brubaker, S., Tammana, H., Helt, G., Struhl, K. and Gingeras, T.R. (2004) Unbiased Mapping of Transcription Factor Binding Sites along Human Chromosomes 21 and 22 Points to Widespread Regulation of Noncoding RNAs. *Cell*, **116**, 499-509.

Celniker, S.E., Wheeler, D.A., Kronmiller, B., Carlson, J.W., Halpern, A., Patel, S., Adams, M., Champe, M., Dugan, S.P., Frise, E., Hodgson, A., George, R.A., Hoskins, R.A., Laverty, T., Muzny, D.M., Nelson, C.R., Pacleb, J.M., Park, S., Pfeiffer, B.D., Richards, S., *et al.* (2002) Finishing a whole-genome shotgun: release 3 of the Drosophila melanogaster euchromatic genome sequence. *Genome Biol*, **3**, RESEARCH0079.

Chafin, D.R., Claussen, T.J. and Price, D.H. (1991) Identification and purification of a yeast protein that affects elongation by RNA polymerase II. **266**, 9256-9262.

Chandonia, J.M., Walker, N.S., Lo Conte, L., Koehl, P., Levitt, M. and Brenner, S.E. (2002) ASTRAL compendium enhancements. *Nucleic Acids Res*, **30**, 260-263.

Chinsky, J.M., Maa, M.C., Ramamurthy, V. and Kellems, R.E. (1989) Adenosine deaminase gene expression. Tissue-dependent regulation of transcriptional elongation. *Journal of Biological Chemistry*, **264**, 14561-14565.

Chothia, C. (1992) Proteins. One thousand families for the molecular biologist. *Nature*, **357**, 543-544.

Chothia, C. and Lesk, A.M. (1986) The relation between the divergence of sequence and structure in proteins. *Embo J*, **5**, 823-826.

Collins, J.E., Goward, M.E., Cole, C.G., Smink, L.J., Huckle, E.J., Knowles, S., Bye, J.M., Beare, D.M. and Dunham, I. (2003) Reevaluating human gene annotation: a second-generation analysis of chromosome 22. *Genome Research*, **13**, 27-36.

Connelly, S. and Manley, J.L. (1989a) A CCAAT box sequence in the adenovirus major late promoter functions as part of an RNA polymerase II termination signal. *Cell*, **57**, 561-571.

Connelly, S. and Manley, J.L. (1989b) RNA polymerase II transcription termination is mediated specifically by protein binding to a CCAAT box sequence. *Molecular and Cellular Biology*, **9**, 5254-5259.

Cook, P.R. (1999) The organization of replication and transcription. *Science*, **284**, 1790-1795.

Cooke, C., Hans, H. and Alwine, J.C. (1999) Utilization of splicing elements and polyadenylation signal elements in the coupling of polyadenylation and last-intron removal. *Molecular and Cellular Biology*, **19**, 4971-4979.

Cosson, B., Couturier, A., Chabelskaya, S., Kiktev, D., Inge-Vechtomov, S., Philippe, M. and Zhouravleva, G. (2002) Poly(A)-Binding Protein Acts in Translation Termination via Eukaryotic Release Factor 3 Interaction and Does Not Influence [PSI+] Propagation. *Molecular and Cellular Biology*, **22**, 3301-3315.

Cozzarelli, N.R., Gerrard, S.P., Schlissel, M., Brown, D.D. and Bogenhagen, D.F. (1983) Purified RNA polymerase III accurately and efficiently terminates transcription of 5S RNA genes. *Cell*, **34**, 829-835.

Cramer, P., Caceres, J.F., Cazalla, D., Kadener, S., Muro, A.F., Baralle, F.E. and Kornblihtt, A.R. (1999) Coupling of transcription with alternative splicing: RNA pol II promoters modulate SF2/ASF and 9G8 effects on an exonic splicing enhancer. *Molecular Cell*, **4**, 251-258.

Cramer, P., Pesce, C.G., Baralle, F.E. and Kornblihtt, A.R. (1997) Functional association between promoter structure and transcript alternative splicing. *Proceedings of the National Academy of Sciences of the United States of America*, **94**, 11456-11460.

Cramer, P., Srebrow, A., Kadener, S., Werbajh, S., de la Mata, M., Melen, G., Nogues, G. and Kornblihtt, A.R. (2001) Coordination between transcription and pre-mRNA processing. *FEBS Letters*, **498**, 179-182.

Crispino, J.D., Mermoud, J.E., Lamond, A.I. and Sharp, P.A. (1996) Cis-acting elements distinct from the 5' splice site promote U1-independent pre-mRNA splicing. *RNA*, **2**, 664-673.

Cristianini, N. and Shawe-Taylor, J. (2000) *An introduction to support vector machines*. Cambridge University Press, Cambridge.

CSC. (1998) Genome sequence of the nematode C. elegans: a platform for investigating biology. The C. elegans Sequencing Consortium. *Science*, **282**, 2012-2018.

Cuello, P., Boyd, D.C., Dye, M.J., Proudfoot, N.J. and Murphy, S. (1999) Transcription of the human U2 snRNA genes continues beyond the 3' box in vivo. *EMBO Journal*, **18**, 2867-2877.

Dantonel, J.C., Murthy, K.G., Manley, J.L. and Tora, L. (1997) Transcription factor TFIID recruits factor CPSF for formation of 3' end of mRNA. *Nature*, **389**, 399-402.

Das, A. (1993) Control of transcription termination by RNA-binding proteins. *Annual Review of Biochemistry*, **62**, 893-930.

di Bernardo, D., Down, T. and Hubbard, T. (2003) ddbRNA: detection of conserved secondary structures in multiple alignments. *Bioinformatics*, **19**, 1606-1611.

Dichtl, B., Blank, D., Ohnacker, M., Friedlein, A., Roeder, D., Langen, H. and Keller, W. (2002a) A role for SSU72 in balancing RNA polymerase II transcription elongation and termination. *Molecular Cell*, **10**, 1139-1150.

Dichtl, B., Blank, D., Sadowski, M., Hubner, W., Weiser, S. and Keller, W. (2002b) Yhh1p/Cft1p directly links poly(A) site recognition and RNA polymerase II transcription termination. *EMBO Journal*, **21**, 4125-4135.

Ditzel, L., Lowe, J., Stock, D., Stetter, K.O., Huber, H., Huber, R. and Steinbacher, S. (1998) Crystal structure of the thermosome, the archaeal chaperonin and homolog of CCT. *Cell*, **93**, 125-138.

Doi, N., Itaya, M., Yomo, T., Tokura, S. and Yanagawa, H. (1997) Insertion of foreign random sequences of 120 amino acid residues into an active enzyme. *FEBS Lett*, **402**, 177-180.

Dong, S. and Searls, D.B. (1994) Gene structure prediction by linguistic methods. *Genomics*, **23**, 540-551.

Dowell, R.D., Jokerst, R.M., Day, A., Eddy, S.R. and Stein, L. (2001) The Distributed Annotation System. *BMC Bioinformatics*, **2**, 7.

Down, T. (2003) Computational localization of promoters and transcription start sites in mammalian genomes. *The Wellcome Trust Sanger Institute*. University of Cambridge, Cambridge, p. 149.

Down, T. and Hubbard, T. (2004) Relevance Vector Machines for classifying points and regions in biological sequences.

Down, T.A. and Hubbard, T.J. (2002) Computational detection and location of transcription start sites in mammalian genomic DNA. *Genome Research*, **12**, 458-461.

Dunham, I., Shimizu, N., Roe, B.A., Chissoe, S., Hunt, A.R., Collins, J.E., Bruskiewich, R., Beare, D.M., Clamp, M., Smink, L.J., Ainscough, R., Almeida, J.P., Babbage, A., Bagguley, C., Bailey, J., Barlow, K., Bates, K.N., Beasley, O., Bird, C.P., Blakey, S., *et al.* (1999) The DNA sequence of human chromosome 22. *Nature*, **402**, 489-495.

Duplay, P., Szmelcman, S., Bedouelle, H. and Hofnung, M. (1987) Silent and functional changes in the periplasmic maltose-binding protein of *Escherichia coli* K12. I. Transport of maltose. *Journal of Molecular Biology*, **194**, 663-673.

Durbin, R., Eddy, S.R., Krogh, A. and Mitchison, G. (1998) *Biological Sequence Analysis*. Cambridge University Press.

Dye, M.J. and Proudfoot, N.J. (2001) Multiple transcript cleavage precedes polymerase release in termination by RNA polymerase II. *Cell*, **105**, 669-681.

Eddy, S.R. (1996) Hidden Markov models. *Curr Opin Struct Biol*, **6**, 361-365.

Edwalds-Gilbert, G., Prescott, J. and Falck-Pedersen, E. (1993) 3' RNA processing efficiency plays a primary role in generating termination-competent RNA polymerase II elongation complexes. *Molecular and Cellular Biology*, **13**, 3472-3480.

Enright, A.J., Van Dongen, S. and Ouzounis, C.A. (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res*, **30**, 1575-1584.

Enriquez-Harris, P., Levitt, N., Briggs, D. and Proudfoot, N.J. (1991) A pause site for RNA polymerase II is associated with termination of transcription. *EMBO Journal*, **10**, 1833-1842.

Evers, R., Smid, A., Rudloff, U., Lottspeich, F. and Grummt, I. (1995) Different domains of the murine RNA polymerase I-specific termination factor mTTF-I serve distinct functions in transcription termination. *EMBO Journal*, **14**, 1248-1256.

Ewing, B. and Green, P. (2000) Analysis of expressed sequence tags indicates 35,000 human genes. *Nature Genetics*, **25**, 232-234.

Friedman, D.I. and Court, D.L. (1995) Transcription antitermination: the lambda paradigm updated. *Molecular Microbiology*, **18**, 191-200.

Ganem, C., Devaux, F., Torchet, C., Jacq, C., Quevillon-Cheruel, S., Labesse, G., Facca, C. and Faye, G. (2003) Ssu72 is a phosphatase essential for transcription termination of snoRNAs and specific mRNAs in yeast. *EMBO Journal*, **22**, 1588-1598.

Gautheret, D. and Lambert, A. (2001) Direct RNA motif definition and identification from multiple sequence alignments using secondary structure profiles. *Journal of Molecular Biology*, **313**, 1003-1011.

Gavin, A.C., Bosche, M., Krause, R., Grandi, P., Marzioch, M., Bauer, A., Schultz, J., Rick, J.M., Michon, A.M., Cruciat, C.M., Remor, M., Hofert, C., Schelder, M., Brajenovic, M., Ruffner, H., Merino, A., Klein, K., Hudak, M., Dickson, D., Rudi, T., *et al.* (2002) Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, **415**, 141-147.

Ge, H. and Roeder, R.G. (1994) Purification, cloning, and characterization of a human coactivator, PC4, that mediates transcriptional activation of class II genes. *Cell*, **78**, 513-523.

Gene Ontology Consortium. (2004) The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research*, **32**, D258-261.

Gerber, J.K., Gogel, E., Berger, C., Wallisch, M., Muller, F., Grummt, I. and Grummt, F. (1997) Termination of mammalian rDNA replication: polar arrest of replication fork movement by transcription termination factor TTF-I. *Cell*, **90**, 559-567.

Gerstein, M. (1997) A structural census of genomes: comparing bacterial, eukaryotic, and archaeal genomes in terms of protein structure. *Journal of Molecular Biology*, **274**, 562-576.

Gibson, W.T. and Dormor, D.J. (2003) Searching for the 'natural': the case for the gene 'for' homosexuality. *Journal of Human Reproduction and Genetic Ethics*, **9**, 30-35.

Graveley, B.R. (2000) Sorting out the complexity of SR protein functions. *RNA*, **6**, 1197-1211.

Gray, N.K. and Wickens, M. (1998) Control of translation initiation in animals. *Annual Review of Cell and Developmental Biology*, **14**, 399-458.

Greenblatt, J., Nodwell, J.R. and Mason, S.W. (1993) Transcriptional antitermination. *Nature*, **364**, 401-406.

Greger, I.H., Demarchi, F., Giacca, M. and Proudfoot, N.J. (1998) Transcriptional interference perturbs the binding of Sp1 to the HIV-1 promoter. *Nucleic Acids Research*, **26**, 1294-1301.

Griffiths-Jones, S., Bateman, A., Marshall, M., Khanna, A. and Eddy, S.R. (2003) Rfam: an RNA family database. *Nucleic Acids Research*, **31**, 439-441.

Grundy, W.N., Bailey, T.L. and Elkan, C.P. (1997) Meta-MEME: Motif-based Hidden Markov Models of Protein Families. *Computer Applications in the Biosciences*, **13**, 397-406.

Guigo, R. (1997) Computational gene identification. *Journal of Molecular Medicine*, **75**, 389-393.

Guigo, R., Agarwal, P., Abril, J.F., Burset, M. and Fickett, J.W. (2000) An assessment of gene prediction accuracy in large DNA sequences. *Genome Research*, **10**, 1631-1642.

Guigo, R., Knudsen, S., Drake, N. and Smith, T. (1992) Prediction of gene structure. *J Mol Biol*, **226**, 141-157.

Harger, C., Chen, G., Farmer, A., Huang, W., Inman, J., Kiphart, D., Schilkey, F., Skupski, M.P. and Weller, J. (2000) The Genome Sequence DataBase. *Nucleic Acids Research*, **28**, 31-32.

Hartmuth, K. and Barta, A. (1988) Unusual branch point selection in processing of human growth hormone pre-mRNA. *Molecular and Cellular Biology*, **8**, 2011-2020.

Hastings, M.L. and Krainer, A.R. (2001) Functions of SR proteins in the U12-dependent AT-AC pre-mRNA splicing pathway. *RNA*, **7**, 471-482.

Hatzigeorgiou, A.G. (2002) Translation initiation start prediction in human cDNAs with high accuracy. *Bioinformatics*, **18**, 343-350.

He, X., Khan, A.U., Cheng, H., Pappas, D.L., Jr., Hampsey, M. and Moore, C.L. (2003) Functional interactions between the transcription and mRNA 3' end processing machineries mediated by Ssu72 and Sub1. *Genes and Development*, **17**, 1030-1042.

Hegyi, H. and Gerstein, M. (1999) The relationship between protein structure and function: a comprehensive survey with application to the yeast genome. *J Mol Biol*, **288**, 147-164.

Heinemann, M. and Wagner, R. (1997) Guanosine 3',5'-bis(diphosphate) (ppGpp)-dependent inhibition of transcription from stringently controlled Escherichia coli promoters can be explained by an altered initiation pathway that traps RNA polymerase. *European Journal of Biochemistry*, **247**, 990-999.

Henkin, T.M. (1996) Control of transcription termination in prokaryotes. *Annual Review of Genetics*, **30**, 35-57.

Hentze, M.W. (2001) Protein synthesis. Believe it or not-translation in the nucleus. *Science*, **293**, 1058-1059.

Hernandez, N. (1992) *Transcription of verterbrate snRNA genes and related genes.* Cold Spring Harbor Laboratory, New York.

Hillman, R.T., Green, R. and Brenner, S. (2004) An unappreciated role for RNA surveillance. *Genome Biology*, **5**, R8.

Hobohm, U. and Sander, C. (1994) Enlarged representative set of protein structures. *Protein Sci*, **3**, 522-524.

Holm, L. and Sander, C. (1993) Protein structure comparison by alignment of distance matrices. *J Mol Biol,* **233**, 123-138.

Holm, L. and Sander, C. (1996) Mapping the protein universe. *Science*, **273**, 595-603.

Houdebine, L.M. and Attal, J. (1999) Internal ribosome entry sites (IRESs): reality and use. *Transgenic Research*, **8**, 157-177.

Howe, K.L. (2003) Gene Prediction using a configurable system for the integration of data by dyanmic programming. *The Wellcome Trust Sanger Institute*. University of Cambridge, Cambridge, p. 209.

Howe, K.L., Chothia, T. and Durbin, R. (2002) GAZE: a generic framework for the integration of gene-prediction data by dynamic programming. *Genome Research*, **12**, 1418-1427.

Hua, S., Guo, T., Gough, J. and Sun, Z. (2002) Proteins with class alpha/beta fold have high-level participation in fusion events. *J Mol Biol*, **320**, 713-719.

Hubbard, T. (1994) Measuring distance between structural aligned residues. Cambridge. *Personal Communication*.

Iborra, F.J., Escargueil, A.E., Kwek, K.Y., Akoulitchev, A. and Cook, P.R. (2004) Molecular cross-talk between the transcription, translation, and nonsense-mediated decay machineries. *Journal of Cell Science*, **117**, 899-906.

Iborra, F.J., Jackson, D.A. and Cook, P.R. (2001) Coupled transcription and translation within nuclei of mammalian cells. *Science*, **293**, 1139-1142.

Ichiyanagi, K., Ishino, Y., Ariyoshi, M., Komori, K. and Morikawa, K. (2000) Crystal structure of an archaeal intein-encoded homing endonuclease PI-PfuI. *J Mol Biol*, **300**, 889-901.

Iida, Y. and Kanagu, D. (2000) Quantification analysis of translation initiation signal in vertebrate mRNAs: effect of nucleotides at positions +4(-)+6 upon efficiency of translation initiation. *Nucleic Acids Symposium Series*, 77-78.

Johnson, M.R., Norman, C., Reeve, M.A., Scully, J. and Proudfoot, N.J. (1986) Tripartite sequences within and 3' to the sea urchin H2A histone gene display properties associated with a transcriptional termination process. *Molecular and Cellular Biology*, **6**, 4008-4018.

Jurica, M.S. and Moore, M.J. (2003) Pre-mRNA splicing: awash in a sea of proteins. *Molecular Cell*, **12**, 5-14.

Jurica, M.S. and Stoddard, B.L. (1999) Homing endonucleases: structure, function and evolution. *Cell Mol Life Sci*, **55**, 1304-1326.

Kisselev, L.L. and Frolova, L. (1995) Termination of translation in eukaryotes. *Biochemistry and Cell Biology*, **73**, 1079-1086.

Kiyosawa, H., Yamanaka, I., Osato, N., Kondo, S. and Hayashizaki, Y. (2003) Antisense transcripts with FANTOM2 clone set and their implications for gene regulation. *Genome Research*, **13**, 1324-1334.

Kozak, M. (1987) An analysis of 5'-noncoding sequences from 699 vertebrate messenger RNAs. *Nucleic Acids Research*, **15**, 8125-8132.

Kozak, M. (1999) Initiation of translation in prokaryotes and eukaryotes. *Gene*, **234**, 187-208.

Kozak, M. (2001) New ways of initiating translation in eukaryotes? *Molecular and Cellular Biology*, **21**, 1899-1907.

Kozak, M. (2002) Pushing the limits of the scanning mechanism for initiation of translation. *Gene*, **299**, 1-34.

Krogh, A., Brown, M., Mian, I.S., Sjolander, K. and Haussler, D. (1994) Hidden Markov models in computational biology. Applications to protein modeling. *J Mol Biol*, **235**, 1501-1531.

Kwek, K.Y., Murphy, S., Furger, A., Thomas, B., O'Gorman, W., Kimura, H., Proudfoot, N.J. and Akoulitchev, A. (2002) U1 snRNA associates with TFIIH and regulates transcriptional initiation. *Nature Structural Biology*, **9**, 800-805.

Lander, E.S., Linton, L.M., Birren, B., Nusbaum, C., Zody, M.C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., Funke, R., Gage, D., Harris, K., Heaford, A., Howland, J., Kann, L., Lehoczky, J., LeVine, R., McEwan, P., McKernan, K., *et al.* (2001) Initial sequencing and analysis of the human genome. *Nature*, **409**, 860-921.

Lang, W.H., Morrow, B.E., Ju, Q., Warner, J.R. and Reeder, R.H. (1994) A model for transcription termination by RNA polymerase I. *Cell*, **79**, 527-534.

Lang, W.H. and Reeder, R.H. (1993) The REB1 site is an essential component of a terminator for RNA polymerase I in Saccharomyces cerevisiae. *Molecular and Cellular Biology*, **13**, 649-658.

Lanzotti, D.J., Kaygun, H., Yang, X., Duronio, R.J. and Marzluff, W.F. (2002) Developmental control of histone mRNA and dSLBP synthesis during Drosophila embryogenesis and the role of dSLBP in histone mRNA 3' end processing in vivo. *Molecular and Cellular Biology*, **22**, 2267-2282.

Le Hir, H., Nott, A. and Moore, M.J. (2003) How introns influence and enhance eukaryotic gene expression. *Trends in Biochemical Sciences*, **28**, 215-220.

Legendre, M. and Gautheret, D. (2003) Sequence determinants in human polyadenylation site selection. *BMC Genomics*, **4**, 7.

Levine, A. (2001a) Bioinformatics Approcahes to RNA splicing. *The Wellcome Trust Sanger Institute*. University of Cambridge, Cambridge, p. 74.

Levine, A. (2001b) http://www.sanger.ac.uk/Software/analysis/stratasplice/

Lewis, B.P., Green, R.E. and Brenner, S.E. (2003) Evidence for the widespread coupling of alternative splicing and nonsense-mediated mRNA decay in humans. *Proceedings of the National Academy of Sciences of the United States of America*, **100**, 189-192.

Liang, F., Holt, I., Pertea, G., Karamycheva, S., Salzberg, S.L. and Quackenbush, J. (2000) Gene index analysis of the human genome estimates approximately 120,000 genes. *Nature Genetics*, **25**, 239-240.

Lindahl, E. and Elofsson, A. (2000) Identification of related proteins on family, superfamily and fold level. *J Mol Biol*, **295**, 613-625.

Lin-Marq, N. and Clarkson, S.G. (1998) Efficient synthesis, termination and release of RNA polymerase III transcripts in Xenopus extracts depleted of La protein. *EMBO Journal*, **17**, 2033-2041.

Liu, H., Hao, H., Li, J. and Wong, L. (2003) A New Method to Predict Translation Initiation sites. *Proceedings - European Conference on Computational Biology*.

Logan, J., Falck-Pedersen, E., Darnell, J.E., Jr. and Shenk, T. (1987) A poly(A) addition site and a downstream termination region are required for efficient cessation of transcription by RNA polymerase II in the mouse beta maj-globin gene. *Proceedings of the National Academy of Sciences of the United States of America*, **84**, 8306-8310.

Lois, R., Freeman, L., Villeponteau, B. and Martinson, H.G. (1990) Active beta-globin gene transcription occurs in methylated, DNase I-resistant chromatin of nonerythroid chicken cells. *Molecular and Cellular Biology*, **10**, 16-27.

Lund, M., Tange, T.O., Dyhr-Mikkelsen, H., Hansen, J. and Kjems, J. (2000) Characterization of human RNA splice signals by iterative functional selection of splice sites. *RNA*, **6**, 528-544.

MacDonald, C.C. and Redondo, J.L. (2002) Reexamining the polyadenylation signal: were we wrong about AAUAAA? *Molecular and Cellular Endocrinology*, **190**, 1-8.

Mackay, D.J.C. (2003) *Information Theory, Inference and Learning algorithms*. Cambridge University Press, Cambridge.

Mangalam, H. (2002) The Bio* toolkits--a brief overview. *Brief Bioinform*, **3**, 296-302.

Manley, J.L. (2002) Nuclear coupling: RNA processing reaches back to transcription. *Nature Structural Biology*, **9**, 790-791.

Maraia, R.J., Kenan, D.J. and Keene, J.D. (1994) Eukaryotic transcription termination factor La mediates transcript release and facilitates reinitiation by RNA polymerase III. *Molecular and Cellular Biology*, **14**, 2147-2158.

Marshall, N.F. and Price, D.H. (1992) Control of formation of two distinct classes of RNA polymerase II elongation complexes. *Molecular and Cellular Biology*, **12**, 2078-2090.

Martens, J.A. (2003) Expression of an intergenic RNA represses transcription of the adjacent gene. New York. *Personal Communication*.

Mathe, C., Sagot, M.F., Schiex, T. and Rouze, P. (2002) Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Research*, **30**, 4103-4117.

Mathews, D.H., Sabina, J., Zuker, M. and Turner, D.H. (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure1, *1. *Journal of Molecular Biology*, **288**, 911-940.

Matsumoto, K., Wassarman, K.M. and Wolffe, A.P. (1998) Nuclear history of a pre-mRNA determines the translational activity of cytoplasmic mRNA. *EMBO Journal*, **17**, 2107-2121.

Mattick, J.S. (1994) Introns: evolution and function. *Current Opinion in Genetics and Development*, **4**, 823-831.

Mattick, J.S. (2001) Non-coding RNAs: the architects of eukaryotic complexity. *EMBO Reports*, **2**, 986-991.

McCullagh, P. and Nelder, J.A. (1983) *Generalized Linear Models*. Chapman and Hall, London.

Mechti, N., Piechaczyk, M., Blanchard, J.M., Jeanteur, P. and Lebleu, B. (1991) Sequence requirements for premature transcription arrest within the first intron of the mouse c-fos gene. *Molecular and Cellular Biology*, **11**, 2832-2841.

Meinhart, A., Silberzahn, T. and Cramer, P. (2003) The mRNA transcription/processing factor Ssu72 is a potential tyrosine phosphatase. *Journal of Biological Chemistry*, **278**, 15917-15921.

Micklem, G. http://www.rfcgr.mrc.ac.uk/Software/EMBOSS/Apps/cpgreport.html

Morea, V. (2001) Structural analysis of P-loops proteins. Cambridge. *Personal Communication*.

Moreira, A., Takagaki, Y., Brackenridge, S., Wollerton, M., Manley, J.L. and Proudfoot, N.J. (1998) The upstream sequence element of the C2 complement poly(A) signal activates mRNA 3' end formation by two distinct mechanisms. *Genes and Development*, **12**, 2522-2534.

Mount, S.M. (1982) A catalogue of splice junction sequences. *Nucleic Acids Research*, **10**, 459-472.

Murzin, A.G., Brenner, S.E., Hubbard, T. and Chothia, C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, **247**, 536-540.

Namy, O., Hatin, I. and Rousset, J.P. (2001) Impact of the six nucleotides downstream of the stop codon on translation termination. *EMBO Reports*, **2**, 787-793.

NCBI. http://www.ncbi.nlm.nih.gov

Neugebauer, K.M. (2002) On the importance of being co-transcriptional. *Journal of Cell Science*, **115**, 3865-3871.

Nussinov, R. (1978) Algorithms for loop matching. *Journal of Applied Mathematics*, **35**, 68-92.

Nussinov, R. (1986a) Sequence signals which may be required for efficient formation of mRNA 3' termini. *Nucleic Acids Research*, **14**, 3557-3571.

Nussinov, R. (1986b) TGTG, G clustering and other signals near non-mammalian vertebrate mRNA 3' termini: some implications. *Journal of Biomolecular Structure and Dynamics*, **3**, 1145-1153.

Nussinov, R. (1987) Asymmetry in the distributions of the four nucleotides at mRNA initiation and 3' termini sites: some geometrical implications. *Biochimica et Biophysica Acta*, **908**, 143-149.

Nussinov, R. (1990) Sequence signals in eukaryotic upstream regions. *Critical Reviews in Biochemistry and Molecular Biology*, **25**, 185-224.

Ohrt, M. (2004) http://www.phpinsider.com/php/code/pmatch/

Orengo, C.A., Bray, J.E., Buchan, D.W., Harrison, A., Lee, D., Pearl, F.M., Sillitoe, I., Todd, A.E. and Thornton, J.M. (2002) The CATH protein family database: a resource for structural and functional annotation of genomes. *Proteomics*, **2**, 11-21.

Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B. and Thornton, J.M. (1997) CATH--a hierarchic classification of protein domain structures. *Structure*, **5**, 1093-1108.

Orozco, I.J., Kim, S.J. and Martinson, H.G. (2002) The Poly(A) Signal, without the Assistance of Any Downstream Element, Directs RNA Polymerase II to Pause in

Vivo and Then to Release Stochastically from the Template. *Journal of Biological Chemistry*, **277**, 42899-42911.

Orphanides, G. and Reinberg, D. (2002) A unified theory of gene expression. *Cell*, **108**, 439-451.

Osheim, Y.N., Proudfoot, N.J. and Beyer, A.L. (1999) EM visualization of transcription by RNA polymerase II: downstream termination requires a poly(A) signal but not transcript cleavage. *Molecular Cell*, **3**, 379-387.

Osheim, Y.N., Sikes, M.L. and Beyer, A.L. (2002) EM visualization of Pol II genes in Drosophila: most genes terminate without prior 3' end cleavage of nascent transcripts. *Chromosoma*, **111**, 1-12.

Ozawa, Y., Hanaoka, S., Saito, R., Washio, T., Nakano, S., Shinagawa, A., Itoh, M., Shibata, K., Carninci, P. and Konno, H. (2002) Comprehensive sequence analysis of translation termination sites in various eukaryotes. *Gene*, **300**, 79-87.

Pain, V.M. (1996) Initiation of protein synthesis in eukaryotic cells. *European Journal of Biochemistry*, **236**, 747-771.

Palm, G.J., Billy, E., Filipowicz, W. and Wlodawer, A. (2000) Crystal structure of RNA 3'-terminal phosphate cyclase, a ubiquitous enzyme with unusual topology. *Structure Fold Des*, **8**, 13-23.

Pan, Q. and Simpson, R.U. (1999) c-myc intron element-binding proteins are required for 1, 25-dihydroxyvitamin D3 regulation of c-myc during HL-60 cell differentiation and the involvement of HOXB4. *Journal of Biological Chemistry*, **274**, 8437-8444.

Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T. and Chothia, C. (1998) Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J Mol Biol*, **284**, 1201-1210.

Park, J., Teichmann, S.A., Hubbard, T. and Chothia, C. (1997) Intermediate sequences increase the detection of homology between sequences. *J Mol Biol*, **273**, 349-354.

Parra, G., Agarwal, P., Abril, J.F., Wiehe, T., Fickett, J.W. and Guigo, R. (2003) Comparative gene prediction in human and mouse. *Genome Research*, **13**, 108-117.

Pearson, W.R. (1991) Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. *Genomics*, **11**, 635-650.

Pearson, W.R. (1995) Comparison of methods for searching protein sequence databases. *Protein Sci*, **4**, 1145-1160.

Pearson, W.R. and Lipman, D.J. (1988) Improved tools for biological sequence comparison. *Proc Natl Acad Sci U S A*, **85**, 2444-2448.

Pedersen, A.G. and Nielsen, H. (1997a) http://www.cbs.dtu.dk/services/NetStart

Pedersen, A.G. and Nielsen, H. (1997b) Neural network prediction of translation initiation sites in eukaryotes: perspectives for EST and genome analysis. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, **5**, 226-233.

Pertea, M. (2001) http://www.tigr.org/tdb/GeneSplicer/index.shtml

Pertea, M., Lin, X. and Salzberg, S.L. (2001) GeneSplicer: a new computational method for splice site prediction. *Nucleic Acids Research*, **29**, 1185-1190.

Pesole, G., Grillo, G. and Liuni, S. (1996) Databases of mRNA untranslated regions for metazoa. *Computers and Chemistry*, **20**, 141-144.

Pocock, M.R. (2003) Computational Analysis of Genomes. *The Wellcome Trust Sanger Institute*. University of Cambridge, Cambridge, p. 170.

Pocock, M.R., Down, T. and Hubbard, T. (2000) BioJava: open source components for bioinformatics. *ACM SIGBIO Newsletter*, **20**, 10-12.

Poland, B.W., Xu, M.Q. and Quiocho, F.A. (2000) Structural insights into the protein splicing mechanism of PI-SceI. *Journal of Biological Chemistry*, **275**, 16408-16413.

Proudfoot, N.J. (1989) How RNA polymerase II terminates transcription in higher eukaryotes. *Trends in Biochemical Sciences*, **14**, 105-110.

Proudfoot, N.J., Furger, A. and Dye, M.J. (2002) Integrating mRNA processing with transcription. *Cell*, **108**, 501-512.

Pruitt, K.D. and Maglott, D.R. (2001) RefSeq and LocusLink: NCBI gene-centered resources. *Nucleic Acids Research*, **29**, 137-140.

Ptitsyn, O.B. (1998) Protein folding: nucleation and compact intermediates. *Biochemistry (Mosc)*, **63**, 367-373.

Rappsilber, J., Ajuh, P., Lamond, A.I. and Mann, M. (2001) SPF30 is an essential human splicing factor required for assembly of the U4/U5/U6 tri-small nuclear ribonucleoprotein into the spliceosome. *Journal of Biological Chemistry*, **276**, 31142-31150.

Reed, R. (2000) Mechanisms of fidelity in pre-mRNA splicing. *Current Opinion in Cell Biology*, **12**, 340-345.

Reeder, R.H. and Lang, W.H. (1997) Terminating transcription in eukaryotes: lessons learned from RNA polymerase I. *Trends in Biochemical Sciences*, **22**, 473-477.

Reese, M.G., Kulp, D., Tammana, H. and Haussler, D. (2000) Genie--gene finding in Drosophila melanogaster. *Genome Research*, **10**, 529-538.

Reines, D. (1994) Transcription: Mechanisms and Regulation. In Conaway, J.W. and Conaway, R.C. (eds.), *Transcription: Mechanisms and Regulation*. Raven Press, New York, pp. 263-278.

Renner, D.B., Yamaguchi, Y., Wada, T., Handa, H. and Price, D.H. (2001) A highly purified RNA polymerase II elongation control system. *Journal of Biological Chemistry*, **276**, 42601-42609.

Rivas, E. and Eddy, S.R. (2001) Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics*, **2**, 8.

Rivas, E., Klein, R.J., Jones, T.A. and Eddy, S.R. (2001) Computational identification of noncoding RNAs in E. coli by comparative genomics. *Current Biology*, **11**, 1369-1373.

Roberts, G.C., Gooding, C., Mak, H.Y., Proudfoot, N.J. and Smith, C.W. (1998) Co-transcriptional commitment to alternative splice site selection. *Nucleic Acids Research*, **26**, 5568-5572.

Roberts, S. and Bentley, D.L. (1992) Distinct modes of transcription read through or terminate at the c-myc attenuator. *EMBO Journal*, **11**, 1085-1093.

ROC-Curve. http://gim.unmc.edu/dxtests/ROC1.htm

Rodriguez, C.R., Cho, E.J., Keogh, M.C., Moore, C.L., Greenleaf, A.L. and Buratowski, S. (2000) Kin28, the TFIIH-associated carboxy-terminal domain kinase, facilitates the recruitment of mRNA processing machinery to RNA polymerase II. *Molecular and Cellular Biology*, **20**, 104-112.

Rogic, S., Mackworth, A.K. and Ouellette, F.B. (2001) Evaluation of gene-finding programs on mammalian sequences. *Genome Research*, **11**, 817-832.

Rogozin, I.B., Kochetov, A.V., Kondrashov, F.A., Koonin, E.V. and Milanesi, L. (2001) Presence of ATG triplets in 5' untranslated regions of eukaryotic cDNAs correlates with a 'weak' context of the start codon. *Bioinformatics*, **17**, 890-900.

Russell, R.B. (1994) Domain insertion. *Protein Eng*, **7**, 1407-1410.

Salamov, A.A., Nishikawa, T. and Swindells, M.B. (1998a) Assessing protein coding region integrity in cDNA sequencing projects. *Bioinformatics*, **14**, 384-390.

Salamov, A.A., Nishikawa, T. and Swindells, M.B. (1998b) http://www.hri.co.jp/atgpr/

Salamov, A.A. and Solovyev, V.V. (2000) Ab initio gene finding in Drosophila genomic DNA. *Genome Research*, **10**, 516-522.

Salamov, A.A., Suwa, M., Orengo, C.A. and Swindells, M.B. (1999) Combining sensitive database searches with multiple intermediates to detect distant homologues. *Protein Eng*, **12**, 95-100.

Salzberg, S., Delcher, A.L., Fasman, K.H. and Henderson, J. (1998) A decision tree system for finding genes in DNA. *Journal of Computational Biology*, **5**, 667-680.

Salzberg, S.L. (1997) A method for identifying splice sites and translational start sites in eukaryotic mRNA. *Computer Applications in the Biosciences*, **13**, 365-376.

Sauder, J.M., Arthur, J.W. and Dunbrack, R.L., Jr. (2000) Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins*, **40**, 6-22.

Schneider, C., Will, C.L., Makarova, O.V., Makarov, E.M. and Luhrmann, R. (2002) Human U4/U6.U5 and U4atac/U6atac.U5 tri-snRNPs exhibit similar protein compositions. *Molecular and Cellular Biology*, **22**, 3219-3229.

Scholkopf, C., Burges, C. and Smola, A.J. (1999) *Advances in kernel methods - Support Vector Learning*. MIT Press, Cambridge.

Schwer, B. (2001) A new twist on RNA helicases: DExH/D box proteins as RNPases. *Nature Structural Biology*, **8**, 113-116.

Sheets, M.D., Ogg, S.C. and Wickens, M.P. (1990) Point mutations in AAUAAA and the poly (A) addition site: effects on the accuracy and efficiency of cleavage and polyadenylation in vitro. *Nucleic Acids Research*, **18**, 5799-5805.

Shindyalov, I.N. and Bourne, P.E. (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*, **11**, 739-747.

Slater, G. http://www.ebi.ac.uk/~guy/exonerate/exonerate.man.1.html

Smit, A.F.A. and Green, P. (1996) http://www.repeatmasker.org

Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences. *J Mol Biol*, **147**, 195-197.

Snyder, E.E. and Stormo, G.D. (1995) Identification of protein coding regions in genomic DNA. *Journal of Molecular Biology*, **248**, 1-18.

Solovyev, V. and Salamov, A. (1997) The Gene-Finder computer tools for analysis of human and model organisms genome sequences. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, **5**, 294-302.

Solovyev, V.V., Salamov, A.A. and Lawrence, C.B. (1995) Identification of human gene structure using linear discriminant functions and dynamic programming. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, **3**, 367-375.

Sonenberg, N. and Dever, T.E. (2003) Eukaryotic translation initiation factors and regulators. *Current Opinion in Structural Biology*, **13**, 56-63.

Spencer, C.A. and Groudine, M. (1990) Transcription elongation and eukaryotic gene regulation. *Oncogene*, **5**, 777-785.

Staden, R. (1984) Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Research*, **12**, 505-519.

Stajich, J.E., Block, D., Boulez, K., Brenner, S.E., Chervitz, S.A., Dagdigian, C., Fuellen, G., Gilbert, J.G., Korf, I., Lapp, H., Lehvaslaiho, H., Matsalla, C., Mungall, C.J., Osborne, B.I., Pocock, M.R., Schattner, P., Senger, M., Stein, L.D., Stupka, E., Wilkinson, M.D., *et al.* (2002) The Bioperl toolkit: Perl modules for the life sciences. *Genome Research*, **12**, 1611-1618.

Steinmetz, E.J., Conrad, N.K., Brow, D.A. and Corden, J.L. (2001) RNA-binding protein Nrd1 directs poly(A)-independent 3'-end formation of RNA polymerase II transcripts. *Nature*, **413**, 327-331.

Sternberg. (2001) Clustering can be used to assign function to unknown protein. Cambridge. *Personal Communication*.

Sun, Z.W. and Hampsey, M. (1996) Synthetic enhancement of a TFIIB defect by a mutation in SSU72, an essential yeast gene encoding a novel protein that affects transcription start site selection in vivo. *Molecular and Cellular Biology*, **16**, 1557-1566.

Tabaska, J.E. and Zhang, M.Q. (1999) Detection of polyadenylation signals in human DNA sequences. *Gene*, **231**, 77-86.

Takagaki, Y. and Manley, J.L. (1998) Levels of polyadenylation factor CstF-64 control IgM heavy chain mRNA accumulation and other events associated with B cell differentiation. *Molecular Cell*, **2**, 761-771.

Takagaki, Y., Seipelt, R.L., Peterson, M.L. and Manley, J.L. (1996) The polyadenylation factor CstF-64 regulates alternative processing of IgM heavy chain pre-mRNA during B cell differentiation. *Cell*, **87**, 941-952.

Talerico, M. and Berget, S.M. (1994) Intron definition in splicing of small Drosophila introns. *Molecular and Cellular Biology*, **14**, 3434-3445.

Tantravahi, J., Alvira, M. and Falck-Pedersen, E. (1993) Characterization of the mouse beta maj globin transcription termination region: a spacing sequence is required between the poly(A) signal sequence and multiple downstream termination elements. *Molecular and Cellular Biology*, **13**, 578-587.

Tarn, W.Y. and Steitz, J.A. (1996) A novel spliceosome containing U11, U12, and U5 snRNPs excises a minor class (AT-AC) intron in vitro. *Cell*, **84**, 801-811.

Tarn, W.Y. and Steitz, J.A. (1997) Pre-mRNA splicing: the discovery of a new spliceosome doubles the challenge. *Trends in Biochemical Sciences*, **22**, 132-137.

Teichmann, S.A., Park, J. and Chothia, C. (1998) Structural assignments to the Mycoplasma genitalium proteins show extensive gene duplications and domain rearrangements. *Proc Natl Acad Sci U S A*, **95**, 14658-14663.

Thanaraj, T.A. (2000) Positional characterisation of false positives from computational prediction of human splice sites. *Nucleic Acids Research*, **28**, 744-754.

Thompson, J.D., Higgins, D.G. and Gibson, T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, **22**, 4673-4680.

Tipping, M.E. (2001a) The Relevance Vector Machine. *Advances in Neural Information Processing Systems*, **12**, 652-658.

Tipping, M.E. (2001b) Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, **1**, 211-244.

Uberbacher, E.C., Xu, Y. and Mural, R.J. (1996) Discovering and understanding genes in human DNA sequence using GRAIL. *Methods in Enzymology*, **266**, 259-281.

Vagner, S., Vagner, C. and Mattaj, I.W. (2000) The carboxyl terminus of vertebrate poly(A) polymerase interacts with U2AF 65 to couple 3'-end processing and splicing. *Genes and Development*, **14**, 403-413.

Van Agtmael, T., Forrest, S.M., Del-Favero, J., Van Broeckhoven, C. and Williamson, R. (2003) Parametric and nonparametric genome scan analyses for human handedness. *European Journal of Human Genetics*, **11**, 779-783.

Vapnik, V.N. (1995) *The nature of Statistical Learning Theory*. Springer-Verlag, New York.

Wagner, R. (2000) *Transcription Regulation in Prokaryotes*. Oxford University Press, Oxford.

Watson, R.J. (1988) A transcriptional arrest mechanism involved in controlling constitutive levels of mouse c-myb mRNA. *Oncogene*, **2**, 267-272.

Westhead, D.R., Slidel, T.W., Flores, T.P. and Thornton, J.M. (1999) Protein structural topology: Automated analysis and diagrammatic representation. *Protein Sci*, **8**, 897-904.

Wetlaufer, D.B. (1973) Nucleation, rapid folding, and globular intrachain regions in proteins. *Proc Natl Acad Sci U S A*, **70**, 697-701.

Wheeler, D.L., Church, D.M., Edgar, R., Federhen, S., Helmberg, W., Madden, T.L., Pontius, J.U., Schuler, G.D., Schriml, L.M., Sequeira, E., Suzek, T.O., Tatusova, T.A. and Wagner, L. (2004) Database resources of the National Center for Biotechnology Information: update. *Nucleic Acids Research*, **32**, D35-40.

Wiest, D.K., Wang, D. and Hawley, D.K. (1992) Mechanistic studies of transcription arrest at the adenovirus major late attenuation site. Comparison of purified RNA polymerase II and washed elongation complexes. *Journal of Biological Chemistry*, **267**, 7733-7744.

Will, C.L., Schneider, C., MacMillan, A.M., Katopodis, N.F., Neubauer, G., Wilm, M., Luhrmann, R. and Query, C.C. (2001) A novel U2 and U11/U12 snRNP protein that associates with the pre-mRNA branch site. *EMBO Journal*, **20**, 4536-4546.

Will, C.L., Schneider, C., Reed, R. and Luhrmann, R. (1999) Identification of both shared and distinct proteins in the major and minor spliceosomes. *Science*, **284**, 2003-2005.

Wilson, C., Hilyer, L. and Green, P. (1990) http://cgap.nci.nih.gov/Genes/GeneFinder

Woychik, N.A. and Hampsey, M. (2002) The RNA polymerase II machinery: structure illuminates function. *Cell*, **108**, 453-463.

WTSI. http://www.sanger.ac.uk/Software/GFF

Yang, A.S. and Honig, B. (2000) An integrated approach to the analysis and modeling of protein sequences and structures. I. Protein structural alignment and a quantitative measure for protein structural distance. *J Mol Biol*, **301**, 665-678.

Yeung, G., Choi, L.M., Chao, L.C., Park, N.J., Liu, D., Jamil, A. and Martinson, H.G. (1998) Poly(A)-driven and poly(A)-assisted termination: two different modes of poly(A)-dependent transcription termination. *Molecular and Cellular Biology*, **18**, 276-289.

Yonaha, M. and Proudfoot, N.J. (1999) Specific transcriptional pausing activates polyadenylation in a coupled in vitro system. *Molecular Cell*, **3**, 593-600.

Yonaha, M. and Proudfoot, N.J. (2000) Transcriptional termination and coupled polyadenylation in vitro. *EMBO Journal*, **19**, 3770-3777.

Yoo, C.J. and Wolin, S.L. (1997) The yeast La protein is required for the 3' endonucleolytic cleavage that matures tRNA precursors. *Cell*, **89**, 393-402.

Zanier, K., Luyten, I., Crombie, C., Muller, B., Schumperli, D., Linge, J.P., Nilges, M. and Sattler, M. (2002) Structure of the histone mRNA hairpin required for cell cycle regulation of histone gene expression. *RNA*, **8**, 29-46.

Zarudnaya, M.I., Kolomiets, I.M., Potyahaylo, A.L. and Hovorun, D.M. (2003) Downstream elements of mammalian pre-mRNA polyadenylation signals: primary, secondary and higher-order structures. *Nucleic Acids Research*, **31**, 1375-1386.

Zeng, F., Yap, R.H. and Wong, L. (2002) Using feature generation and feature selection for accurate prediction of translation initiation sites. *Genome Inform Ser Workshop Genome Inform*, **13**, 192-200.

Zhang, M.Q. (1997) Identification of protein coding regions in the human genome by quadratic discriminant analysis. *Proceedings of the National Academy of Sciences of the United States of America*, **94**, 565-568.

Zhang, M.Q. (2002) Computational prediction of eukaryotic protein-coding genes. *Nature Review Genetics*, **3**, 698-709.

Zhang, M.Q. and Marr, T.G. (1993) A weight array method for splicing signal analysis. *Computer Applications in the Biosciences*, **9**, 499-509.

Zhang, R., Evans, G., Rotella, F.J., Westbrook, E.M., Beno, D., Huberman, E., Joachimiak, A. and Collart, F.R. (1999) Characteristics and crystal structure of bacterial inosine-5'-monophosphate dehydrogenase. *Biochemistry*, **38**, 4691-4700.

Zhang, X., Dong, G. and Wong, L. (2000) Using CAEP to Predict Translation Initiation Sites from Genomic DNA Sequences. *Manuscript*.

Zhao, J., Hyman, L. and Moore, C. (1999) Formation of mRNA 3' ends in eukaryotes: mechanism, regulation, and interrelationships with other steps in mRNA synthesis. *Microbiology and Molecular Biology Reviews*, **63**, 405-445.

Zien, A., Ratsch, G., Mika, S., Scholkopf, B., Lengauer, T. and Muller, K.R. (2000) Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, **16**, 799-807.

Zuker, M. (1994) Prediction of RNA secondary structure by energy minimization. *Methods in Molecular Biology*, **25**, 267-294.

Zuker, M. (2000) Calculating nucleic acid secondary structure. *Current Opinion in Structural Biology*, **10**, 303-310.

Zuker, M. (2003) Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, **31**, 3406-3415.

Zuker, M. and Stiegler, P. (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, **9**, 133-148.

APPENDIX A: DOMAIN INSERTION

A.1 Introduction

Taking advantage of an evolutionary basis of domain classification, here I describe the nature and characteristics of domain insertions in protein structures, a phenomenon that is different from the usual pattern of sequential arrangement of domains in multi-domain proteins.

Domains constitute the basic structural, functional and evolutionary unit of proteins (Holm and Sander, 1996; Murzin *et al.,* 1995; Orengo *et al.,* 1997). Proteins can comprise a single domain or a combination of domains. It is well established that multi-domain proteins with widely diversified architecture and functions are generated from a limited repertoire of domain families (Bork *et al.,* 1996; Chothia, 1992). Structural assignments to complete genomes revealed that almost two-thirds of prokaryotic proteins and 80% of eukaryotic proteins are multi-domain proteins (Teichmann *et al.,* 1998). In 1973, Donald Wetlaufer introduced the classification of domains into continuous and discontinuous (Wetlaufer, 1973). A continuous domain is formed by one part of a polypeptide chain, while a discontinuous domain is formed by two or more parts of a single polypeptide chain. Thus, discontinuous domains are essentially formed by one-dimensionally non-contiguous segments of a polypeptide. While most multi-domain proteins have continuous domains, some proteins exhibit non-contiguous arrangement of their domains (Wetlaufer, 1973). In this work, I focus on insertions (Russell, 1994), which are the cases of one domain being inserted into another domain (Figure 67).

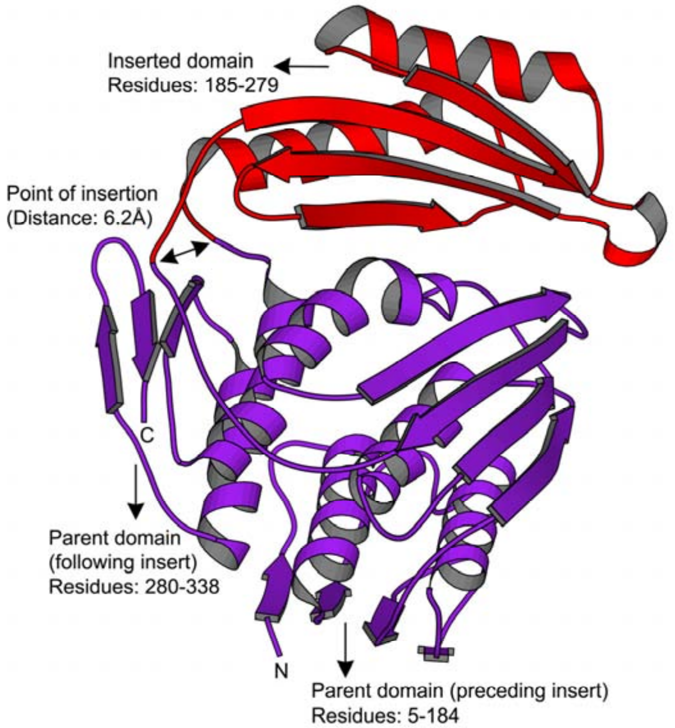


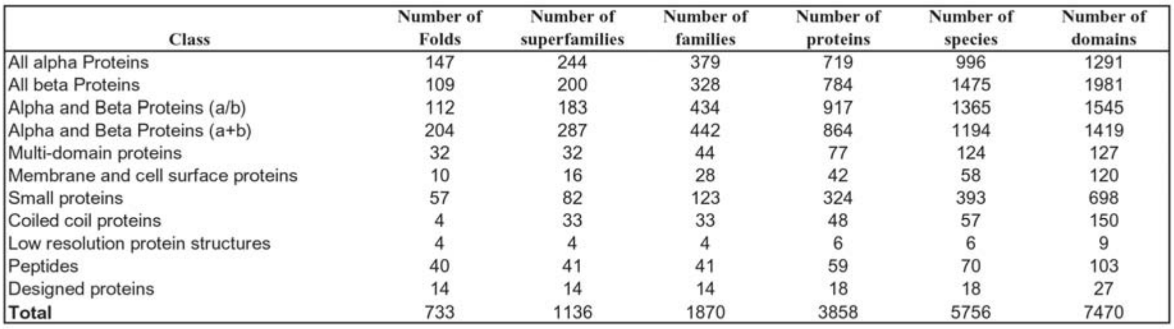
Figure 67. Domain insertion in Escherichia coli enzyme RNA 3'-terminal phosphate cyclase (PDB 1qmhA). The E. coli *enzyme RNA 3'-terminal phosphate cyclase consists of two domains, of which one is contained within the other. The parent domain (residues 5-184, 280-338, coloured purple) consists of three repeated folding units; each unit has two α-helices and a four-stranded β-sheet. The folding unit resembles the C-terminal domain of bacterial translation initiation factor 3 (IF3). Between an α-helix and a β-strand of the third IF3-like repeat of the parent domain, there is a smaller inserted domain (residues 185-279, coloured red). Although the inserted domain has the same secondary structural elements as the parent domain, it has different topology and a different fold. Insert resembles the fold observed in human thioredoxin.*

I followed the definition of protein domains in the Structural Classification Of Proteins (SCOP) database (version 1.61) (Murzin *et al.*, 1995). Although there are several available schemes of protein structure classification, I chose SCOP because it is a manually curated classification of protein structures based on their structural and evolutionary relationship. In SCOP, a protein domain is considered as a unit of evolution if it occurs independently or in combination with other domains.

SCOP represents a hierarchical classification scheme with four principal levels: family, superfamily, fold and class. Domains clustered into families are evolutionarily related and can be detected at the sequence level. Domains grouped into superfamilies can have low sequence identity but their structural and functional features suggest a common evolutionary origin. Superfamilies with similar topology are grouped under a fold. Folds are assigned to classes based on their secondary structure. For my analysis, I considered the fold and superfamily levels of SCOP hierarchy and the five major classes (all-α, all-β, α/β, α+β and 'small proteins'). All-α and all-β classes include proteins with abundant α-helices or β-sheets, respectively. The α/β class is distinguished mainly by parallel beta sheets (β-α-β units), whereas the α+β class contains proteins with predominantly anti-parallel beta sheets (segregated α and β regions). Small proteins are distinguished by their size rather than other features.

Data for this analysis was obtained from the Protein Data Bank (PDB) (Berman *et al.*, 2002). To overcome the redundancy inherent in PDB, I chose a pre-computed list of non-redundant protein chains provided by PDB_Select (April 2002 release obtained from *ftp://ftp.embl-heidelberg.de/pub/databases/protein_extras/pdb_select*) (Hobohm and Sander, 1994). I used the set of proteins that had pair-wise sequence identities less than 90% and designated this set as PDB_90. Out of the 6182 chains in PDB_90, only 5883 chains were assigned SCOP domain definitions, extracted from the SCOP parseable file *dir.cla.scop.txt_1.61*. Table 24 shows the distribution of SCOP folds, superfamilies, families and domains in each class for chains present in PDB_90.

Table 24. SCOP (1.61 release) classification statistics for chains in PDB_90 (April 2002 release)



Class	Number of Folds	Number of superfamilies	Number of families	Number of proteins	Number of species	Number of domains
All alpha Proteins	147	244	379	719	996	1291
All beta Proteins	109	200	328	784	1475	1981
Alpha and Beta Proteins (a/b)	112	183	434	917	1365	1545
Alpha and Beta Proteins (a+b)	204	287	442	864	1194	1419
Multi-domain proteins	32	32	44	77	124	127
Membrane and cell surface proteins	10	16	28	42	58	120
Small proteins	57	82	123	324	393	698
Coiled coil proteins	4	33	33	48	57	150
Low resolution protein structures	4	4	4	6	6	9
Peptides	40	41	41	59	70	103
Designed proteins	14	14	14	18	18	27
Total	733	1136	1870	3858	5756	7470

It is self-evident that insertions can only be found in multi-domain proteins, where one domain (insert) is contained within another domain (parent). Parent and insert domains can belong to the same or different SCOP superfamilies. Likewise, a combination of two domains can be viewed as a combination of superfamily combinations. I obtained a total of

140 proteins that conformed to this definition. When I considered the 140 pairs of parent-insert superfamily combinations, I observed that several pairs were identical. Whenever there was also the same topological relationship between the parent and insert domains, I retained only one example of a pair of superfamily combinations. This procedure left 40 unique parent-insert superfamily combinations. Variations on the simple scheme ‘one insert within one parent’ were present; they are shown in Figure 68.

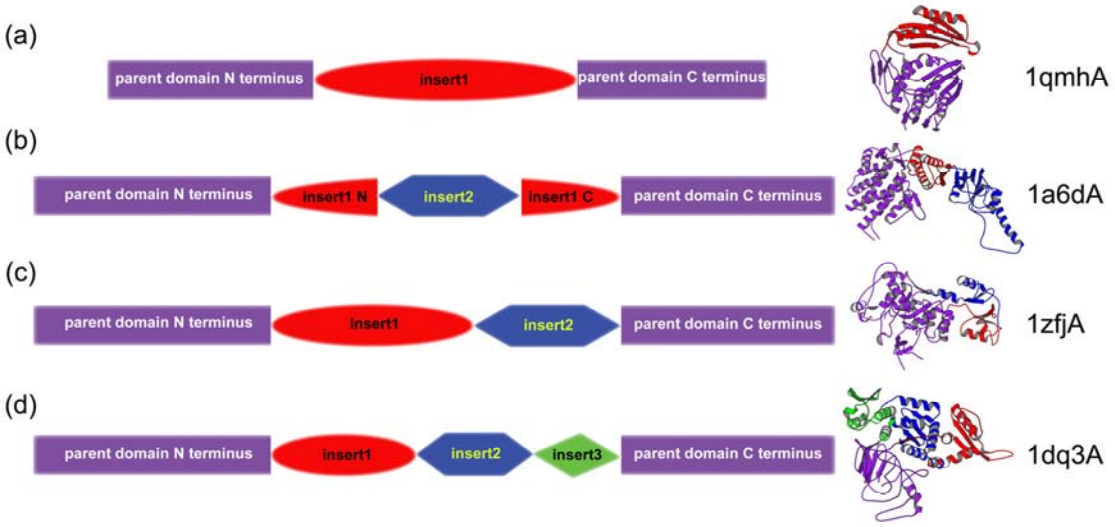


Figure 68. Schematic representation of types of domain insertions observed in protein structures. (a) Single insertion (e.g., 1qmhA). (b) Nested insertion (e.g., 1a6dA). 'insert1 N' and 'insert1 C' represent the N- and C-terminus of insert, respectively. (c) Two-domain insertion (e.g., 1zfjA). (d) Three-domain insertion (e.g., 1dq3A).

For all cases of identified domain insertions, I checked for artefacts arising from missing coordinates. This was necessary because SCOP domain definitions are based on atomic coordinates provided in PDB. To ascertain consistency, I compared atomic coordinates (ATOM records) *versus* sequences (SEQRES records) that were obtained from the ASTRAL compendium (Chandonia *et al.*, 2002). In the majority of cases, sequences were completely covered by coordinates, but in other cases, there were parts of sequences with missing coordinates. However, in none of the latter cases did the absent coordinates obscure the position of inserts.

I then calculated unique superfamily combinations for all multi-domain proteins and found 450 unique superfamily combinations for 5883 single or multi-domain proteins in SCOP. Thus, domain insertions constitute 9% (40/450) of all unique superfamily occurrences.

A.2 Types of domain insertions

Domain insertions can be categorized as either single or multiple depending on the number of inserts (Figure 68). In single insertions, one domain is inserted into another domain, and both domains can belong to the same or different superfamilies. For example, in Figure 68a, the *Escherichia coli* enzyme RNA 3'-terminal phosphate cyclase (PDB: 1qmhA, Palm *et al.,* 2000) has two domains, a small insert and a larger parent that belong to different superfamilies. Close to 90% (36/40) of observed insertions are single insertions. In multiple insertions, more than one domain, either of the same or different superfamily, is inserted into the parent domain. I observed three types of multiple insertions (i) Nested insertions: In *Thermoplasma acidophilum* thermosome (PDB: 1a6dA, Ditzel *et al.,* 1998), the archael chaperonin, the apical domain is inserted into the intermediate domain, which is in turn inserted into an ATPase domain (ii) Two-domain insertions: The type II inosine monophosphate dehydrogenase from *Streptococcus pyogenes* (PDB: 1zfjA, Zhang *et al.,* 1999) contains two tandem cystathionine-β-synthase domains inserted into the catalytic TIM-barrel domain. The second example is the *Saccharomyces cerevisiae* PI-*Sce*I intein (PDB: 1ef0A, Poland *et al.,* 2000), a homing endonuclease with protein splicing activity, which has the duplicated endonuclease domain inserted into the Hint domain (iii) Three-domain insertions: In PI-*Pfu*I, an intein-encoded homing endonuclease from the archaebacteria *Pyrococcus furiosus* (PDB: 1dq3A, Ichiyanagi *et al.*, 2000), the Hint domain has three tandem inserts, two intein endonuclease domains with αββαββαα structural motifs, and one Stirrup domain.

Previous work on intron-encoded homing endonucleases, from the dodecapeptide family, showed that for their folding, dimerisation and catalysis, they should form a dimer that has two copies of the LAGLIDADG motif (one copy per subunit of a dimer), or alternatively they could be monomeric if a monomer has both copies of the motif (Jurica and Stoddard, 1999). I found that in PI-*Sce*I (case [ii] above) and PI-*Pfu*I (case [iii] above), two monomeric domains were tandemly inserted into one parent domain. The previous observation that motifs are only functional as a dimer suggests that during the course of evolution, there was a simultaneous insertion of two monomeric domains into the parent domain, rather than an insertion of one monomeric domain followed by its duplication.

In this analysis, I treated multiple insertions as several separate parent-insert combinations, resulting in the total of 45 such combinations within 40 protein chains. There were 41 unique parent-insert superfamily combinations. Upon examination of relationships among proteins containing insertions, levels of SCOP hierarchy, and superfamily participation of parent and inserted domains, I identified several biologically meaningful patterns. These findings are discussed below.

A.3 Nature and characteristics of domain insertions: Class level

As mentioned before, I considered five SCOP classes, leading to a maximum of 25 (5*5) pair-wise combinations. From the data, I observed only 15 combinations when investigating class participation of parent-insert pairs. The combination of α/β-parent-α+β-insert was predominant, while 50% of all parents belonged to α/β class and 40% of all inserts belonged to α+β class. Domains from α/β class were parent domains, which were two and four fold more often than domains from all-β and all-α class respectively. Domains from the class of small proteins were seen only as inserts. This bias could be explained, at least to a certain extent, by taking into consideration the size and function of parents and inserts, which is discussed in the next section.

A.3.1 Size and function of domains involved in insertions

Figure 69a shows the domain length distribution for proteins from PDB_90 set across the five SCOP classes. The average domain length was longest for α/β class followed by the all-β, α+β, and all-α class. When I calculated distribution of average domain lengths for 41 parent domains, I observed the same trend (Figure 69b). However, the average length of parent domains was noticeably larger than the average length of domains from PDB_90 set; this was true for each SCOP class (compare Figure 69a and Figure 69b). Thus, combining the fact that α/β parent domains are the most abundant with the fact that α/β domains are the longest on average, I arrived at the explanation that longer domains more readily accept insertions during evolution. As for the inserted domains, α+β and all-α class were equal and major contributors to the number of domains. Therefore, the trend observed for parents is not applicable for inserts.

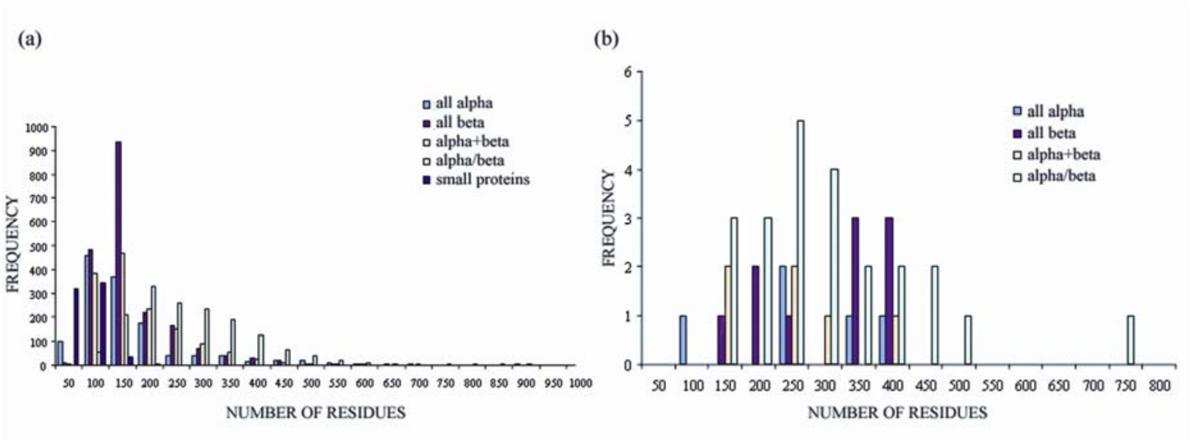


Figure 69. (a) Domain length distribution for all domains in the non-redundant set of proteins (PDB_90). (b) Domain length distribution for parent domains.

In most cases, inserted domains were shorter than parent domains. This is despite the fact that inserted domains could belong to SCOP classes with the longest average domain length (Figure 70a). Parents comprised 50-80% of protein length, while inserts comprised 20-50%. Close to 80% of inserts were shorter than 175 residues, which is the average length of a protein domain calculated from crystal structures (Gerstein, 1997). More than 60% of inserts were shorter than 130 residues. This observation is consistent with the heuristic logic that smaller domains are less likely to disturb the structure and folding of parent domains; it could explain short lengths of inserted domains. This explanation does not contradict an important experiment by Doi and colleagues (Doi *et al.*, 1997). They were able to show that when random sequences of 120-130 amino acid residues were inserted into a surface loop region of *Escherichia coli* RNase HI, about 10% of the clones retained >1% of the wild-type RNase HI activity (Doi *et al.*, 1997).

The high proportion of α/β class domains, as parents, can be correlated with their biochemical function. Previous work showed that more than a half of PDB families are enzymes and close to one half of all enzyme families contain multi-domain proteins. Multi-domain enzymes often consist of a catalytic domain and a nucleotide binding domain (Hegyi and Gerstein, 1999). It is therefore possible to predict that domain insertions are likely to occur in enzymes. Indeed, in the dataset, 39 out of 40 parent-insert pairs conform to this prediction. The remaining non-enzymatic protein is the bluetongue virus capsid protein vp-7, which has the central domain from all-β class inserted into the multi-helical parent domain. A genome-scale analysis of the structural features of proteins revealed that proteins with α/β fold are frequently involved in fusion events (Hua *et al.*, 2002). α/β folds are also known to be disproportionately associated with enzymatic function (Hegyi and Gerstein, 1999), which lends further credence to the prominent role of α/β folds in accepting insertions.

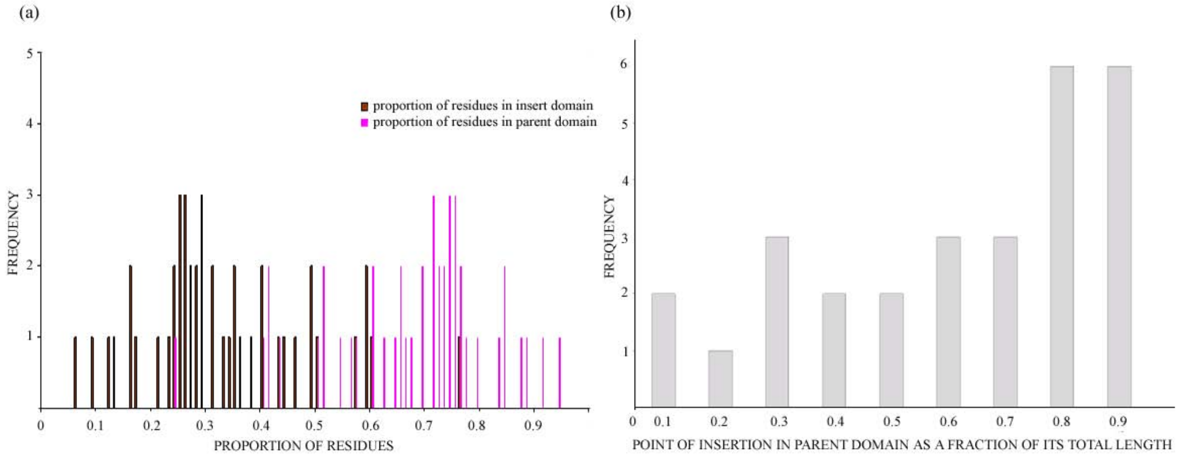


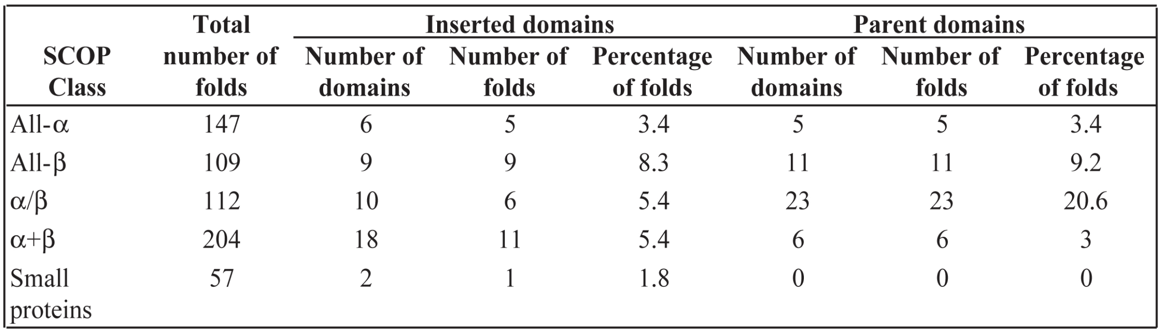
Figure 70. (a) Proportion of residues in parent and insert domains in parent-insert combinations. (b) Point of insertion in parent domain. Insert position is given as a fraction of total length of parent domain.

A.4 Nature and characteristics of domain insertions: Fold and superfamily level

Out of 57 folds in the class of small proteins, two domains with one fold (Rubredoxin fold) were found as inserts; both inserted domains belong to the same superfamily. Within the α+β class, the 18 inserted domains (from 15 superfamilies) spanned 11 folds; there are 204 different folds in the α+β class (Table 25). The trend was the same for the other SCOP classes, where folds of inserted domains constituted minor fractions of all known folds. In contrast to the inserts, all parent domains had different folds. Thus, I observed another distinction between parents and inserts at the fold level.

Similarly, parent superfamilies were found to be more versatile than insert superfamilies (most insert superfamilies combine with only one parent superfamily). There are merely 3 out of 45 insert superfamilies that combine with two different parent superfamilies. These insert superfamilies are NAD(P)-binding Rossmann superfamily, FAD/NAD(P)-binding superfamily and C-terminal domain of FAD-linked reductases superfamily.

Table 25. Distribution of inserted and parent domains at the SCOP class and fold level. The number of domains and the number of folds they come from is given for inserted and parent domains across the five different classes in the SCOP hierarchy. Percentage gives the number of folds contributing to insertions over total number of folds under the class.



SCOP Class	**Total number of folds**	**Inserted domains**			**Parent domains**		
		Number of domains	**Number of folds**	**Percentage of folds**	**Number of domains**	**Number of folds**	**Percentage of folds**
All-α	147	6	5	3.4	5	5	3.4
All-β	109	9	9	8.3	11	11	9.2
α/β	112	10	6	5.4	23	23	20.6
$\alpha+\beta$	204	18	11	5.4	6	6	3
Small proteins	57	2	1	1.8	0	0	0

While many parent superfamilies conservatively combine with one insert superfamily, there are conspicuous exceptions. There are three parent superfamilies each combining two different insert superfamilies. The three parent superfamilies in question are Zn-dependent exopepetidases superfamily, nucleotidyl transferase superfamily, and nucleotide-binding domain superfamily. Moreover, there are two parent superfamilies each combining with three different insert superfamilies. The two parent superfamilies are P-loop containing NTP hydrolases superfamily, and FAD/NAD(P)-binding domain superfamily.

Two further observations at the superfamily level are worth mentioning. Firstly, all parents and inserts belong to different superfamilies. There is only one exception: in *Escherichia coli* enzyme glutathione reductase (PDB: 1gesB), the parent and insert belong to the same superfamily of FAD/NAD(P)-binding domains. Secondly, superfamilies that are popular in the parent or insert context also appear to be popular in the sequential domain combination context (Apic *et al.*, 2001). They were found combining with more than one superfamily in the sequential domain order. One exception to this correlation is the superfamily of C-terminal domains of FAD-linked reductases; this superfamily is popular in the insert context, but does not tandemly combine with other superfamilies.

A.5 Point of insertion

I did not find any bias in the distribution of insertion points within 41 unique parent-insert combinations. However, a significant bias in the location of the insertion point was observed when I considered a subset of 28 parent-insert combinations, where either the parent or insert superfamily also participated in sequential combination with other superfamilies. As shown in Figure 70b, for the 28 cases in question, the insertion point occurred in the last third part of the parent domain sequence (confidence level 98%). Spatially, all 41 insertions were observed in loop regions of the 3D structure of parent domains.

Though it may not be feasible to provide a definitive explanation for the observation of bias towards C-terminus for insertion in the parent domain, an event in the N-terminus or the middle of the domain are likely to disrupt the gene structure and pose a problem during transcription or translation.

Also insertions in the C-terminus indicate most of the insertions seen in the database are not *strictly* insertions but normal sequential combinations with the second domain starting before the end of the first domain. This stem from the fact, C-terminus bias in insertion is found only in cases of parent-insert combinations, where either the parent or insert also occur in sequential combinations with other superfamilies. Further research on the domain insertions involving the core structure of the parent and insert domains can throw more light on this view.

A.6 Proximity of N- and C-termini in inserts

I wanted to determine how the insertion context affects the distance between N- and C-terminus of an inserted domain. The distance between termini was defined as the distance between C-alpha atoms of the first and the last residue of the domain. I first calculated distances for domains that do not participate in insertions. In order to do this, I considered 1000 domains, each representative of one SCOP superfamily. I obtained sequences and coordinates for the domains from the ASTRAL compendium (Chandonia *et al.*, 2002). Only 687 domain sequences were completely covered by coordinates. Using AEROSPACI scores (Chandonia *et al.*, 2002), I was able to find 60 substitutes for the 313 representative domains that were not entirely covered by coordinates. Altogether, I obtained complete coordinate information for 747 domains (687 + 60). Because I confined the analysis to five major SCOP classes, I calculated distances between termini for the 711 domains, which belong to the five classes being investigated. The average distance for representative domains was 25 Å.

Calculation of distances between the termini of inserted domains was less straightforward. Domain boundaries reported in SCOP are human defined. Therefore, I compared SCOP domain boundaries for 41 inserted domains against the domain boundaries reported in CATH database (Orengo *et al.*, 2002). In contrast to SCOP, CATH structural classification of proteins has been produced automatically. However, only 28 out of 41 inserted domains were available in CATH, whereas the other 13 have either differences in domain classification or the corresponding proteins were absent from CATH classification. For 28 inserted domains, boundaries were identical between SCOP and CATH. The average distance between domain termini of inserted domains was 8 Å (confidence level 99%), which is two-thirds shorter than the distance between termini in normal domains.

There are two superfamilies that occur in both parent and insert context. This example allowed me to compare distances between termini for a parent and an insert from the same superfamily. In case of FAD/NAD(P)-binding domain superfamily, the distances were 30 Å and 5 Å for parent and an insert, respectively. These figures were 11 Å and 8 Å for NAD-binding Rossmann domain superfamily. Thus, this analysis shows that the ends of inserted domains are significantly closer than ends of parent domains or domains not participating in insertions. However one must be cautious in interpreting the results as the N and C termini distances for the parent domain is not calculated for the core structure.

It is interesting to speculate how the distance between domain termini can affect stability and conformational flexibility of a protein domain. While insertion context might generally reduce conformational freedom of the domain, it can simultaneously contribute to the stability of the domain, which would in turn affect its function. One can also imagine how the close proximity of domain termini can restore protein conformational flexibility by mimicking an inter-domain link observed in sequentially ordered domains.

A.7 Conclusions

Utilising an evolutionary basis of domain classification, I described the nature and characteristics of domain insertions in protein structures. Domain insertions represent an unusual but abundant case of multi-domain proteins. This analysis gave several novel insights into the nature and characteristics of domain insertions.

(1) Close to 9% multi-domain proteins contain insertions.

(2) The majority of insertions are the single domain insertions. Also found there were two-domain, three-domain, and nested insertions in PDB.

(3) α/β class has a higher propensity to accept insertions. This could be correlated to the size and function of proteins within the class.

(4) Parent domains were found to be longer than the inserted domains in most cases.

(5) When fold and superfamily combinations were considered for parents and inserts, the former was found to be more versatile than the latter, in that the parent domains combined with more partners.

(6) The point of insertion is biased towards the C-terminus of parents whenever the parent domain belongs to the superfamily that sequentially combines with other superfamilies.

(7) Inserted domains have juxtaposed termini compared to parent domains.

Perhaps, domains are more viable in the insert context when their termini are close in space; small size can further contribute to their viability.

These results clearly indicate that despite the structural and functional constraints inherent in the process of domain insertion, this process is an effective way of creating multi-domain proteins. This description of the many features of domain insertions could be used in protein engineering for producing novel multi-functional fusion proteins. Betton and co-workers (Betton *et al.*, 1997) created hybrid proteins by inserting a penicillin-hydrolysing enzyme TEM beta-lactamase (Bla) into the maltodextrin-binding protein (MalE); they used the permissive insertion sites identified before (Duplay *et al.*, 1987). Two insertions resulted in the functional hybrids, one insertion occurred in the first quarter of the MalE protein, while the other occurred in the last quarter. The parent protein (MalE) belongs to the α/β class, and the authors experimentally showed the 5 Å distance between the termini of the inserted domain (Bla). Thus, there is recent experimental data that nicely fit into the picture of insertions found in natural multi-domain proteins.

APPENDIX B: PROTEIN EVOLUTION

B.1 Introduction

Divergence in structure and function of proteins is due to an evolutionary process driven by functional and environmental constraints. These constraints bring about changes in the protein sequence through mutations, insertions and deletions with the preservation of residues important for the structure and function of the protein (Chothia and Lesk, 1986). However, not all the sequence modifications are incorporated or maintained since some changes may be deleterious to the structure or function of the protein. Hence, the structural *'core'* (Chothia and Lesk, 1986) tends to be well conserved during evolution. When proteins evolve, the constraints on the protein structure are relaxed or rather replaced by new constraints and the sequence and structure can change more radically. These changes are generally slow processes and leave a trail of *homologs*. Homologs are proteins evolved from a common ancestor and their evolutionary relationship is evident from similarities in sequence, structure and function. Homologous proteins have been studied for a long time to understand their evolutionary relationships and to assign function or structure to new protein sequences. For homolog searches in the sequence databases, one needs an alignment algorithm, residue similarity matrix, scoring scheme and knowledge about scoring thresholds to identify true relationships.

Among the available pairwise alignment algorithms, one of the most sensitive is the Smith-Waterman algorithm (Smith and Waterman, 1981) adopted in the SSEARCH program (Pearson, 1991). Although this algorithm is more sensitive and rigorous, it is computationally expensive in comparison to FASTA (Pearson and Lipman, 1988) and BLAST (Altschul *et al.*, 1990). The speed and convenience of BLAST made it the most popular program, although it compromises sensitivity. FASTA ranks between these two programs and can be run in two modes: either at greater speed (ktup = 2) or greater accuracy (ktup =1). Pearson (Pearson, 1991, 1995) did a comparison of these three methods and showed that the Smith-Waterman algorithm worked slightly better than FASTA, which was in turn much more effective than BLAST.

Although pairwise comparison methods are a common way to find sequence homologs, they have difficulty in detecting remote homologs when sequence identity falls below 30% (Brenner *et al.*, 1998). Alternate methods like Profile Hidden Markov Models (Eddy, 1996; Krogh *et al.*, 1994), psi-BLAST (Altschul *et al.*, 1997) and Intermediate Sequence Search (Park *et al.*, 1997) reduce this limitation and increase sensitivity.

Intermediate Sequence Search (ISS) is a search technique, wherein two related sequences which cannot be detected directly by pairwise sequence comparison methods are matched using an intermediate sequence sharing close homology with the two distantly related sequences. This concept has been extended to include multiple intermediate sequences (MISS) between two distant sequences (Salamov *et al.*, 1999). The disadvantage with ISS is that the errors caused in the intermediate are likely to propagate as it is not dependent on multiple sequence alignment. Errors caused by ISS when comparing multi-domain protein sequences, can be avoided by splitting query sequence to individual domains. Figure 71 gives an overall idea on how different methods are exploring the sequence space (Lindahl and Elofsson, 2000).

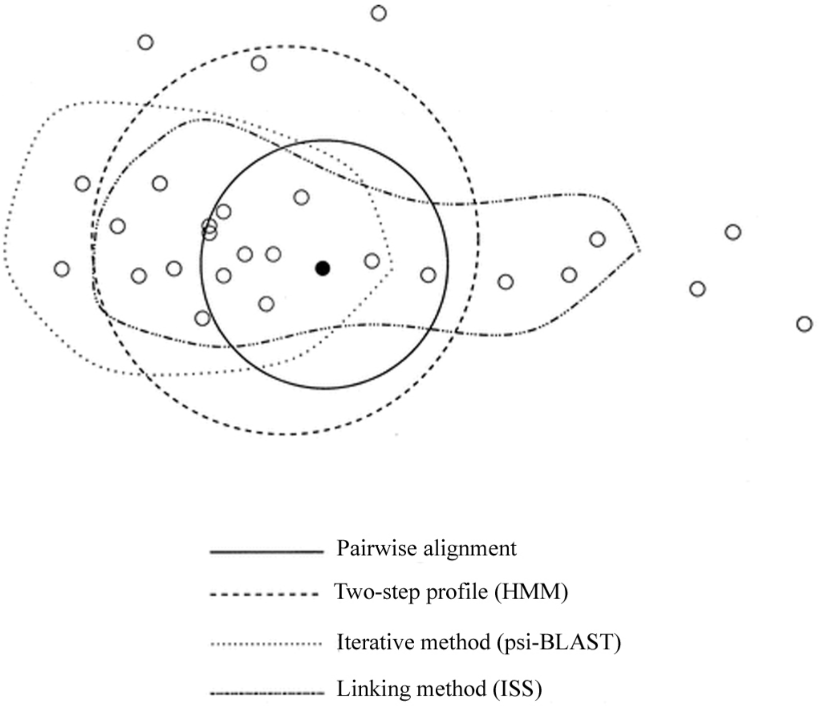


Figure 71. Schematic diagram showing performance of different sequence comparison methods. The filled circle represents the query sequence used in the database search and the open circles represent family members. The distance between two circles represents some arbitrary distance.

A comparison of these recent methods with pairwise sequence comparison methods, performed by searching remote homologs in a Structural Classification Of Proteins (SCOP, Murzin *et al.,* 1995) sequence database having less than 40% identity, show that ISS performs one and half times better than FASTA. In sequences with less than 30% identity, a HMM-based SAM-T98 and psi-BLAST detected three times more relationships than pairwise sequence comparison methods (Park *et al.,* 1998). Sauder *et al.* compared the quality of alignments produced by BLAST, psi-BLAST, ISS and ClustalW (Thompson *et al.,* 1994) with structural alignments. ISS produced longer alignments than psi-BLAST with nearly comparable per-residue alignment quality. At 10-15% identity, BLAST correctly aligned 28%, psi-BLAST 40% and ISS 46% of residues to the structural alignment (Sauder *et al.,* 2000).

All these results show that ISS performs as well as psi-BLAST in identifying distant homologs. However it is not yet clear how ISS is able to detect remote relationships. Moreover, I was interested to determine whether intermediates identified by ISS can provide any knowledge about protein evolution. This study tries to find answers to these questions.

To aid this objective, I also used structure comparisons to understand relationships between proteins. The degree of fitness between structures is usually calculated by a scoring scheme. The common way to represent the structural fitness is Root Mean Square Deviation (RMSD) for all residues of the two protein structures. The RMSD gives a measure of the average level of deviations over the superposed atoms.

$$\sqrt{\sum_{i=1}^{n} \frac{D_i^2}{N}}$$

Where, *D* refers to deviation of the atoms and *N* refers to the number of atoms matched.

There are different structural alignment methods adopting the aforementioned algorithms. Amongst the common implementations are DALI (Holm and Sander, 1993), Combinatorial Extension (CE) (Shindyalov and Bourne, 1998), and Protein Informatics System for Modelling (PrISM) (Yang and Honig, 2000). Here, I used PrISM to compare the structures.

Protein evolution may occur in two ways: divergent or convergent evolution. When a protein structure diverges to form a new fold or function, it results in divergent evolution (e.g., P-loops). However if two evolutionarily independent folds converge to represent similar structure or function it becomes convergent evolution (e.g., serine proteases). Proteins evolved through a divergent mechanism are likely to have a trail of homologs and can be detected using sequence and structure comparisons. Here, I attempt to study this using two well known protein families – *Cytochrome c* and *P-loops* and answer the following questions.

(1) Is it possible to understand the evolutionary pattern of any protein family or superfamily based solely on its structure and sequence divergence?

(2) Whether understanding this will help us in assigning hierarchies for a protein in the existing classification of protein structures?

B.2 Datasets

I used SCOP database for this study (please refer to Appendix A for details of SCOP). The *All-α* protein class contains a fold level called *cytochrome c*, which in turn is composed of a single superfamily named *cytochrome c*. This superfamily has four families. The *Di-haem cytochrome c peroxidase* family has only synthetic protein structures and, therefore, only domains from the other families (39 sequences) were used in this analysis.

P-loop domains are found in the class α/β and fold/superfamily *P-loop containing nucleotide triphosphate hydrolases* (this fold has only one superfamily). The superfamily has domains composed of parallel beta sheets of varied sizes connected by helices. For example, the *Nucleoside and nucleotide kinases* family has 5 strands with architecture type 23145 and *Nitrogenase iron-protein like group* family has 7 strands with architecture type 3241567. The superfamily is composed of 14 families. I used all the domains (85 sequences, excluding domains involving multiple chains) from these 14 families for this analysis.

From these datasets, I then found sequence homologs and structure homologs that can be detected by the above described methods.

B.3 Intermediate sequence search

I collected homologs for each of the domains in the two superfamily datasets using FASTA 3.3 (with BLOSUM 62 matrix, ktup = 1) by searching against the pdb90d_1.53 database. The pdb90d_1.53 database is derived from sequences of SCOP domains (version 1.53) sharing 90% or less sequence identity.

Domains (query and target), with scores better than the threshold value 0.01, are referred as 'direct hits'. For domains that cannot be detected directly, I used the ISS procedure described above to link the query and target.

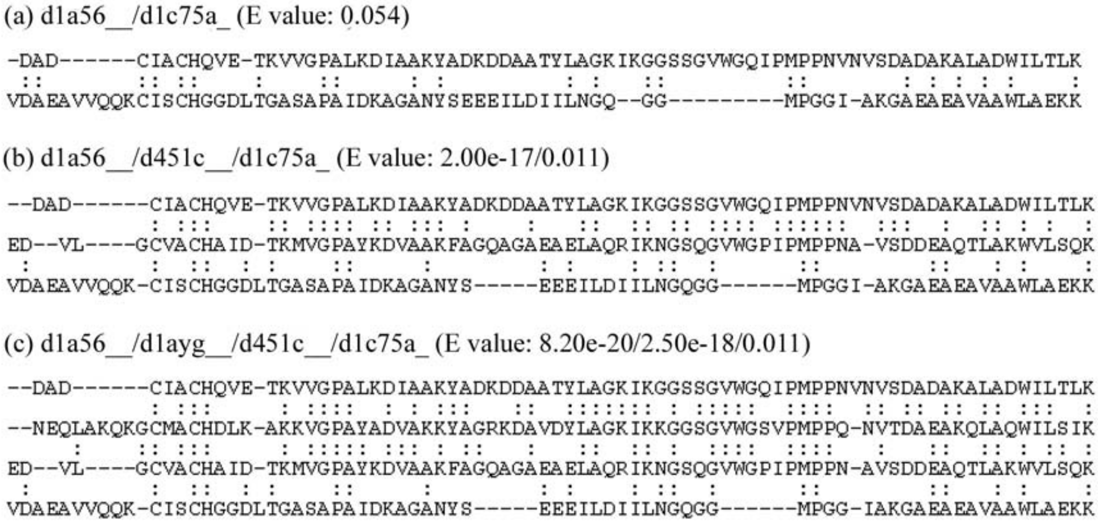
A comparison of ISS hits with psi-BLAST shows that psi-BLAST can detect all the remote homologs identified by ISS in P-loops superfamily and only about half of them in cytochrome c superfamily. The advantage ISS has in some cases might be due to the match score it gains by producing longer alignments around conserved regions of the protein. However, both the methods fail to detect remote homologs from P-loops superfamily than found from cytochrome c superfamily. This might be due to the extensive divergence of sequences in P-loops superfamily (they are quoted to have some converged domains (Bossemeyer, 1994) and differences in sequence length (average length of P-loops is ≈ 230 amino acids, twice the size of cytochrome c).

Intermediate searches based on structural information could find new remote homologs that ISS could not detect. This is expected because it is known that different sequences can have similar folds. Therefore, by comparing structures it is more likely to detect remote homologs. I suggest that by using intermediate structural search, even more distant relationships can be detected.

Then I used the alignments obtained from the query-intermediate and target-intermediate to generate a "progressive alignment" (i.e., a multiple sequence alignment generated by progressively aligning pairwise alignments using *ClustalW* alignments and structure information) of query-intermediate-target or query-intermediate-intermediate-target.

These progressive alignments show that the intermediates can improve the quality of alignments between query and target. An example of this alignment is shown in Figure 72.

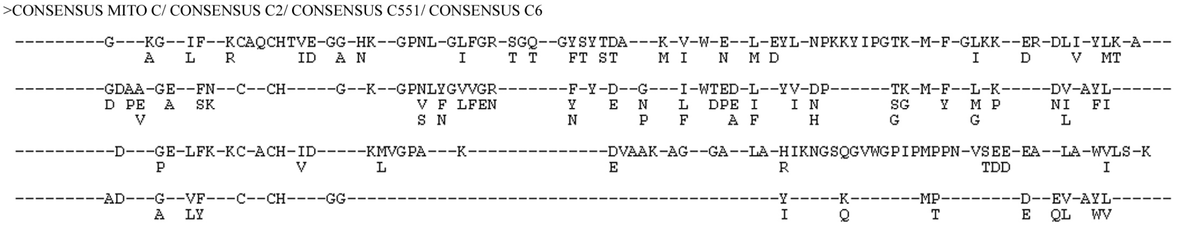
The figure shows the improvement in alignment between query-target (SCOP Ids: *d1a56__ - d1c75a_*) produced by FASTA (Figure 72a) and the progressive alignment generated manually after introducing one (*d451c__*) and two intermediate (*d1ayg__* and *d451c__*) sequences (Figure 72b and Figure 72c). The alignment shows that there are some residues common in all the sequences and some between query-intermediate, target-intermediate and intermediate-intermediate.



```
(a) d1a56__/d1c75a_ (E value: 0.054)

-DAD------CIACHQVE-TKVVGPALKDIAAKYADKDDAATYLAGKIKGGSSGVWGQIPMPPNVNVSDADAKALADWILTLK
 ::       :: ::    :    ::     : :         : :   ::         ::       : : : : :    :
VDAEAVVQQKCISCHGGDLTGASAPAIDKAGANYSEEEILDIILNGQ--GG---------MPGGI-AKGAEAEAVAAWLAEKK

(b) d1a56__/d451c__/d1c75a_ (E value: 2.00e-17/0.011)

--DAD------CIACHQVE-TKVVGPALKDIAAKYADKDDAATYLAGKIKGGSSGVWGQIPMPPNVNVSDADAKALADWILTLK
           : :::    :: :::: :: ::: :    :   ::  :: :: :::: ::::::  :::  :  :: : :  :
ED--VL----GCVACHAID-TKMVGPAYKDVAAKFAGQAGAEAELAQRIKNGSQGVWGPIPMPPNA-VSDDEAQTLAKWVLSQK
 :         :  ::  : :    ::     :        : :    : ::  :      ::        ::   : :    :
VDAEAVVQQK-CISCHGGDLTGASAPAIDKAGANYS-----EEEILDIILNGQGG------MPGGI-AKGAEAEAVAAWLAEKK

(c) d1a56__/d1ayg__/d451c__/d1c75a_ (E value: 8.20e-20/2.50e-18/0.011)

--DAD------CIACHQVE-TKVVGPALKDIAAKYADKDDAATYLAGKIKGGSSGVWGQIPMPPNVNVSDADAKALADWILTLK
           : :::     : ::::  : : :::   ::  ::::::: : :::::  ::::  :: :: :: :: :::  :
--NEQLAKQKGCMACHDLK-AKKVGPAYADVAKKYAGRKDAVDYLAGKIKKGGSGVWGSVPMPPQ-NVTDAEAKQLAQWILSIK
     :    :: :::     : ::::: ::: : ::   :   ::  :: :  ::::  ::::     : ::  :: : :: :
ED--VL----GCVACHAID-TKMVGPAYKDVAAKFAGQAGAEAELAQRIKNGSQGVWGPIPMPPN-AVSDDEAQTLAKWVLSQK
 :         :  ::  : :    ::     :        : :    : ::  :      ::        ::   : :    :
VDAEAVVQQK-CISCHGGDLTGASAPAIDKAGANYS-----EEEILDIILNGQGG------MPGG-IAKGAEAEAVAAWLAEKK
```

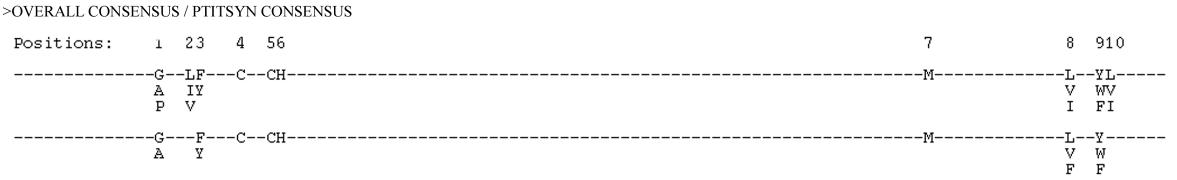
Figure 72. Comparison of alignments of two distant proteins with and without intermediates. (a) Alignment of the two domain produced by FASTA 3.3. (b) The progressive alignment generated by including one intermediate. (c) The progressive alignment generated by including two intermediates.

Likewise, I selected closely clustered domains from each of the four SCOP protein groups (*mitochondrial cytochrome*, *cytochrome c_2*, *cytochrome c_{551}* and *cytochrome c_6*) to make a progressive alignment. These groups were used due to the fact that they represent most of the members of the superfamily. From the progressive alignment made for each of the protein groups, I derived a consensus (Figure 73). This consensus was then used to derive an overall consensus shown in Figure 74. The figure shows that there are 10 invariable residues in the consensus and it agrees with the consensus derived by Ptitsyn by aligning 164 sequences from the cytochrome c superfamily (Ptitsyn, 1998). His alignments were generated using the PileUP program and manually edited taking functional residues into consideration.



```
>CONSENSUS MITO C/ CONSENSUS C2/ CONSENSUS C551/ CONSENSUS C6

---------G---KG--IF--KCAQCHTVE-GG-HK--GPNL-GLFGR-SGQ--GYSYTDA---K-V-W-E--L-EYL-NPKKYIPGTK-M-F-GLKK--ER-DLI-YLK-A---
             A   L   R      ID  A N         I    T T   FT ST    M I   N  M D                   I    D    V  MT

---------GDAA-GE--FN--C--CH-----G--K--GPNLYGVVGR-------F-Y-D--G---I-WTED-L--YV-DP------TK-M-F--L-K-----DV-AYL------
         D PE  A  SK                    V F LFEN       Y   E  N   L  DPE I   I N       SG   Y  M P     NI FI
            V                           S N            N      P   F    A F     H       G       G        L

----------D---GE-LFK-KC-ACH-ID-----KMVGPA---K--------------DVAAK-AG--GA--LA-HIKNGSQGVWGPIPMPPN-VSEE-EA--LA-WVLS-K
              P             V       L                      E                R                   TDD         I

---------AD---G--VF---C--CH----GG-------------------------------------------Y-----K-------MP--------D--EV-AYL------
              A  LY                                                         I     Q        T        E  QL WV
```

Figure 73. Consensus sequences derived for the four SCOP protein group in monodomain cytochrome c family



```
>OVERALL CONSENSUS / PTITSYN CONSENSUS

Positions:   1  23   4  56                                                            7            8  910

-------------G--LF---C--CH------------------------------------------------------------M------------L--YL-----
             A  IY                                                                                 V  WV
             P  V                                                                                  I  FI

-------------G---F---C--CH------------------------------------------------------------M------------L--Y------
             A   Y                                                                                 V  W
                                                                                                   F  F
```

Figure 74. Consensus of consensus for sequences in monodomain cytochrome c family

The conserved residues were involved in heme binding and needed for functional role of the protein. The other conserved residues do not have any functional role and are found to be key residues needed to maintain structural fold of cytochromes. The key residues reported here agree well with the results found in the literature (Ptitsyn, 1998). Figure 74 shows the key residues identified by Ptitsyn. The differences include two additional residues conserved at position 3 (aliphatic residue) and position 10 (aliphatic residue), the presence of a proline at position 1 and a phenyalanine instead of an isoleucine at position 8. These discrepancies might be due to number of sequences compared and the kind of alignment generated. Ptitsyn used 164 sequences whereas here only 19 sequences were used. Although comparatively very few sequences were used, the result seems to be almost the same. This is a promising result opening opportunities in extending the procedure to other superfamilies. However, an attempt on P-loops failed primarily due to the fact that the superfamily is much more diverged and only very few sequences form distinct clusters.

B.4 Structural homologs

I did an all-against-all structural comparison of the domains using PrISM. Then I used the alignment from PrISM as input to another program called MSARMS (Hubbard, 1994) that measures the distance in Angstrom between the matched residues in the superposition. These RMSD values from PrISM and MSARMS programs were used for this study.

B.5 Clustering

With these homologs and their relationship (given as *E-value* for sequences and *RMSD* for structures), I represented proteins as clusters in two-dimensional space. This was done using the procedure given in Figure 75 using sequence/structure distance matrices (or similarity matrices).

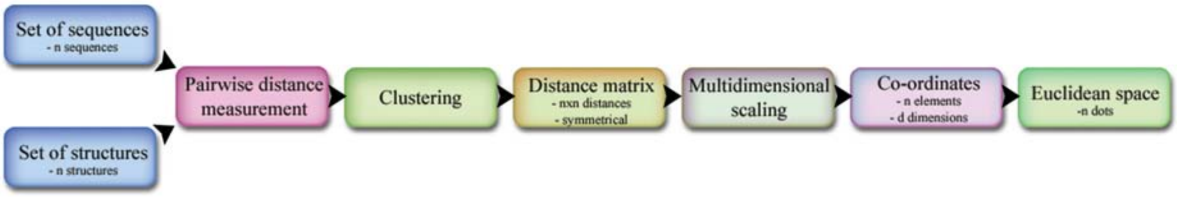


Figure 75. Flow chart describing steps used in clustering and visualisation of data.

I did initial clustering based on the sequence based distance matrix using single and complete linkage methods with a threshold E-value of 0.001 and 0.05 respectively. Then I merged the resulting sets of clusters based on the RMSD values using the Unweighted Pair Group Method using Arithmetic average approach. A threshold value of 4.00Å was used for the P-loops superfamily and a threshold of 2.00Å was used for the cytochrome c superfamily. I also applied the complete linkage approach to merge the initial set of clusters using a threshold value of 6.00Å for both superfamilies.

To find co-ordinates of the data set in 2D space, I used Principal Co-ordinate Analysis (PCoA). For a problem of *N* objects, there could be *N*(N-1)* distances and displayed in *(N-1)* dimensional space. This *(N-1)* dimensional space was reduced to 2D/3D space and plotted.

A manual plotting of the data gave a cluster map for both cytochrome c (Figure 76) and P-loops superfamilies (Figure 77). Figure 78 shows the demarcation of clusters into family and protein levels based on the SCOP classification for cytochrome c. Similarly, Figure 79 shows the demarcation of family levels in P-loops. The protein levels were not marked in P-loops to avoid the complexity in the figure.

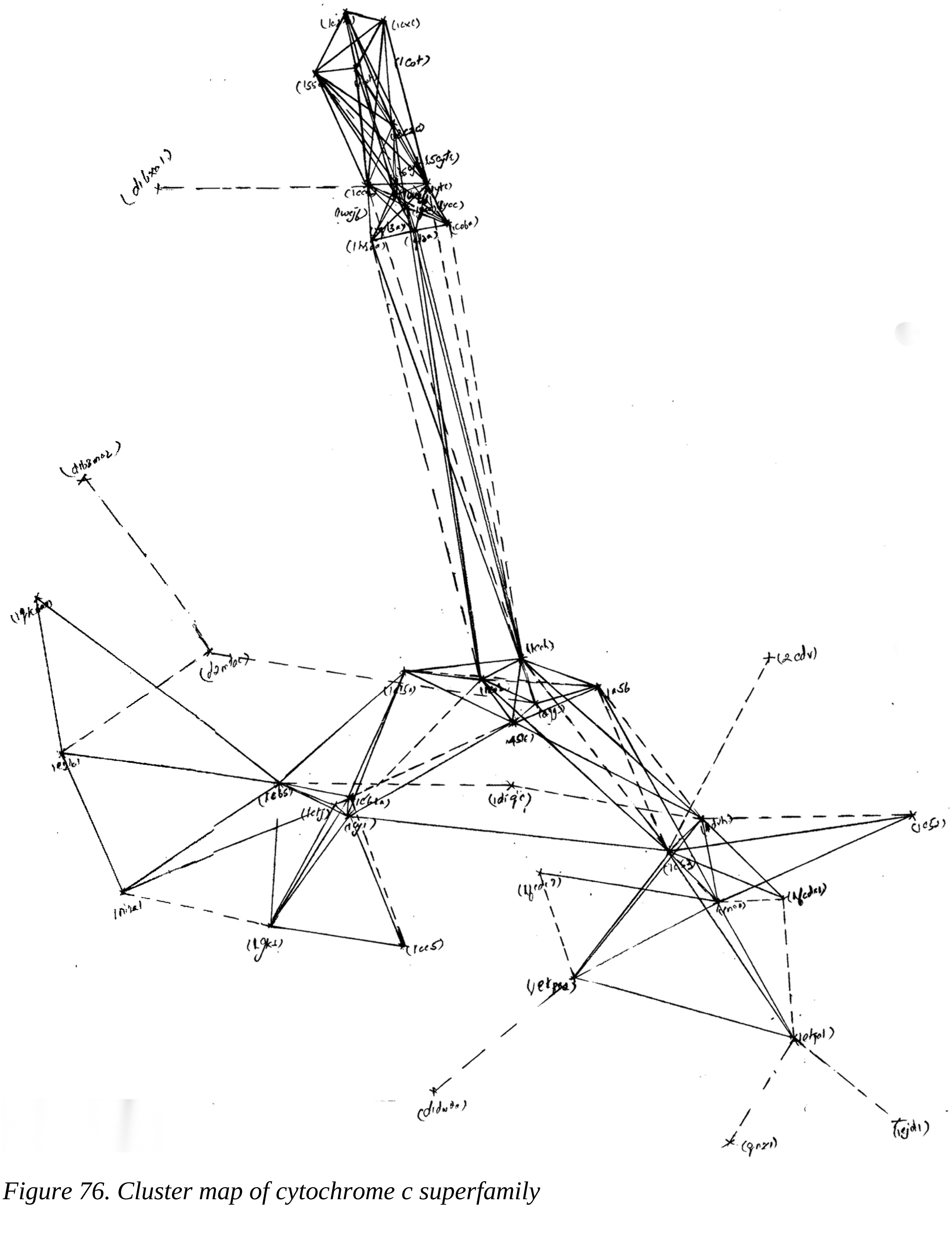


Figure 76. Cluster map of cytochrome c superfamily

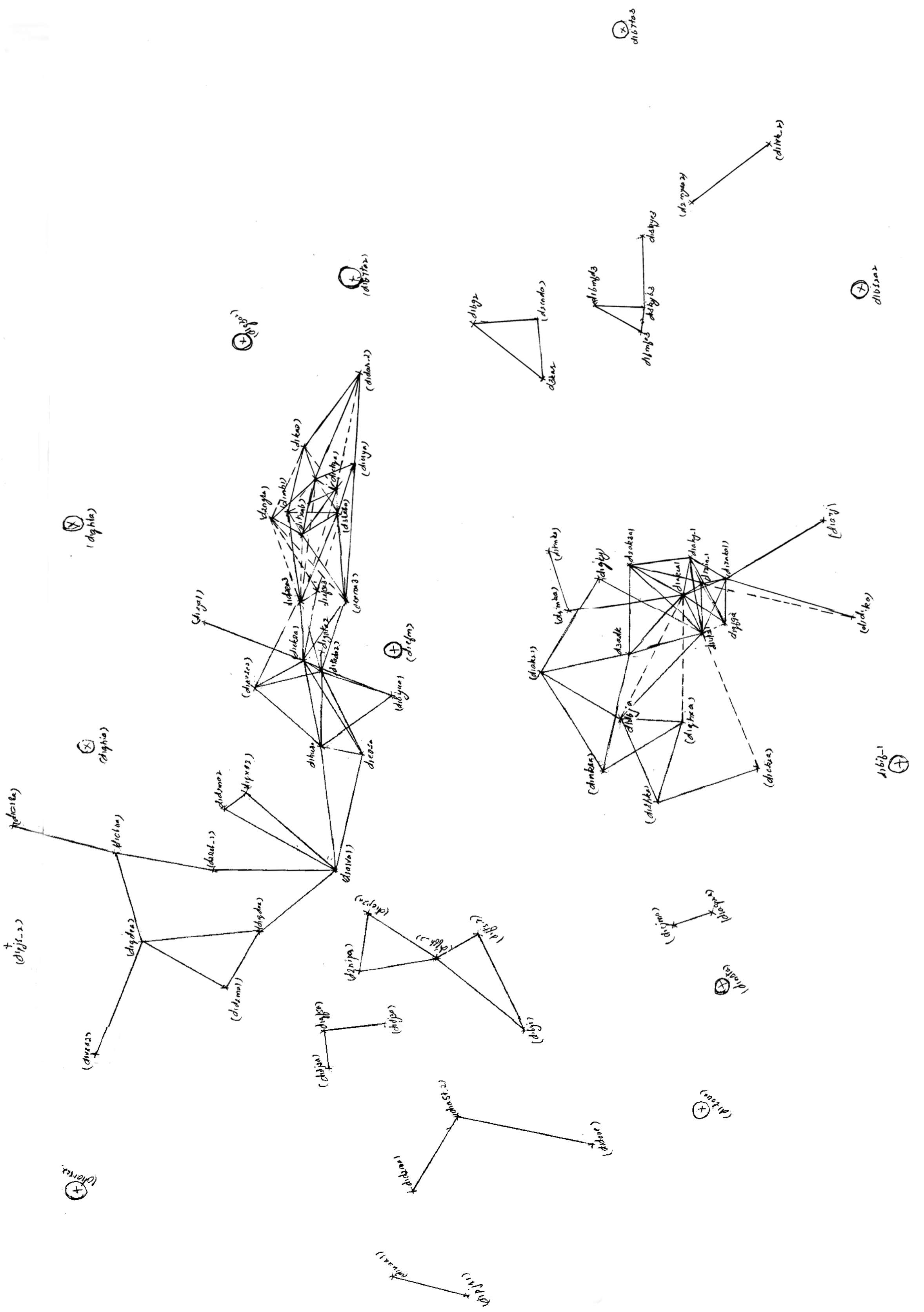


Figure 77. Cluster map of P-loops superfamily

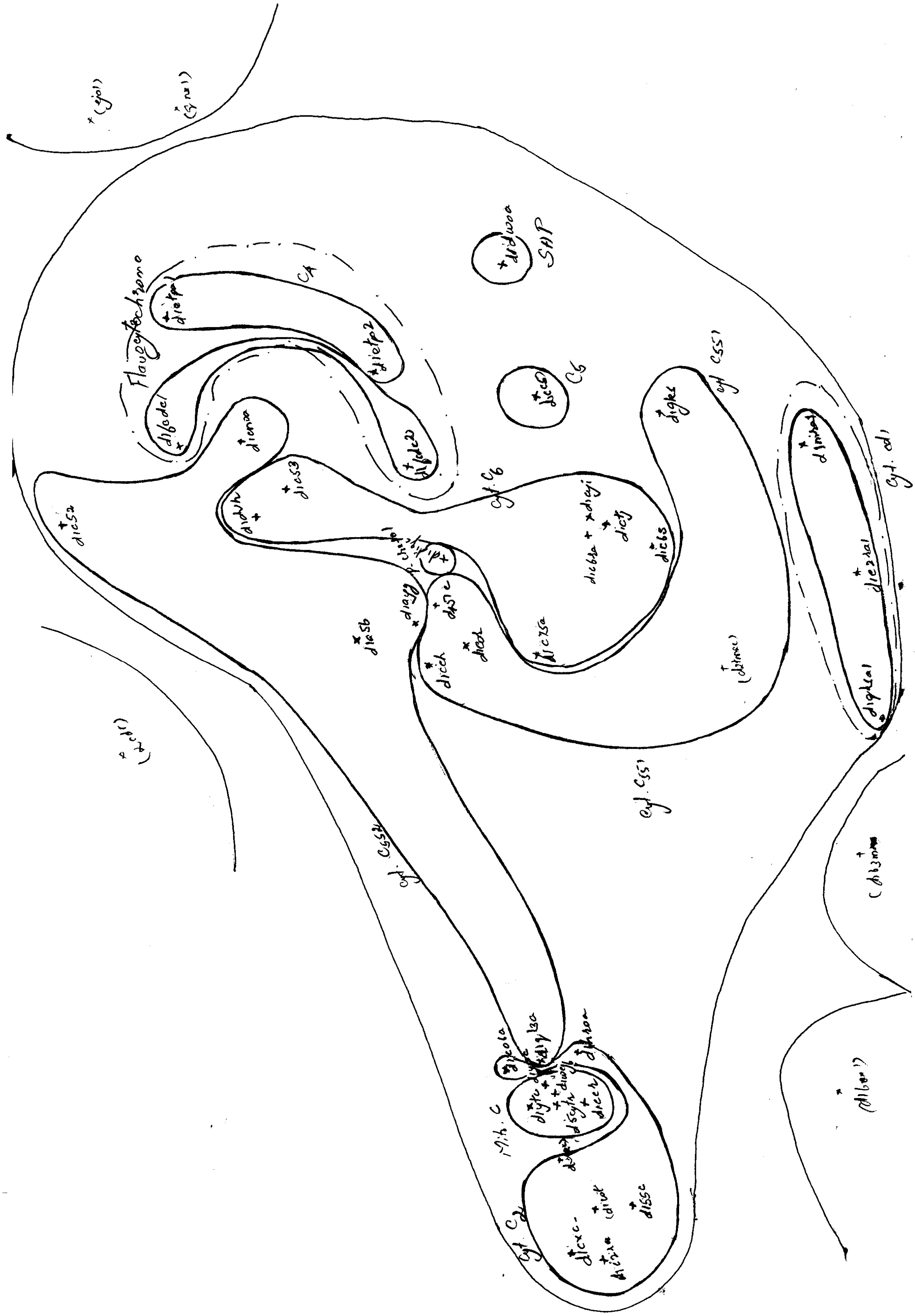


Figure 78. Cluster map of cytochrome c superfamily with demarcation of SCOP superfamily, family and protein levels

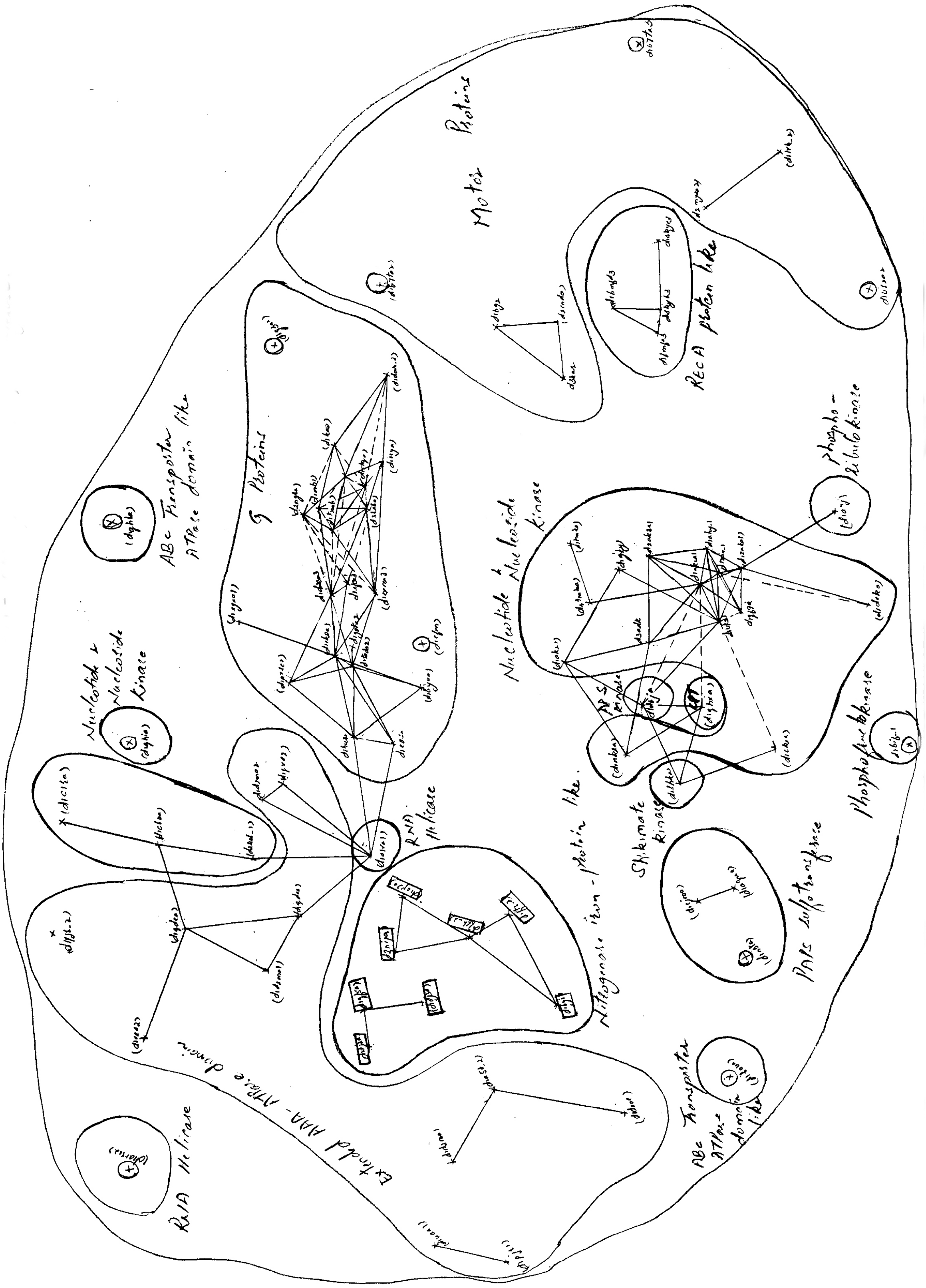

Motor Proteins
RecA protein like
G Proteins
ABC Transporter like ATPase domain
Nucleotide & Nucleoside kinase
Nucleotide & Nucleoside Kinase
RNA helicase
Nitrogenase iron-protein like
APS kinase
Shikimate kinase
Phospho-ribulokinase
Phosphofructokinase
PAPS sulfotransferase
ABC Transporter ATPase domain like
Extended AAA-ATPase domain
RNA Helicase

Figure 79. Cluster map of P-loops superfamily with demarcation of SCOP superfamily, family levels

The maps (Figure 76 and Figure 77) show domain relationships either by solid lines or dashed lines. The solid lines indicate domains having strong relationship between them (E-value < 0.4 and RMSD < 4 Å). Also, the length of the solid line represents real Euclidean distance in the cluster map. The dashed lines show there is a relationship between the connected domains. However, the position of domains in the map is not true. This is due to the non-availability of a relationship between the connected domains and its neighbors. Also, the length of the broken line does not represent real Euclidean space in the map.

The cytochrome maps (Figure 76 and Figure 78) show that two SCOP protein groups, *mitochondrial cytochrome c* and *cytochrome* c_2, were well separated from other protein groups. The domains forming the *cytochrome* c_{552} cluster show that they have diverged more than any other SCOP protein group. Also, it can be seen that most of the domains from the *cytochrome* c_6 and *cytochrome* c_{551} SCOP protein groups form closer clusters while some of them get away from this cluster and act as outliers.

P-loops cluster maps (Figure 77 and Figure 79) show that the domains have diverged more when compared to the cytochrome c domains. The maps show a number of domains represented as singletons or as small groups not connected to each other. As stated earlier, absence of a line between domains means no relationship can be identified among them (with score below the threshold limit), although some of the singletons belong to SCOP family. Only members of two families (*Nucleoside and nucleotide kinase* and *G-proteins*) were found to be grouped together on the map. This may be due to more environmental constraints and less active site requirements on P-loop superfamily or may be due to a convergence phenomena as seen in phosphate binding proteins (Bossemeyer, 1994).

These cluster maps are a useful tool to aid in understanding of the relationship between protein members of a family:

(1) It gives an overall picture of the divergence of a protein superfamily.
(2) It shows the relationships between SCOP families.
(3) The method could be used as an initial automated classification procedure of protein structures. A new protein structure can be used as a query to find its sequence or structure homologs. Then based on the sequence and structural relationship (E-value and

RMSD), the protein can be added in the cluster map. Such a map will give a good idea to which of the superfamily or family the new protein belongs. Then with detailed knowledge, the protein can be allocated in a specific family (manual curation). The clustering approach can be exploited to assign function to an unknown protein (Sternberg, 2001), but it cannot be trusted fully as a similar structure does not always represent the same function.

(4) It gives a clear picture about any particular SCOP family and allows the identification of any outliers in it. In the P-loops cluster map (Figure 79), there are two clusters one with domains *d1d2ja__, d1qf5a__ and d1dj3a_* and another with *d2nipa_, d1cp2a_, d1ffh__, d1byi__ and d1fts__* (boxed). But all of these domains are placed in the same family in SCOP. On discussion with Alexey Murzin (the primary curator of SCOP database), he recalled he considered that it might be better to keep these two clusters in two separate groups, say as, two different sub-families/families. He only kept them together due to limitations in the current SCOP classification system.

Likewise the domain *d1qhia_*, classified in the *Nucleotide and nucleoside kinase* family in SCOP, are positioned separately from the main cluster. The outlier was later cross-checked with structural analysis (Morea, 2001). The analysis also agreed that the domain is distinct from its family members. The probable reason for the isolated cluster of *d1qhia__* is that it is a chimeric protein and does not exist naturally i.e. it does not have sequence or structure homology with other *Nucleotide and nucleoside kinase* proteins even though it retains the same function. It was for this reason and since the domain satisfied minimal the P-loop topology, that Alexey Murzin classified the domain under the same family.

Thus, cluster maps might help us to be aware of outliers in a particular superfamily/family classification before starting any kind of detailed analysis on it.

Because of these advantages of the cluster maps, I automated the clustering process to extend the study later for other families. A comparison between manual and automated clustering procedures shows that the automated method performed equally well with the manual method (Figure 80 and Figure 81). Also, the automated methods provide similar results with another automated clustering procedure based on the MCL algorithm (Enright *et al.*, 2002).

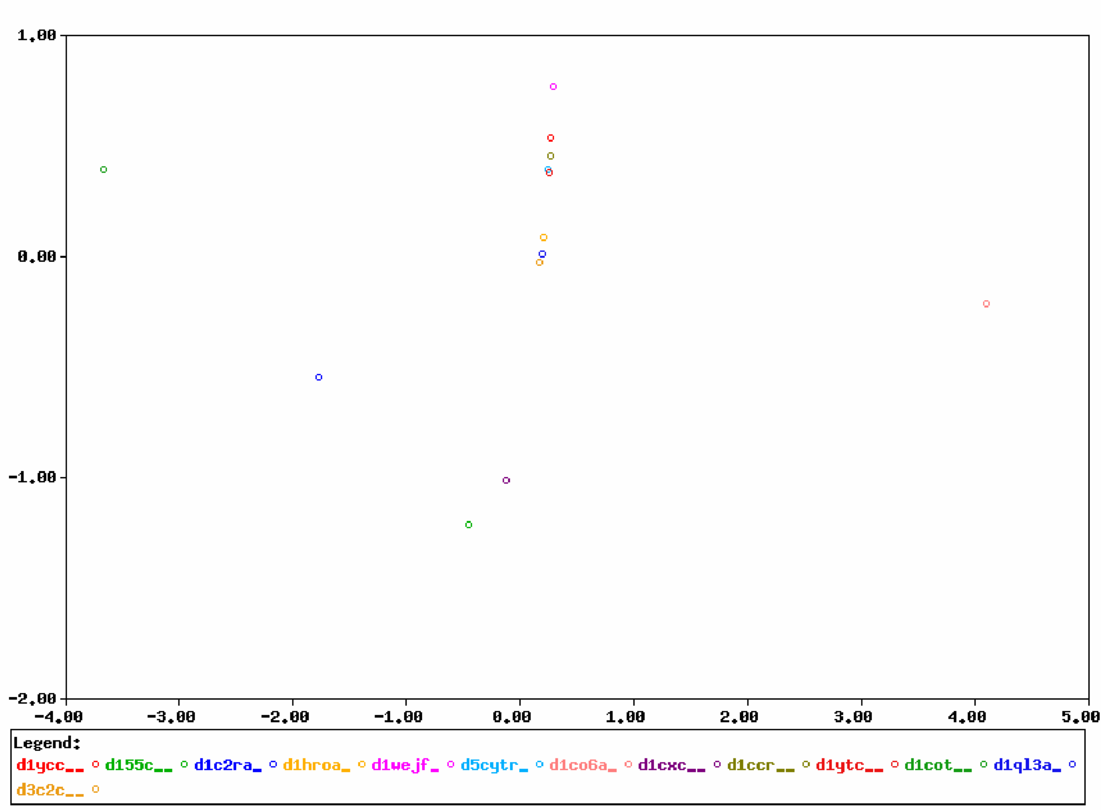


Figure 80. A cluster produced by the automated method for cytochrome c superfamily

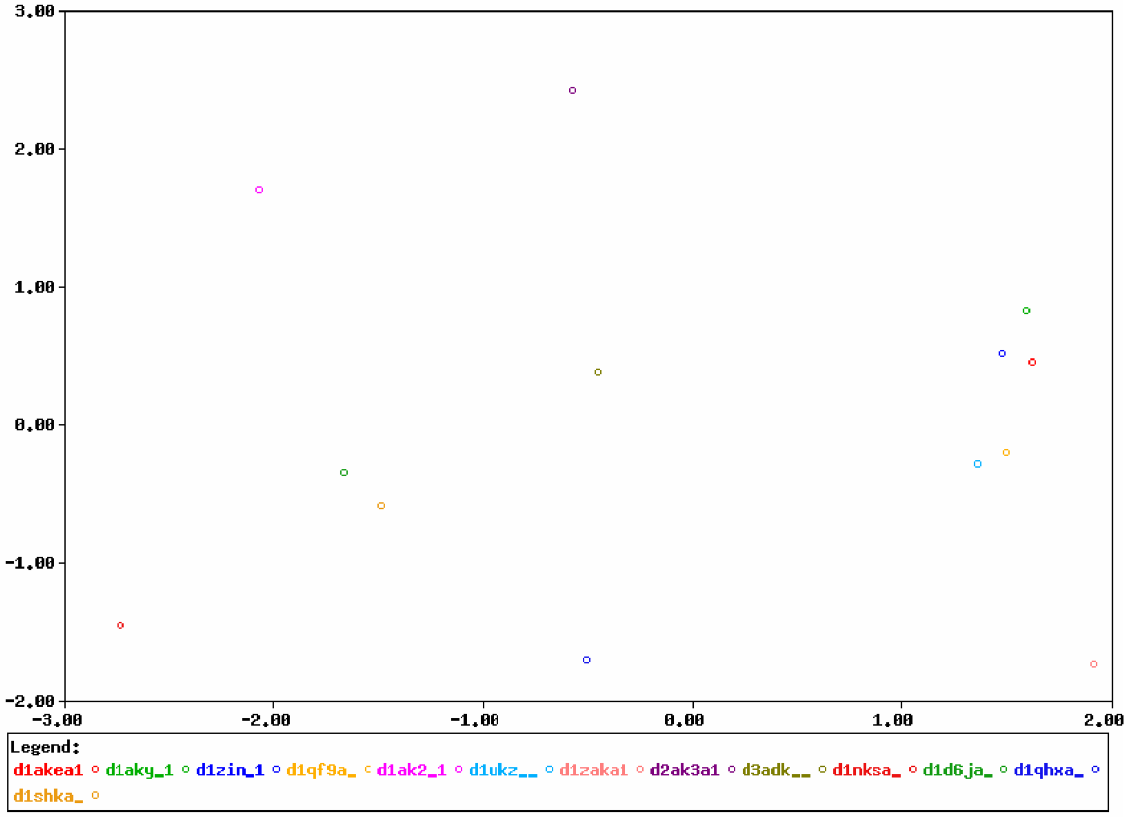


Figure 81. A cluster produced by automated method for P-loops superfamily

In both manual and automated processes, clustering was done using sequence and structural relationships, but it is possible to be done with sequence information alone. However, this will give only the number of clusters that can be formed from the superfamily and members in each cluster. A two dimensional representation of data is difficult with sequence information alone due to the fact that the data needs to undergo significant normalization procedures before it can be used to find co-ordinates.

B.6 Orthology and paralogy

The sequence and structural information, used above to generate cluster maps, can also form the basis for detecting orthologous relationships within protein families in the study of protein evolution. Such a group of ortholog domains was found in P-loops superfamily. The group comprises adenylate kinases from *Escherichia coli, Bacillus stermathermophilus* and *Saccharomyces cerevisiae*. Using species as a time scale, it can be said that adenylate kinase of *Escherichia coli* and *Bacillus stermathermophilus* appeared earlier than yeast protein. However, it does not mean that yeast protein evolved from *Escherichia coli* or *Bacillus* and it would be extremely difficult in assessing the proper time scale for these proteins based on sequence and structure information alone.

All the three adenylate kinases clustered close to each other on the map. So, from tightly clustering domains, it can be presumed that they are possibly to be orthologous to each other.

The TOPS (Westhead *et al.*, 1999) diagrams of these three proteins (Figure 82) shows that *Escherichia coli* and yeast adenylate kinases are identical whereas in *Bacillus*, there is an extra β strand and its orientation is reversed. Interestingly, this part of the protein is not under SCOP domain definition, which means that there is no functional or structural role for this part of the protein. Since this part does not have structural or functional constraints, it is more likely to be subject to mutations and may be influenced by environmental factors of *Bacillus* compared with yeast or *Escherichia coli*. From this, I conclude that the evolution of adenylate kinase would have more likely started from a common ancestor and given rise to *Escherichia coli* and or *Bacillus* and later to yeast protein. Later, *Bacillus* adenylate kinase would have acquired some changes in its protein.

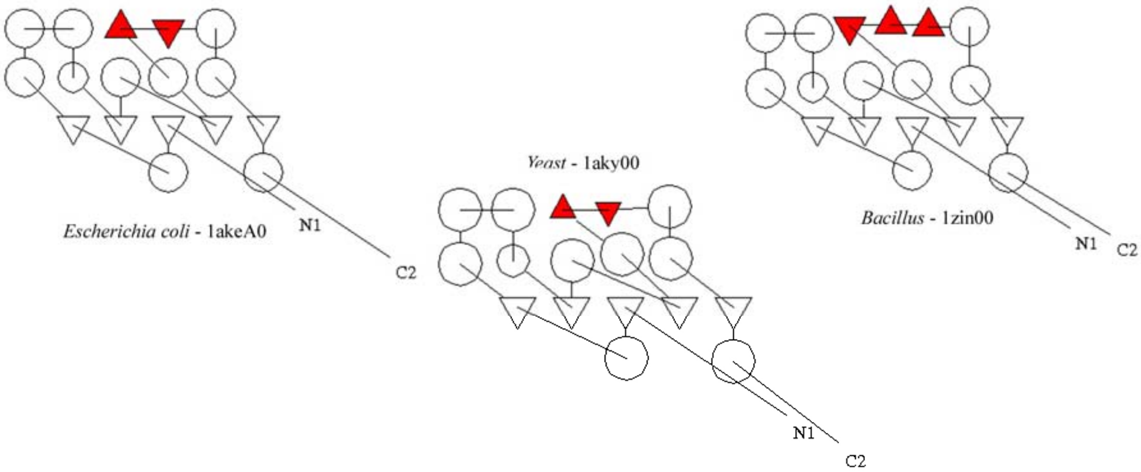


Figure 82. Topology diagram for adenylate kinase

Likewise, from the cytochrome map, two SCOP protein groups form distinct clusters from the rest of the cytochrome members. The overall topology of the cytochrome superfamily members were analyzed using TOPS (Figure 83). Generally, cytochrome c fold has 5 helices. However, some members of *cytochrome* c_{551} group have 6 helices and *cytochrome* c_2 group has 5 helices and 2 β strands except *d3c2c__*, which has only 5 helices. The topology of *cytochrome* c_{552} group (5 helices) remains the same, although its sequence has diverged greatly. However, the domains of this group (*cytochrome* c_{552}) forms close cluster with domains of different cytochrome c protein groups than among itself. It might be one of the typical cases, where orthology/homology cannot be resolved based on sequence identity because an extensive sequence divergence has occurred. However, it can also be argued that *cytochrome* c_{552} proteins were actually formed from convergence of different cytochrome c proteins. But this is highly unlikely to occur given the clear picture of overall divergence of cytochrome proteins and absence of any convergence reports in the cytochrome c fold.

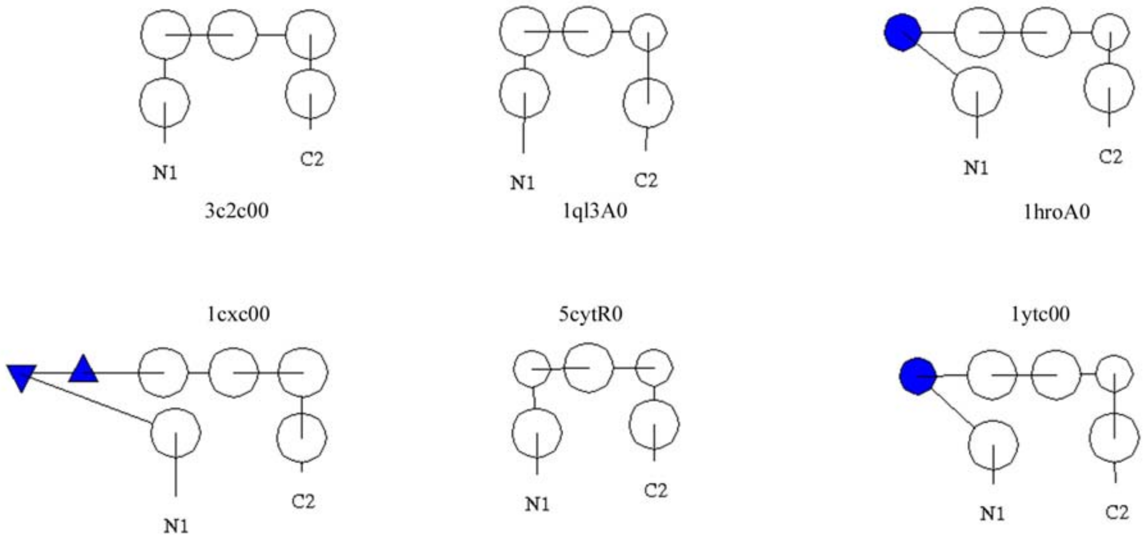


Figure 83. Topology diagram for cytochrome c proteins

Mitochondrial cytochrome c was seen later in the time-scale when compared to bacterial cytochrome c. Given the endosymbiotic hypothesis, it is likely that any bacterial cytochrome c would have given rise to *mitochondrial cytochrome*. Here, it can be seen that *cytochrome* c_2 clustered closely with *mitochondrial cytochrome* (Figure 78). So it is likely that *cytochrome* c_2 would have been the ancestral protein for *mitochondrial cytochrome*. This was confirmed with expertise knowledge of Alexey Murzin. The topology study of these two SCOP protein groups also confirmed this. The general topology of *cytochrome* c_2 and *mitochondrial cytochrome* are 5 helices + 2 β strands and 5 or 6 helices respectively. However, some of the domains of *cytochrome* c_2 (e.g., *d3c2c__*), clustering near to *mitochondrial cytochrome* lack the two β strands, confirming that the earlier forms of *cytochrome* c_2 with β strands, later lost the β strands and have given rise to *mitochondrial cytochrome*.

Thus, cluster maps made with sequence and structural homology is useful in understanding the ancestry of proteins.

B.7 Conclusions

Protein evolution, driven by structural and functional constraints, may leave a trail of homologs. Homologs are identified using sequence comparison methods like BLAST, FASTA, psi-BLAST and ISS. A comparison of ISS with psi-BLAST was made in two protein superfamilies: cytochrome c and P-loops. The result showed that psi-BLAST detected all the remote homologs identified by ISS in P-loops and only half in cytochrome c superfamily. Although, I cannot generalize using these limited results, it can be said that ISS performs better in some cases than psi-BLAST. The advantage ISS has in some cases might be due to the match score it gains by producing longer alignments around conserved regions of the protein. Intermediate search conducted using structural information revealed that more remote homologs that could not be identified with sequence information alone. So structures might be useful in intermediate search when sequence information is inadequate in detection. From the progressive alignments generated using most of the domains in four SCOP protein groups (*mitochondrial cytochrome, cytochrome* c_2, *cytochrome* c_{551} and *cytochrome* c_6), an overall consensus was generated. The highly conserved residues found in the overall consensus are in tandem with the key structural and functional residues needed for the cytochrome c fold (Ptitsyn, 1998). Thus ISS alignments might be useful in understanding highly conserved residues in a protein fold.

Along with sequence information, I used structural comparisons by PrISM to produce a manual cluster map. The cluster map showed a useful representation of the general evolutionary relationships within P-loops and cytochromes. These might be helpful in depicting the relationship between SCOP families, assigning hierarchies to a new protein structure in the existing structural classification and understanding the likely ancestor of a protein. For example, in cytochrome c superfamily, it was shown that the *cytochrome* c_2 protein is likely to be an ancestor for *mitochondrial cytochrome*. The manual process has been automated and can now be used as a tool in exploring evolutionary relationships of any protein family.

APPENDIX C

C.1 Eponine transcription termination parameters

The parameters used to create Eponine transcription termination model –

```
<? xml version="1.0" ?>

<app xmlns=http://www.sanger.ac.uk/Users/td2/specs/epoapps/0/2
jclass="eponine.TrainingCore">

        <bean name="dataSource" jclass="eponine.datasource.XMLDataSource">
                <string name="fileName" value="Datasets/trainingdata.xml " />
        </bean>

        <bean name="basisSource" jclass="eponine.model.MultiplexedBasisSource">
                <int name="reweightFrequency" value="15" />
                <double name="reweightPseudocounts" value="10.0" />

                <child jclass="eponine.model.NewBasisSource">
                     <boolean name="maximize" value="false" />
                     <double name="stringency" value="0.55" />
                     <double name="stringencyVariance" value="0.03" />
                     <int name="minLength" value="4" />
                     <int name="maxLength" value="8" />
                     <double name="minDistWidth" value="2.5" />
                     <double name="maxDistWidth" value="200.0" />
                     <boolean name="reversible" value="false" />
                     <string name="name" value="nbs1_narrow" />
                     <int name="minPos" value="-190" />
                     <int name="maxPos" value="1990" />
                </child>

                <child jclass="eponine.model.SampleWMBasisSource">
                     <double name="nullModelWeighting" value="7.0" />
                     <double name="nullModelPerMarginalColumn" value="1.0" />
                     <int name="sampleCounts" value="203" />
                     <double name="nullModelWeightingN" value="9.0" />
                     <int name="sampleCountsN" value="120" />
                     <string name="name" value="samplewm2" />
                </child>

                <child jclass="eponine.model.DropColumnBasisSource" />

                <child jclass="eponine.model.DistributionBasisSource">
                     <double name="distChangeWidth" value="3.0" />
                     <double name="distChangeGamma" value="3.0" />
                     <double name="distChangeScale" value="25.0" />
                     <double name="distChangeBias" value="0.06" />
                     <!-- double name="shapeChangeProbability" value="0.05" / -->
                     <double name="flipEnvelopeProbability" value="0.00" />
                     <string name="name" value="distwidth" />
                </child>

                <child jclass="eponine.model.PositionBasisSource">
                     <double name="shiftWidth" value="4" />
                </child>

                <child jclass="eponine.model.CrossWMBasisSource" />

                <child jclass="eponine.model.AppendColumnBasisSource" />

                <child jclass="eponine.model.FlipMaxBasisSource" />
        </bean>

        <bean name="trainer" jclass="stats.glm.VRVMTrainer">
                <int name="numThreads" value="4" />
                <int name="maxCycles" value="11000" />
                <!--int name="cleaningCycles" value="0" /-->
                <int name="maxWorkingSet" value="28" />
                <int name="minWorkingSet" value="25" />
                <int name="initialWorkingSet" value="50" />
                <double name="initialAlpha" value="1.0" />
                <boolean name="unityHack" value="true" />
                <double name="unityHackThreshold" value="1.0" />
                <boolean name="resetAlphaHack" value="true" />
                <boolean name="insertUnity" value="true" />
        </bean>

        <bean name="retrainer" jclass="stats.glm.VRVMTrainer">
                <int name="maxCycles" value="100" />
                <!--int name="cleaningCycles" value="0" /-->
                <double name="initialAlpha" value="1.0" />
                <boolean name="unityHack" value="true" />
                <double name="unityHackThreshold" value="1.0" />
        </bean>

        <string name="fileName" value="Models/terminationmodel.xml" />
        <int name="checkpointFrequency" value="500" />
</app>
```

C.2 GAZE gene structure models

The configuration file explaining the gene model with translation features for predicting genes using GenePred –

```
<? xml version="1.0" encoding="US-ASCII" ?>

    <gaze>
        <declarations>
            <feature id="tss" st_off="0" en_off="1" />
            <feature id="tis" st_off="0" en_off="3"/>
            <feature id="5ss" st_off="1" en_off="1" />
            <feature id="3ss" st_off="1" en_off="1" />
            <feature id="tts" st_off="3" en_off="0"/>
            <feature id="polyA" st_off="1" en_off="1"/>

            <feature id="tss_rev" st_off="1" en_off="0" />
            <feature id="tis_rev" st_off="3" en_off="0" />
            <feature id="5ss_rev" st_off="1" en_off="1" />
            <feature id="3ss_rev" st_off="1" en_off="1" />
            <feature id="tts_rev" st_off="0" en_off="3" />
            <feature id="polyA_rev" st_off="1" en_off="1"/>

            <!--lengthfunction id="intron_pen" />
            <lengthfunction id="intergene_pen" />
            <lengthfunction id="inital_exon_pen" />
            <lengthfunction id="internal_exon_pen" />
            <lengthfunction id="terminal_exon_pen" />
            <lengthfunction id="single_exon_gene_pen" /-->
        </declarations>

        <gff2gaze>
            <!-- Features -->
            <gfffeat feature="TSS" strand="+" source="Eponine">
                <feat id="tss"/>
            </gfffeat>

            <gfffeat feature="TSS" strand="-" source="Eponine">
                <feat id="tss_rev"/>
            </gfffeat>

            <gfffeat feature="TIS" strand="+" source="Eponine">
                <feat id="tis"/>
            </gfffeat>

            <gfffeat feature="TIS" strand="-" source="Eponine">
                <feat id="tis_rev"/>
            </gfffeat>

            <gfffeat feature="5SS" strand="+" source="Eponine">
                <feat id="5ss"/>
            </gfffeat>

            <gfffeat feature="5SS" strand="-" source="Eponine">
                <feat id="5ss_rev"/>
            </gfffeat>

            <gfffeat feature="3SS" strand="+" source="Eponine">
                <feat id="3ss"/>
            </gfffeat>

            <gfffeat feature="3SS" strand="-" source="Eponine">
                <feat id="3ss_rev"/>
            </gfffeat>

            <gfffeat feature="TTS" strand="+" source="Eponine">
                <feat id="tts"/>
            </gfffeat>
            <gfffeat feature="TTS" strand="-" source="Eponine">
                <feat id="tts_rev"/>
            </gfffeat>

            <gfffeat feature="POLYA" strand="+" source="Eponine">
                <feat id="polyA"/>
            </gfffeat>

            <gfffeat feature="POLYA" strand="-" source="Eponine">
                <feat id="polyA_rev"/>
            </gfffeat>
        </gff2gaze>

        <dna2gaze>
            <!--dnafeat pattern="tataaa">
                <feat id="tss" />
            </dnafeat>

            <dnafeat pattern="atg" score="0.001">
                <feat id="tis" />
            </dnafeat>

            <dnafeat pattern="taa" score="0.001">
                <feat id="tts" />
            </dnafeat>

            <dnafeat pattern="tag" score="0.001">
                <feat id="tts" />
            </dnafeat>

            <dnafeat pattern="tga" score="0.001">
                <feat id="tts" />
            </dnafeat>

            <dnafeat pattern="aataaa" score="0.001">
                <feat id="polyA" />
            </dnafeat>

            <dnafeat pattern="tttata">
                <feat id="tss_rev" />
            </dnafeat>

            <dnafeat pattern="cat" score="0.001">
                <feat id="tis_rev" />
            </dnafeat>

            <dnafeat pattern="tta" score="0.001">
                <feat id="tts_rev" />
            </dnafeat>

            <dnafeat pattern="cta" score="0.001">
                <feat id="tts_rev" />
            </dnafeat>
            <dnafeat pattern="tca" score="0.001">
                <feat id="tts_rev" />
            </dnafeat>

            <dnafeat pattern="tttatt" score="0.001">
                <feat id="polyA_rev" />
            </dnafeat-->

            <!--takedna id="5ss_1" st_off="0" en_off="1"/>
            <takedna id="3ss_1" st_off="1" en_off="-1"/>
            <takedna id="5ss_2" st_off="-1" en_off="1"/>
            <takedna id="3ss_2" st_off="1" en_off="0"/>
            <takedna id="5ss_1_rev" st_off="1" en_off="0"/>
            <takedna id="3ss_1_rev" st_off="-1" en_off="1"/>
            <takedna id="5ss_2_rev" st_off="1" en_off="-1"/>
            <takedna id="3ss_2_rev" st_off="0" en_off="1"/-->
        </dna2gaze>

        <model>
            <target id="END">
                <source id="BEGIN" out_feat="No_genes"/>
                <source id="polyA" out_feat="GEN_DNA" />
                <source id="tss_rev" out_feat="GEN_DNA"/>
            </target>

            <!--Forward strand gene-->

            <target id="tss">
                <source id="BEGIN" out_feat="GEN_DNA"/>
                <source id="polyA" mindis="1" out_feat="intergenic"/>
                <source id="tss_rev" mindis="1" out_feat="intergenic"/>
            </target>

            <target id="tis">
                <source id="tss" mindis="1" out_feat="5UTR" out_str="+"/>
            </target>

            <target id="5ss">
                <!--killfeat id="tts"/-->
                <source id="tis" out_feat="inital_exon" mindis="3" maxdis= "10000" out_str="+" />
                <source id="3ss" out_feat="internal_exon"  mindis="6" maxdis= "10000" out_str="+"
            />
            </target>

            <target id="3ss">
                <source id="5ss" out_feat="intron"  mindis="6" out_str="+"/>
            </target>

            <target id="tts">
                <!--killfeat id="tts" /-->
                <!--source id="tis" out_feat="single_exon_gene" mindis="60"out_str="+"/-->
                <source id="3ss" out_feat="terminal_exon"  mindis="3" out_str="+"/>
            </target>

            <target id="polyA">
```

```
            <source id="tts" out_feat="3UTR" mindis="1" out_str="+"/>
        </target>

        <!--Reverse strand gene-->

        <target id="polyA_rev">
            <source id="BEGIN" out_feat="GEN_DNA"/>
            <source id="polyA" out_feat="intergenic" mindis="1"/>
            <source id="tss_rev" out_feat="intergenic" mindis="1"/>
        </target>

        <target id="tts_rev">
            <source id="polyA_rev" out_feat="3UTR" mindis="1" out_str="-"/>
        </target>

        <target id="3ss_rev">
            <!--killfeat id="tts_rev"/-->
            <source id="tts_rev" out_feat="terminal_exon" mindis="3" maxdis= "10000"
        out_str="-"/>
            <source id="5ss_rev" out_feat="internal_exon" mindis="6" maxdis= "10000"
        out_str="-"/>
        </target>

        <target id="5ss_rev">
            <source id="3ss_rev" out_feat="intron" mindis="6" out_str="-"/>
        </target>

        <target id="tis_rev">
            <!--killfeat id="tts_rev" phase="0"/-->
            <!--source id="tts_rev" out_feat="single_exon_gene" mindis="60" out_str="-"/-->
            <source id="5ss_rev" out_feat="initial_exon" mindis="3" out_str="-"/>
        </target>

        <target id="tss_rev">
            <source id="tis_rev" out_feat="5UTR" mindis="1" out_str="-"/>
        </target>
    </model>

    <lengthfunctions>
        <!-- lengthfunc id="intron_pen" file="./tables/intron_penalty"/>
        <lengthfunc id="initial_exon_pen" file="./tables/exon_penalty.initial"/>
        <lengthfunc id="terminal_exon_pen" file="./tables/exon_penalty.terminal"/>
        <lengthfunc id="internal_exon_pen" file="./tables/exon_penalty.internal"/-->

        <!--lengthfunc id="single_exon_gene_pen">
            <point x="500" y ="0.001"/>
            <point x="20000" y="0.2"/>
        </lengthfunc-->

        <!--lengthfunc id="intergene_pen">
            <point x="200000" y ="0.01"/>
            <point x="200001" y="0.01"/>
        </lengthfunc -->
    </lengthfunctions>
</gaze>
```

The configuration file explaining the gene model without translation features for predicting genes using GenePred –

```
<?xml version="1.0" encoding="US-ASCII"?>

    <gaze>
        <declarations>
            <feature id="tss" st_off="0" en_off="1" />
            <!--feature id="tis" st_off="0" en_off="3"/-->
            <feature id="5ss" st_off="1" en_off="1" />
            <feature id="3ss" st_off="1" en_off="1" />
            <!--feature id="tts" st_off="3" en_off="0"/-->
            <feature id="polyA" st_off="1" en_off="1"/>

            <feature id="tss_rev" st_off="1" en_off="0" />
            <!--feature id="tis_rev" st_off="3" en_off="0" /-->
            <feature id="5ss_rev" st_off="1" en_off="1" />
            <feature id="3ss_rev" st_off="1" en_off="1" />
            <!--feature id="tts_rev" st_off="0" en_off="3" /-->
            <feature id="polyA_rev" st_off="1" en_off="1"/>

            <!--lengthfunction id="intron_pen" />
            <lengthfunction id="intergene_pen" />
            <lengthfunction id="inital_exon_pen" />
            <lengthfunction id="internal_exon_pen" />
            <lengthfunction id="terminal_exon_pen" />
            <lengthfunction id="single_exon_gene_pen" /-->
        </declarations>

        <gff2gaze>
            <!-- Features -->
            <gfffeat feature="TSS" strand="+" source="Eponine">
                <feat id="tss"/>
            </gfffeat>

            <gfffeat feature="TSS" strand="-" source="Eponine">
                <feat id="tss_rev"/>
            </gfffeat>

            <!--gfffeat feature="TIS" strand="+" source="Eponine">
                <feat id="tis"/>
            </gfffeat>

            <gfffeat feature="TIS" strand="-" source="Eponine">
                <feat id="tis_rev"/-->
            </gfffeat>

            <gfffeat feature="5SS" strand="+" source="Eponine">
                <feat id="5ss"/>
            </gfffeat>

            <gfffeat feature="5SS" strand="-" source="Eponine">
                <feat id="5ss_rev"/>
            </gfffeat>

            <gfffeat feature="3SS" strand="+" source="Eponine">
                <feat id="3ss"/>
            </gfffeat>

            <gfffeat feature="3SS" strand="-" source="Eponine">
                <feat id="3ss_rev"/>
            </gfffeat>

            <!--gfffeat feature="TTS" strand="+" source="Eponine">
                <feat id="tts"/>
            </gfffeat>

            <gfffeat feature="TTS" strand="-" source="Eponine">
                <feat id="tts_rev"/-->
            </gfffeat>

            <gfffeat feature="POLYA" strand="+" source="Eponine">
                <feat id="polyA"/>
            </gfffeat>

            <gfffeat feature="POLYA" strand="-" source="Eponine">
                <feat id="polyA_rev"/>
            </gfffeat>
        </gff2gaze>

        <dna2gaze>
            <!--dnafeat pattern="tataaa">
                <feat id="tss" />
            </dnafeat>

            <dnafeat pattern="atg" score="0.001">
                <feat id="tis" />
            </dnafeat>

            <dnafeat pattern="taa" score="0.001">
                <feat id="tts" />
            </dnafeat>

            <dnafeat pattern="tag" score="0.001">
                <feat id="tts" />
            </dnafeat>

            <dnafeat pattern="tga" score="0.001">
                <feat id="tts" />
            </dnafeat>

            <dnafeat pattern="aataaa" score="0.001">
                <feat id="polyA" />
            </dnafeat>

            <dnafeat pattern="tttata">
                <feat id="tss_rev" />
            </dnafeat>
            <dnafeat pattern="cat" score="0.001">
                <feat id="tis_rev" />
            </dnafeat>

            <dnafeat pattern="tta" score="0.001">
                <feat id="tts_rev" />
            </dnafeat>

            <dnafeat pattern="cta" score="0.001">
                <feat id="tts_rev" />
            </dnafeat>

            <dnafeat pattern="tca" score="0.001">
                <feat id="tts_rev" />
            </dnafeat>

            <dnafeat pattern="tttatt" score="0.001">
                <feat id="polyA_rev" />
            </dnafeat-->

            <!--takedna id="5ss_1" st_off="0" en_off="1"/>
            <takedna id="3ss_1" st_off="1" en_off="-1"/>
            <takedna id="5ss_2" st_off="-1" en_off="1"/>
            <takedna id="3ss_2" st_off="1" en_off="0"/>
            <takedna id="5ss_1_rev" st_off="1" en_off="0"/>
            <takedna id="3ss_1_rev" st_off="-1" en_off="1"/>
            <takedna id="5ss_2_rev" st_off="1" en_off="-1"/>
            <takedna id="3ss_2_rev" st_off="0" en_off="1"/-->
        </dna2gaze>

        <model>
            <target id="END">
                <source id="BEGIN" out_feat="No_genes"/>
                <source id="polyA" out_feat="GEN_DNA" />
                <source id="tss_rev" out_feat="GEN_DNA"/>
            </target>

            <!--Forward strand gene-->
            <target id="tss">
                <source id="BEGIN" out_feat="GEN_DNA"/>
                <source id="polyA" mindis="1" out_feat="intergenic"/>
                <source id="tss_rev" mindis="1" out_feat="intergenic"/>
            </target>

            <!--target id="tis">
                <source id="tss" mindis="1" out_feat="5UTR" out_str="+"/>
            </target-->

            <target id="5ss">
                <!--killfeat id="tts"/-->
                <!--source id="tis" out_feat="inital_exon" mindis="3" maxdis= "10000" out_str="+"
            /-->
                <source id="tss" mindis="1" out_feat="initial_exon" out_str="+"/>
```

```
            <source id="3ss" out_feat="internal_exon"  mindis="6" maxdis= "10000"
        out_str="+" />
        </target>

        <target id="3ss">
            <source id="5ss" out_feat="intron"  mindis="6" out_str="+"/>
        </target>

        <!--target id="tts"-->
            <!--killfeat id="tts" /-->
            <!--source id="tis" out_feat="single_exon_gene" mindis="60" out_str="+"/-->
            <!--source id="3ss" out_feat="terminal_exon"  mindis="3" out_str="+"/>
        </target-->

        <target id="polyA">
            <!--source id="tts" out_feat="3UTR" mindis="1" out_str="+"/-->
            <source id="3ss" out_feat="terminal_exon"  mindis="3" out_str="+"/>
        </target>

        <!--Reverse strand gene-->

        <target id="polyA_rev">
            <source id="BEGIN" out_feat="GEN_DNA"/>
            <source id="polyA" out_feat="intergenic" mindis="1"/>
            <source id="tss_rev" out_feat="intergenic" mindis="1"/>
        </target>

        <!--target id="tts_rev">
            <source id="polyA_rev" out_feat="3UTR" mindis="1" out_str="-"/>
        </target-->

        <target id="3ss_rev">
            <!--killfeat id="tts_rev" phase="0"/-->
            <source id="polyA_rev" out_feat="terminal_exon" mindis="3" maxdis= "10000"
        out_str="-"/>
            <source id="5ss_rev" out_feat="internal_exon" mindis="6" maxdis= "10000"
        out_str="-"/>
        </target>

        <target id="5ss_rev">
            <source id="3ss_rev" out_feat="intron" mindis="6" out_str="-"/>
        </target>

        <!--target id="tis_rev"-->
            <!--killfeat id="tts_rev" phase="0"/-->
            <!--source id="tts_rev" out_feat="single_exon_gene" mindis="60" out_str="-"/-->
            <!--source id="5ss_rev" out_feat="initial_exon" mindis="3" out_str="-"/>
        </target-->

        <target id="tss_rev">
            <source id="5ss_rev" out_feat="initial_exon" mindis="1" out_str="-"/>
        </target>
    </model>
    <lengthfunctions>
        <!-- lengthfunc id="intron_pen" file="./tables/intron_penalty"/>
        <lengthfunc id="initial_exon_pen" file="./tables/exon_penalty.initial"/>
        <lengthfunc id="terminal_exon_pen" file="./tables/exon_penalty.terminal"/>
        <lengthfunc id="internal_exon_pen" file="./tables/exon_penalty.internal"/-->

        <!--lengthfunc id="single_exon_gene_pen">
            <point x="500" y ="0.001"/>
            <point x="20000" y="0.2"/>
        </lengthfunc-->

        <!--lengthfunc id="intergene_pen">
            <point x="200000" y ="0.01"/>
            <point x="200001" y="0.01"/>
        </lengthfunc -->
    </lengthfunctions>
</gaze>
```

APPENDIX A: DOMAIN INSERTION

A.1 Introduction

Taking advantage of an evolutionary basis of domain classification, here I describe the nature and characteristics of domain insertions in protein structures, a phenomenon that is different from the usual pattern of sequential arrangement of domains in multi-domain proteins.

Domains constitute the basic structural, functional and evolutionary unit of proteins (Holm and Sander, 1996; Murzin *et al.,* 1995; Orengo *et al.,* 1997). Proteins can comprise a single domain or a combination of domains. It is well established that multi-domain proteins with widely diversified architecture and functions are generated from a limited repertoire of domain families (Bork *et al.,* 1996; Chothia, 1992). Structural assignments to complete genomes revealed that almost two-thirds of prokaryotic proteins and 80% of eukaryotic proteins are multi-domain proteins (Teichmann *et al.,* 1998). In 1973, Donald Wetlaufer introduced the classification of domains into continuous and discontinuous (Wetlaufer, 1973). A continuous domain is formed by one part of a polypeptide chain, while a discontinuous domain is formed by two or more parts of a single polypeptide chain. Thus, discontinuous domains are essentially formed by one-dimensionally non-contiguous segments of a polypeptide. While most multi-domain proteins have continuous domains, some proteins exhibit non-contiguous arrangement of their domains (Wetlaufer, 1973). In this work, I focus on insertions (Russell, 1994), which are the cases of one domain being inserted into another domain (Figure 67).

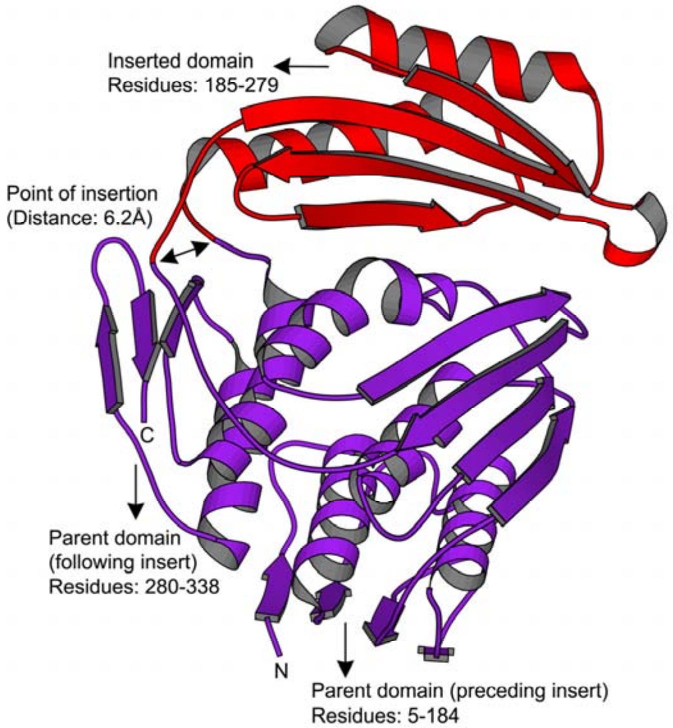


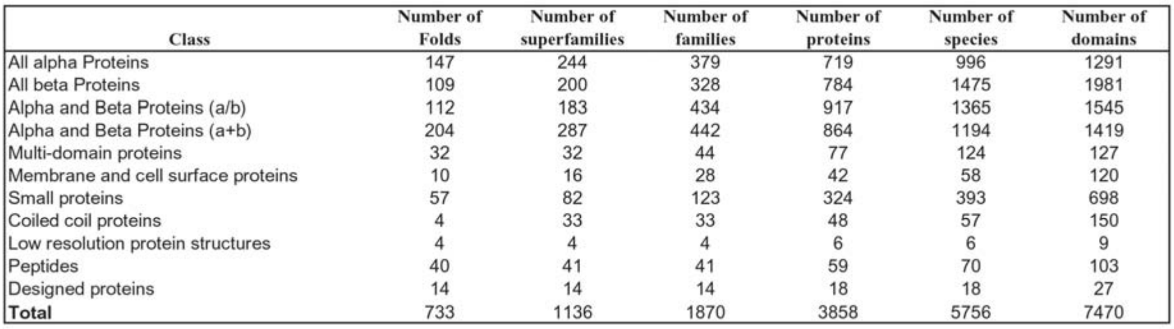
Figure 67. Domain insertion in Escherichia coli enzyme RNA 3'-terminal phosphate cyclase (PDB 1qmhA). The E. coli *enzyme RNA 3'-terminal phosphate cyclase consists of two domains, of which one is contained within the other. The parent domain (residues 5-184, 280-338, coloured purple) consists of three repeated folding units; each unit has two α-helices and a four-stranded β-sheet. The folding unit resembles the C-terminal domain of bacterial translation initiation factor 3 (IF3). Between an α-helix and a β-strand of the third IF3-like repeat of the parent domain, there is a smaller inserted domain (residues 185-279, coloured red). Although the inserted domain has the same secondary structural elements as the parent domain, it has different topology and a different fold. Insert resembles the fold observed in human thioredoxin.*

I followed the definition of protein domains in the Structural Classification Of Proteins (SCOP) database (version 1.61) (Murzin *et al.*, 1995). Although there are several available schemes of protein structure classification, I chose SCOP because it is a manually curated classification of protein structures based on their structural and evolutionary relationship. In SCOP, a protein domain is considered as a unit of evolution if it occurs independently or in combination with other domains.

SCOP represents a hierarchical classification scheme with four principal levels: family, superfamily, fold and class. Domains clustered into families are evolutionarily related and can be detected at the sequence level. Domains grouped into superfamilies can have low sequence identity but their structural and functional features suggest a common evolutionary origin. Superfamilies with similar topology are grouped under a fold. Folds are assigned to classes based on their secondary structure. For my analysis, I considered the fold and superfamily levels of SCOP hierarchy and the five major classes (all-α, all-β, α/β, α+β and 'small proteins'). All-α and all-β classes include proteins with abundant α-helices or β-sheets, respectively. The α/β class is distinguished mainly by parallel beta sheets (β-α-β units), whereas the α+β class contains proteins with predominantly anti-parallel beta sheets (segregated α and β regions). Small proteins are distinguished by their size rather than other features.

Data for this analysis was obtained from the Protein Data Bank (PDB) (Berman *et al.*, 2002). To overcome the redundancy inherent in PDB, I chose a pre-computed list of non-redundant protein chains provided by PDB_Select (April 2002 release obtained from *ftp://ftp.embl-heidelberg.de/pub/databases/protein_extras/pdb_select*) (Hobohm and Sander, 1994). I used the set of proteins that had pair-wise sequence identities less than 90% and designated this set as PDB_90. Out of the 6182 chains in PDB_90, only 5883 chains were assigned SCOP domain definitions, extracted from the SCOP parseable file *dir.cla.scop.txt_1.61*. Table 24 shows the distribution of SCOP folds, superfamilies, families and domains in each class for chains present in PDB_90.

Table 24. SCOP (1.61 release) classification statistics for chains in PDB_90 (April 2002 release)



Class	Number of Folds	Number of superfamilies	Number of families	Number of proteins	Number of species	Number of domains
All alpha Proteins	147	244	379	719	996	1291
All beta Proteins	109	200	328	784	1475	1981
Alpha and Beta Proteins (a/b)	112	183	434	917	1365	1545
Alpha and Beta Proteins (a+b)	204	287	442	864	1194	1419
Multi-domain proteins	32	32	44	77	124	127
Membrane and cell surface proteins	10	16	28	42	58	120
Small proteins	57	82	123	324	393	698
Coiled coil proteins	4	33	33	48	57	150
Low resolution protein structures	4	4	4	6	6	9
Peptides	40	41	41	59	70	103
Designed proteins	14	14	14	18	18	27
Total	733	1136	1870	3858	5756	7470

It is self-evident that insertions can only be found in multi-domain proteins, where one domain (insert) is contained within another domain (parent). Parent and insert domains can belong to the same or different SCOP superfamilies. Likewise, a combination of two domains can be viewed as a combination of superfamily combinations. I obtained a total of

140 proteins that conformed to this definition. When I considered the 140 pairs of parent-insert superfamily combinations, I observed that several pairs were identical. Whenever there was also the same topological relationship between the parent and insert domains, I retained only one example of a pair of superfamily combinations. This procedure left 40 unique parent-insert superfamily combinations. Variations on the simple scheme ‘one insert within one parent’ were present; they are shown in Figure 68.

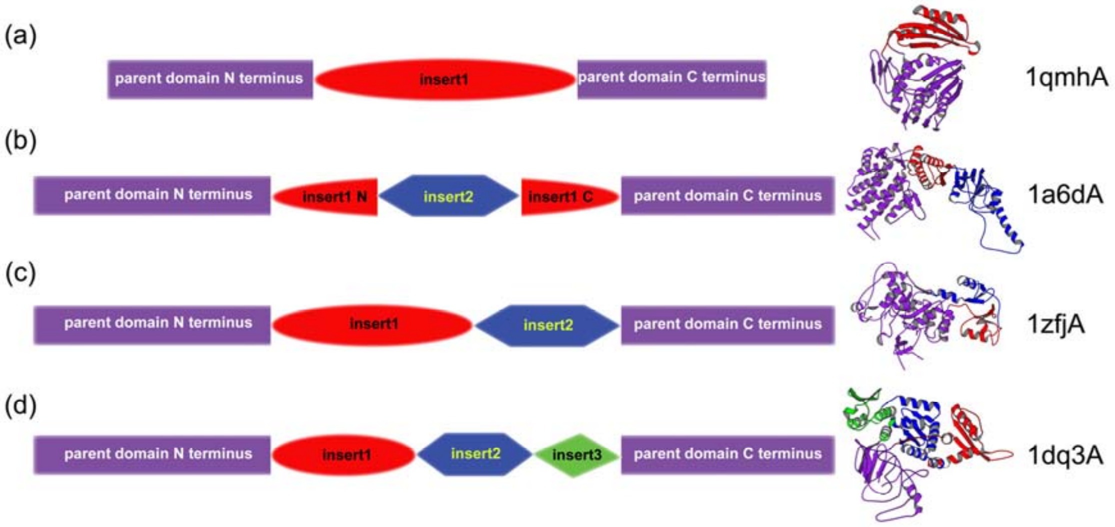


Figure 68. Schematic representation of types of domain insertions observed in protein structures. (a) Single insertion (e.g., 1qmhA). (b) Nested insertion (e.g., 1a6dA). 'insert1 N' and 'insert1 C' represent the N- and C-terminus of insert, respectively. (c) Two-domain insertion (e.g., 1zfjA). (d) Three-domain insertion (e.g., 1dq3A).

For all cases of identified domain insertions, I checked for artefacts arising from missing coordinates. This was necessary because SCOP domain definitions are based on atomic coordinates provided in PDB. To ascertain consistency, I compared atomic coordinates (ATOM records) *versus* sequences (SEQRES records) that were obtained from the ASTRAL compendium (Chandonia *et al.*, 2002). In the majority of cases, sequences were completely covered by coordinates, but in other cases, there were parts of sequences with missing coordinates. However, in none of the latter cases did the absent coordinates obscure the position of inserts.

I then calculated unique superfamily combinations for all multi-domain proteins and found 450 unique superfamily combinations for 5883 single or multi-domain proteins in SCOP. Thus, domain insertions constitute 9% (40/450) of all unique superfamily occurrences.

A.2 Types of domain insertions

Domain insertions can be categorized as either single or multiple depending on the number of inserts (Figure 68). In single insertions, one domain is inserted into another domain, and both domains can belong to the same or different superfamilies. For example, in Figure 68a, the *Escherichia coli* enzyme RNA 3'-terminal phosphate cyclase (PDB: 1qmhA, Palm *et al.*, 2000) has two domains, a small insert and a larger parent that belong to different superfamilies. Close to 90% (36/40) of observed insertions are single insertions. In multiple insertions, more than one domain, either of the same or different superfamily, is inserted into the parent domain. I observed three types of multiple insertions (i) Nested insertions: In *Thermoplasma acidophilum* thermosome (PDB: 1a6dA, Ditzel *et al.*, 1998), the archael chaperonin, the apical domain is inserted into the intermediate domain, which is in turn inserted into an ATPase domain (ii) Two-domain insertions: The type II inosine monophosphate dehydrogenase from *Streptococcus pyogenes* (PDB: 1zfjA, Zhang *et al.*, 1999) contains two tandem cystathionine-β-synthase domains inserted into the catalytic TIM-barrel domain. The second example is the *Saccharomyces cerevisiae* PI-*Sce*I intein (PDB: 1ef0A, Poland *et al.*, 2000), a homing endonuclease with protein splicing activity, which has the duplicated endonuclease domain inserted into the Hint domain (iii) Three-domain insertions: In PI-*Pfu*I, an intein-encoded homing endonuclease from the archaebacteria *Pyrococcus furiosus* (PDB: 1dq3A, Ichiyanagi *et al.*, 2000), the Hint domain has three tandem inserts, two intein endonuclease domains with αββαββαα structural motifs, and one Stirrup domain.

Previous work on intron-encoded homing endonucleases, from the dodecapeptide family, showed that for their folding, dimerisation and catalysis, they should form a dimer that has two copies of the LAGLIDADG motif (one copy per subunit of a dimer), or alternatively they could be monomeric if a monomer has both copies of the motif (Jurica and Stoddard, 1999). I found that in PI-*Sce*I (case [ii] above) and PI-*Pfu*I (case [iii] above), two monomeric domains were tandemly inserted into one parent domain. The previous observation that motifs are only functional as a dimer suggests that during the course of evolution, there was a simultaneous insertion of two monomeric domains into the parent domain, rather than an insertion of one monomeric domain followed by its duplication.

In this analysis, I treated multiple insertions as several separate parent-insert combinations, resulting in the total of 45 such combinations within 40 protein chains. There were 41 unique parent-insert superfamily combinations. Upon examination of relationships among proteins containing insertions, levels of SCOP hierarchy, and superfamily participation of parent and inserted domains, I identified several biologically meaningful patterns. These findings are discussed below.

A.3 Nature and characteristics of domain insertions: Class level

As mentioned before, I considered five SCOP classes, leading to a maximum of 25 (5*5) pair-wise combinations. From the data, I observed only 15 combinations when investigating class participation of parent-insert pairs. The combination of α/β-parent-α+β-insert was predominant, while 50% of all parents belonged to α/β class and 40% of all inserts belonged to α+β class. Domains from α/β class were parent domains, which were two and four fold more often than domains from all-β and all-α class respectively. Domains from the class of small proteins were seen only as inserts. This bias could be explained, at least to a certain extent, by taking into consideration the size and function of parents and inserts, which is discussed in the next section.

A.3.1 Size and function of domains involved in insertions

Figure 69a shows the domain length distribution for proteins from PDB_90 set across the five SCOP classes. The average domain length was longest for α/β class followed by the all-β, α+β, and all-α class. When I calculated distribution of average domain lengths for 41 parent domains, I observed the same trend (Figure 69b). However, the average length of parent domains was noticeably larger than the average length of domains from PDB_90 set; this was true for each SCOP class (compare Figure 69a and Figure 69b). Thus, combining the fact that α/β parent domains are the most abundant with the fact that α/β domains are the longest on average, I arrived at the explanation that longer domains more readily accept insertions during evolution. As for the inserted domains, α+β and all-α class were equal and major contributors to the number of domains. Therefore, the trend observed for parents is not applicable for inserts.

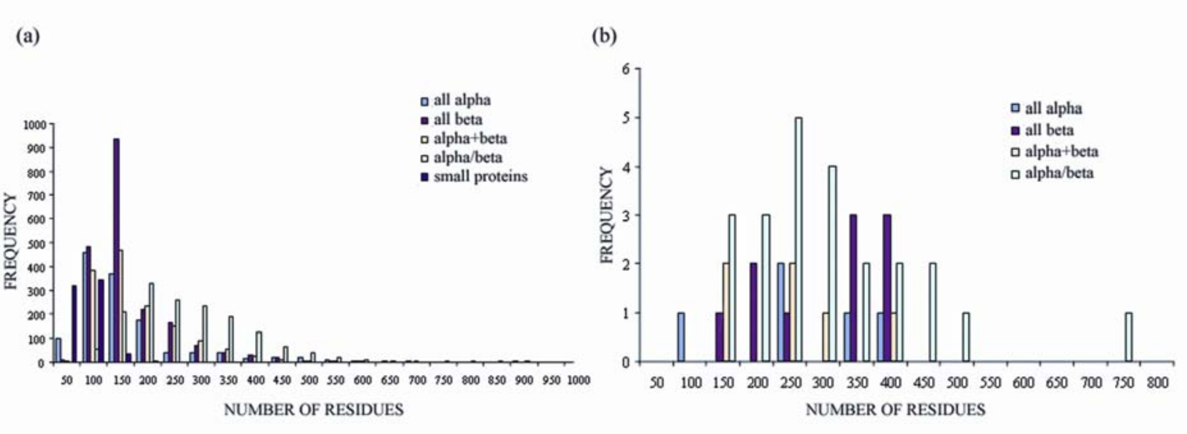


Figure 69. (a) Domain length distribution for all domains in the non-redundant set of proteins (PDB_90). (b) Domain length distribution for parent domains.

In most cases, inserted domains were shorter than parent domains. This is despite the fact that inserted domains could belong to SCOP classes with the longest average domain length (Figure 70a). Parents comprised 50-80% of protein length, while inserts comprised 20-50%. Close to 80% of inserts were shorter than 175 residues, which is the average length of a protein domain calculated from crystal structures (Gerstein, 1997). More than 60% of inserts were shorter than 130 residues. This observation is consistent with the heuristic logic that smaller domains are less likely to disturb the structure and folding of parent domains; it could explain short lengths of inserted domains. This explanation does not contradict an important experiment by Doi and colleagues (Doi *et al.*, 1997). They were able to show that when random sequences of 120-130 amino acid residues were inserted into a surface loop region of *Escherichia coli* RNase HI, about 10% of the clones retained >1% of the wild-type RNase HI activity (Doi *et al.*, 1997).

The high proportion of α/β class domains, as parents, can be correlated with their biochemical function. Previous work showed that more than a half of PDB families are enzymes and close to one half of all enzyme families contain multi-domain proteins. Multi-domain enzymes often consist of a catalytic domain and a nucleotide binding domain (Hegyi and Gerstein, 1999). It is therefore possible to predict that domain insertions are likely to occur in enzymes. Indeed, in the dataset, 39 out of 40 parent-insert pairs conform to this prediction. The remaining non-enzymatic protein is the bluetongue virus capsid protein vp-7, which has the central domain from all-β class inserted into the multi-helical parent domain. A genome-scale analysis of the structural features of proteins revealed that proteins with α/β fold are frequently involved in fusion events (Hua *et al.*, 2002). α/β folds are also known to be disproportionately associated with enzymatic function (Hegyi and Gerstein, 1999), which lends further credence to the prominent role of α/β folds in accepting insertions.

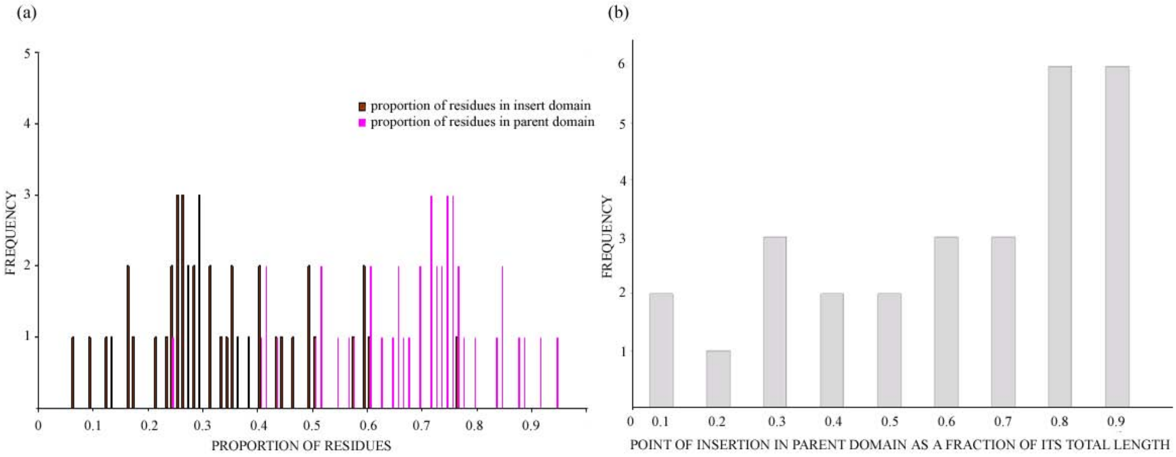


Figure 70. (a) Proportion of residues in parent and insert domains in parent-insert combinations. (b) Point of insertion in parent domain. Insert position is given as a fraction of total length of parent domain.

A.4 Nature and characteristics of domain insertions: Fold and superfamily level

Out of 57 folds in the class of small proteins, two domains with one fold (Rubredoxin fold) were found as inserts; both inserted domains belong to the same superfamily. Within the α+β class, the 18 inserted domains (from 15 superfamilies) spanned 11 folds; there are 204 different folds in the α+β class (Table 25). The trend was the same for the other SCOP classes, where folds of inserted domains constituted minor fractions of all known folds. In contrast to the inserts, all parent domains had different folds. Thus, I observed another distinction between parents and inserts at the fold level.

Similarly, parent superfamilies were found to be more versatile than insert superfamilies (most insert superfamilies combine with only one parent superfamily). There are merely 3 out of 45 insert superfamilies that combine with two different parent superfamilies. These insert superfamilies are NAD(P)-binding Rossmann superfamily, FAD/NAD(P)-binding superfamily and C-terminal domain of FAD-linked reductases superfamily.

Table 25. Distribution of inserted and parent domains at the SCOP class and fold level. The number of domains and the number of folds they come from is given for inserted and parent domains across the five different classes in the SCOP hierarchy. Percentage gives the number of folds contributing to insertions over total number of folds under the class.

SCOP Class	Total number of folds	Inserted domains			Parent domains		
		Number of domains	Number of folds	Percentage of folds	Number of domains	Number of folds	Percentage of folds
All-α	147	6	5	3.4	5	5	3.4
All-β	109	9	9	8.3	11	11	9.2
α/β	112	10	6	5.4	23	23	20.6
α+β	204	18	11	5.4	6	6	3
Small proteins	57	2	1	1.8	0	0	0

While many parent superfamilies conservatively combine with one insert superfamily, there are conspicuous exceptions. There are three parent superfamilies each combining two different insert superfamilies. The three parent superfamilies in question are Zn-dependent exopepetidases superfamily, nucleotidyl transferase superfamily, and nucleotide-binding domain superfamily. Moreover, there are two parent superfamilies each combining with three different insert superfamilies. The two parent superfamilies are P-loop containing NTP hydrolases superfamily, and FAD/NAD(P)-binding domain superfamily.

Two further observations at the superfamily level are worth mentioning. Firstly, all parents and inserts belong to different superfamilies. There is only one exception: in *Escherichia coli* enzyme glutathione reductase (PDB: 1gesB), the parent and insert belong to the same superfamily of FAD/NAD(P)-binding domains. Secondly, superfamilies that are popular in the parent or insert context also appear to be popular in the sequential domain combination context (Apic *et al.*, 2001). They were found combining with more than one superfamily in the sequential domain order. One exception to this correlation is the superfamily of C-terminal domains of FAD-linked reductases; this superfamily is popular in the insert context, but does not tandemly combine with other superfamilies.

A.5 Point of insertion

I did not find any bias in the distribution of insertion points within 41 unique parent-insert combinations. However, a significant bias in the location of the insertion point was observed when I considered a subset of 28 parent-insert combinations, where either the parent or insert superfamily also participated in sequential combination with other superfamilies. As shown in Figure 70b, for the 28 cases in question, the insertion point occurred in the last third part of the parent domain sequence (confidence level 98%). Spatially, all 41 insertions were observed in loop regions of the 3D structure of parent domains.

Though it may not be feasible to provide a definitive explanation for the observation of bias towards C-terminus for insertion in the parent domain, an event in the N-terminus or the middle of the domain are likely to disrupt the gene structure and pose a problem during transcription or translation.

Also insertions in the C-terminus indicate most of the insertions seen in the database are not *strictly* insertions but normal sequential combinations with the second domain starting before the end of the first domain. This stem from the fact, C-terminus bias in insertion is found only in cases of parent-insert combinations, where either the parent or insert also occur in sequential combinations with other superfamilies. Further research on the domain insertions involving the core structure of the parent and insert domains can throw more light on this view.

A.6 Proximity of N- and C-termini in inserts

I wanted to determine how the insertion context affects the distance between N- and C-terminus of an inserted domain. The distance between termini was defined as the distance between C-alpha atoms of the first and the last residue of the domain. I first calculated distances for domains that do not participate in insertions. In order to do this, I considered 1000 domains, each representative of one SCOP superfamily. I obtained sequences and coordinates for the domains from the ASTRAL compendium (Chandonia *et al.*, 2002). Only 687 domain sequences were completely covered by coordinates. Using AEROSPACI scores (Chandonia *et al.*, 2002), I was able to find 60 substitutes for the 313 representative domains that were not entirely covered by coordinates. Altogether, I obtained complete coordinate information for 747 domains (687 + 60). Because I confined the analysis to five major SCOP classes, I calculated distances between termini for the 711 domains, which belong to the five classes being investigated. The average distance for representative domains was 25 Å.

Calculation of distances between the termini of inserted domains was less straightforward. Domain boundaries reported in SCOP are human defined. Therefore, I compared SCOP domain boundaries for 41 inserted domains against the domain boundaries reported in CATH database (Orengo *et al.*, 2002). In contrast to SCOP, CATH structural classification of proteins has been produced automatically. However, only 28 out of 41 inserted domains were available in CATH, whereas the other 13 have either differences in domain classification or the corresponding proteins were absent from CATH classification. For 28 inserted domains, boundaries were identical between SCOP and CATH. The average distance between domain termini of inserted domains was 8 Å (confidence level 99%), which is two-thirds shorter than the distance between termini in normal domains.

There are two superfamilies that occur in both parent and insert context. This example allowed me to compare distances between termini for a parent and an insert from the same superfamily. In case of FAD/NAD(P)-binding domain superfamily, the distances were 30 Å and 5 Å for parent and an insert, respectively. These figures were 11 Å and 8 Å for NAD-binding Rossmann domain superfamily. Thus, this analysis shows that the ends of inserted domains are significantly closer than ends of parent domains or domains not participating in insertions. However one must be cautious in interpreting the results as the N and C termini distances for the parent domain is not calculated for the core structure.

It is interesting to speculate how the distance between domain termini can affect stability and conformational flexibility of a protein domain. While insertion context might generally reduce conformational freedom of the domain, it can simultaneously contribute to the stability of the domain, which would in turn affect its function. One can also imagine how the close proximity of domain termini can restore protein conformational flexibility by mimicking an inter-domain link observed in sequentially ordered domains.

A.7 Conclusions

Utilising an evolutionary basis of domain classification, I described the nature and characteristics of domain insertions in protein structures. Domain insertions represent an unusual but abundant case of multi-domain proteins. This analysis gave several novel insights into the nature and characteristics of domain insertions.

(1) Close to 9% multi-domain proteins contain insertions.

(2) The majority of insertions are the single domain insertions. Also found there were two-domain, three-domain, and nested insertions in PDB.

(3) α/β class has a higher propensity to accept insertions. This could be correlated to the size and function of proteins within the class.

(4) Parent domains were found to be longer than the inserted domains in most cases.

(5) When fold and superfamily combinations were considered for parents and inserts, the former was found to be more versatile than the latter, in that the parent domains combined with more partners.

(6) The point of insertion is biased towards the C-terminus of parents whenever the parent domain belongs to the superfamily that sequentially combines with other superfamilies.

(7) Inserted domains have juxtaposed termini compared to parent domains.

Perhaps, domains are more viable in the insert context when their termini are close in space; small size can further contribute to their viability.

These results clearly indicate that despite the structural and functional constraints inherent in the process of domain insertion, this process is an effective way of creating multi-domain proteins. This description of the many features of domain insertions could be used in protein engineering for producing novel multi-functional fusion proteins. Betton and co-workers (Betton *et al.*, 1997) created hybrid proteins by inserting a penicillin-hydrolysing enzyme TEM beta-lactamase (Bla) into the maltodextrin-binding protein (MalE); they used the permissive insertion sites identified before (Duplay *et al.*, 1987). Two insertions resulted in the functional hybrids, one insertion occurred in the first quarter of the MalE protein, while the other occurred in the last quarter. The parent protein (MalE) belongs to the α/β class, and the authors experimentally showed the 5 Å distance between the termini of the inserted domain (Bla). Thus, there is recent experimental data that nicely fit into the picture of insertions found in natural multi-domain proteins.

APPENDIX B: PROTEIN EVOLUTION

B.1 Introduction

Divergence in structure and function of proteins is due to an evolutionary process driven by functional and environmental constraints. These constraints bring about changes in the protein sequence through mutations, insertions and deletions with the preservation of residues important for the structure and function of the protein (Chothia and Lesk, 1986). However, not all the sequence modifications are incorporated or maintained since some changes may be deleterious to the structure or function of the protein. Hence, the structural *'core'* (Chothia and Lesk, 1986) tends to be well conserved during evolution. When proteins evolve, the constraints on the protein structure are relaxed or rather replaced by new constraints and the sequence and structure can change more radically. These changes are generally slow processes and leave a trail of *homologs*. Homologs are proteins evolved from a common ancestor and their evolutionary relationship is evident from similarities in sequence, structure and function. Homologous proteins have been studied for a long time to understand their evolutionary relationships and to assign function or structure to new protein sequences. For homolog searches in the sequence databases, one needs an alignment algorithm, residue similarity matrix, scoring scheme and knowledge about scoring thresholds to identify true relationships.

Among the available pairwise alignment algorithms, one of the most sensitive is the Smith-Waterman algorithm (Smith and Waterman, 1981) adopted in the SSEARCH program (Pearson, 1991). Although this algorithm is more sensitive and rigorous, it is computationally expensive in comparison to FASTA (Pearson and Lipman, 1988) and BLAST (Altschul *et al.*, 1990). The speed and convenience of BLAST made it the most popular program, although it compromises sensitivity. FASTA ranks between these two programs and can be run in two modes: either at greater speed (ktup = 2) or greater accuracy (ktup =1). Pearson (Pearson, 1991, 1995) did a comparison of these three methods and showed that the Smith-Waterman algorithm worked slightly better than FASTA, which was in turn much more effective than BLAST.

Although pairwise comparison methods are a common way to find sequence homologs, they have difficulty in detecting remote homologs when sequence identity falls below 30% (Brenner *et al.*, 1998). Alternate methods like Profile Hidden Markov Models (Eddy, 1996; Krogh *et al.*, 1994), psi-BLAST (Altschul *et al.*, 1997) and Intermediate Sequence Search (Park *et al.*, 1997) reduce this limitation and increase sensitivity.

Intermediate Sequence Search (ISS) is a search technique, wherein two related sequences which cannot be detected directly by pairwise sequence comparison methods are matched using an intermediate sequence sharing close homology with the two distantly related sequences. This concept has been extended to include multiple intermediate sequences (MISS) between two distant sequences (Salamov *et al.*, 1999). The disadvantage with ISS is that the errors caused in the intermediate are likely to propagate as it is not dependent on multiple sequence alignment. Errors caused by ISS when comparing multi-domain protein sequences, can be avoided by splitting query sequence to individual domains. Figure 71 gives an overall idea on how different methods are exploring the sequence space (Lindahl and Elofsson, 2000).

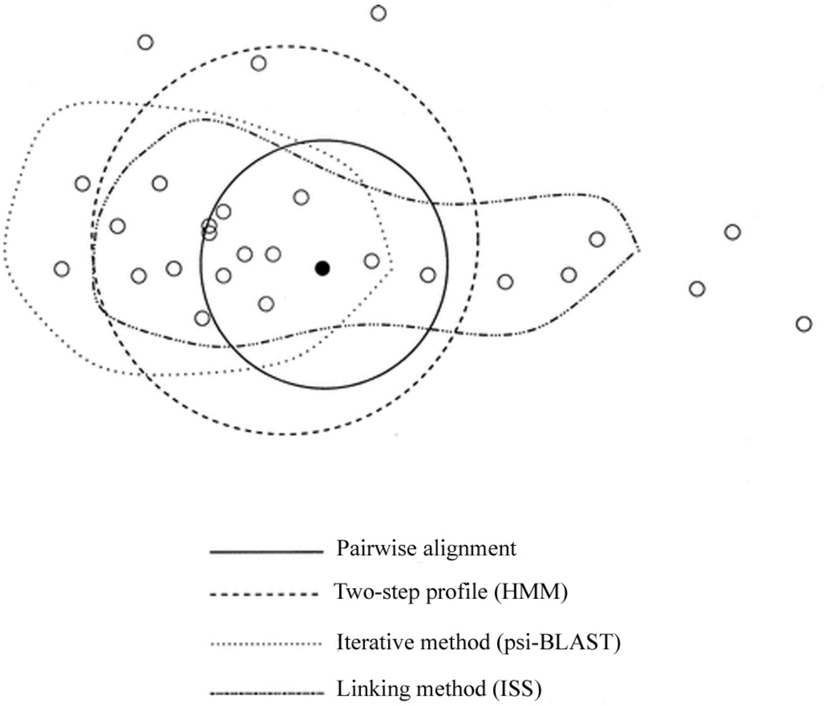


Figure 71. Schematic diagram showing performance of different sequence comparison methods. The filled circle represents the query sequence used in the database search and the open circles represent family members. The distance between two circles represents some arbitrary distance.

A comparison of these recent methods with pairwise sequence comparison methods, performed by searching remote homologs in a Structural Classification Of Proteins (SCOP, Murzin *et al.*, 1995) sequence database having less than 40% identity, show that ISS performs one and half times better than FASTA. In sequences with less than 30% identity, a HMM-based SAM-T98 and psi-BLAST detected three times more relationships than pairwise sequence comparison methods (Park *et al.*, 1998). Sauder *et al.* compared the quality of alignments produced by BLAST, psi-BLAST, ISS and ClustalW (Thompson *et al.*, 1994) with structural alignments. ISS produced longer alignments than psi-BLAST with nearly comparable per-residue alignment quality. At 10-15% identity, BLAST correctly aligned 28%, psi-BLAST 40% and ISS 46% of residues to the structural alignment (Sauder *et al.*, 2000).

All these results show that ISS performs as well as psi-BLAST in identifying distant homologs. However it is not yet clear how ISS is able to detect remote relationships. Moreover, I was interested to determine whether intermediates identified by ISS can provide any knowledge about protein evolution. This study tries to find answers to these questions.

To aid this objective, I also used structure comparisons to understand relationships between proteins. The degree of fitness between structures is usually calculated by a scoring scheme. The common way to represent the structural fitness is Root Mean Square Deviation (RMSD) for all residues of the two protein structures. The RMSD gives a measure of the average level of deviations over the superposed atoms.

$$\sqrt{\sum_{i=1}^{n} \frac{D_i^2}{N}}$$

Where, *D* refers to deviation of the atoms and *N* refers to the number of atoms matched.

There are different structural alignment methods adopting the aforementioned algorithms. Amongst the common implementations are DALI (Holm and Sander, 1993), Combinatorial Extension (CE) (Shindyalov and Bourne, 1998), and Protein Informatics System for Modelling (PrISM) (Yang and Honig, 2000). Here, I used PrISM to compare the structures.

Protein evolution may occur in two ways: divergent or convergent evolution. When a protein structure diverges to form a new fold or function, it results in divergent evolution (e.g., P-loops). However if two evolutionarily independent folds converge to represent similar structure or function it becomes convergent evolution (e.g., serine proteases). Proteins evolved through a divergent mechanism are likely to have a trail of homologs and can be detected using sequence and structure comparisons. Here, I attempt to study this using two well known protein families – *Cytochrome c* and *P-loops* and answer the following questions.

(1) Is it possible to understand the evolutionary pattern of any protein family or superfamily based solely on its structure and sequence divergence?

(2) Whether understanding this will help us in assigning hierarchies for a protein in the existing classification of protein structures?

B.2 Datasets

I used SCOP database for this study (please refer to Appendix A for details of SCOP). The *All-α* protein class contains a fold level called *cytochrome c*, which in turn is composed of a single superfamily named *cytochrome c*. This superfamily has four families. The *Di-haem cytochrome c peroxidase* family has only synthetic protein structures and, therefore, only domains from the other families (39 sequences) were used in this analysis.

P-loop domains are found in the class α/β and fold/superfamily *P-loop containing nucleotide triphosphate hydrolases* (this fold has only one superfamily). The superfamily has domains composed of parallel beta sheets of varied sizes connected by helices. For example, the *Nucleoside and nucleotide kinases* family has 5 strands with architecture type 23145 and *Nitrogenase iron-protein like group* family has 7 strands with architecture type 3241567. The superfamily is composed of 14 families. I used all the domains (85 sequences, excluding domains involving multiple chains) from these 14 families for this analysis.

From these datasets, I then found sequence homologs and structure homologs that can be detected by the above described methods.

B.3 Intermediate sequence search

I collected homologs for each of the domains in the two superfamily datasets using FASTA 3.3 (with BLOSUM 62 matrix, ktup = 1) by searching against the pdb90d_1.53 database. The pdb90d_1.53 database is derived from sequences of SCOP domains (version 1.53) sharing 90% or less sequence identity.

Domains (query and target), with scores better than the threshold value 0.01, are referred as ‘direct hits’. For domains that cannot be detected directly, I used the ISS procedure described above to link the query and target.

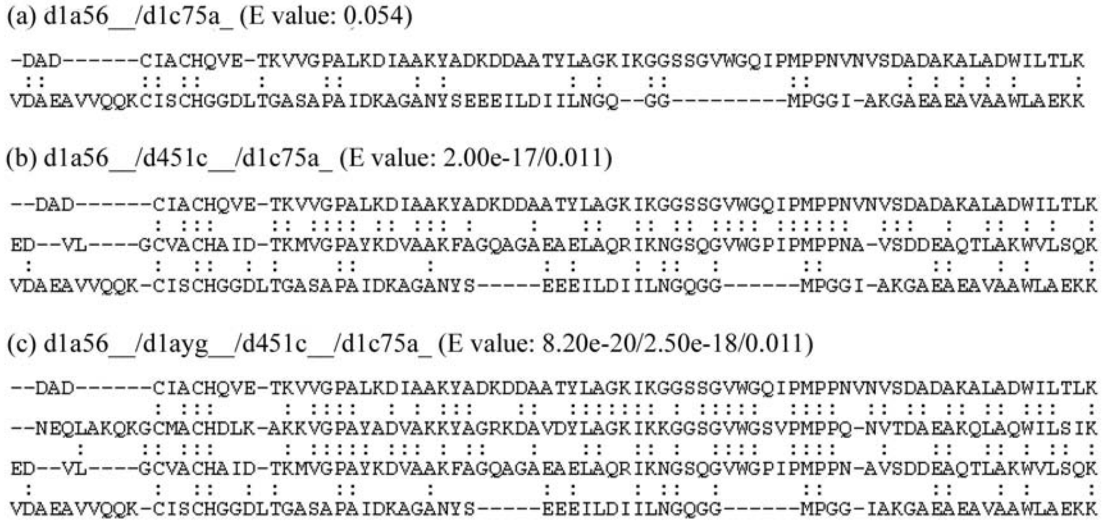
A comparison of ISS hits with psi-BLAST shows that psi-BLAST can detect all the remote homologs identified by ISS in P-loops superfamily and only about half of them in cytochrome c superfamily. The advantage ISS has in some cases might be due to the match score it gains by producing longer alignments around conserved regions of the protein. However, both the methods fail to detect remote homologs from P-loops superfamily than found from cytochrome c superfamily. This might be due to the extensive divergence of sequences in P-loops superfamily (they are quoted to have some converged domains (Bossemeyer, 1994) and differences in sequence length (average length of P-loops is ≈ 230 amino acids, twice the size of cytochrome c).

Intermediate searches based on structural information could find new remote homologs that ISS could not detect. This is expected because it is known that different sequences can have similar folds. Therefore, by comparing structures it is more likely to detect remote homologs. I suggest that by using intermediate structural search, even more distant relationships can be detected.

Then I used the alignments obtained from the query-intermediate and target-intermediate to generate a “progressive alignment” (i.e., a multiple sequence alignment generated by progressively aligning pairwise alignments using *ClustalW* alignments and structure information) of query-intermediate-target or query-intermediate-intermediate-target.

These progressive alignments show that the intermediates can improve the quality of alignments between query and target. An example of this alignment is shown in Figure 72.

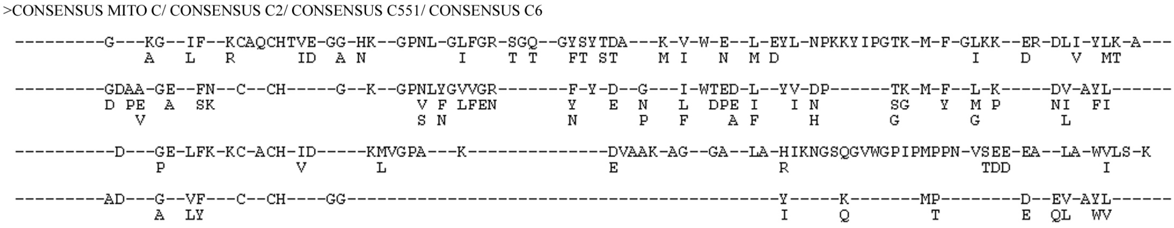
The figure shows the improvement in alignment between query-target (SCOP Ids: *d1a56__* - *d1c75a_*) produced by FASTA (Figure 72a) and the progressive alignment generated manually after introducing one (*d451c__*) and two intermediate (*d1ayg__* and *d451c__*) sequences (Figure 72b and Figure 72c). The alignment shows that there are some residues common in all the sequences and some between query-intermediate, target-intermediate and intermediate-intermediate.



```
(a) d1a56__/d1c75a_ (E value: 0.054)

-DAD------CIACHQVE-TKVVGPALKDIAAKYADKDDAATYLAGKIKGGSSGVWGQIPMPPNVNVSDADAKALADWILTLK
 ::       :: ::    :    ::     : :         : :   ::         ::       : : : : :    :
VDAEAVVQQKCISCHGGDLTGASAPAIDKAGANYSEEEILDIILNGQ--GG---------MPGGI-AKGAEAEAVAAWLAEKK

(b) d1a56__/d451c__/d1c75a_ (E value: 2.00e-17/0.011)

--DAD------CIACHQVE-TKVVGPALKDIAAKYADKDDAATYLAGKIKGGSSGVWGQIPMPPNVNVSDADAKALADWILTLK
           : :::    :: :::: :: ::: :    :   ::  :: :: :::: ::::::  :::  :  :: : :  :
ED--VL----GCVACHAID-TKMVGPAYKDVAAKFAGQAGAEAELAQRIKNGSQGVWGPIPMPPNA-VSDDEAQTLAKWVLSQK
 :         :  ::  : :    ::     :        : :    : ::  :      ::        ::   : :    :
VDAEAVVQQK-CISCHGGDLTGASAPAIDKAGANYS-----EEEILDIILNGQGG------MPGGI-AKGAEAEAVAAWLAEKK

(c) d1a56__/d1ayg__/d451c__/d1c75a_ (E value: 8.20e-20/2.50e-18/0.011)

--DAD------CIACHQVE-TKVVGPALKDIAAKYADKDDAATYLAGKIKGGSSGVWGQIPMPPNVNVSDADAKALADWILTLK
           : :::     : :::: : : :::   ::  ::::::: : :::::  ::::  :: :: :: :: :::  :
--NEQLAKQKGCMACHDLK-AKKVGPAYADVAKKYAGRKDAVDYLAGKIKKGGSGVWGSVPMPPQ-NVTDAEAKQLAQWILSIK
     :    :: :::     : ::::: ::: : ::   :   ::  :: :  ::::  ::::   :   ::  :: : :: :
ED--VL----GCVACHAID-TKMVGPAYKDVAAKFAGQAGAEAELAQRIKNGSQGVWGPIPMPPN-AVSDDEAQTLAKWVLSQK
 :         :  ::  : :    ::     :        : :    : ::  :      ::        ::   : :    :
VDAEAVVQQK-CISCHGGDLTGASAPAIDKAGANYS-----EEEILDIILNGQGG------MPGG-IAKGAEAEAVAAWLAEKK
```

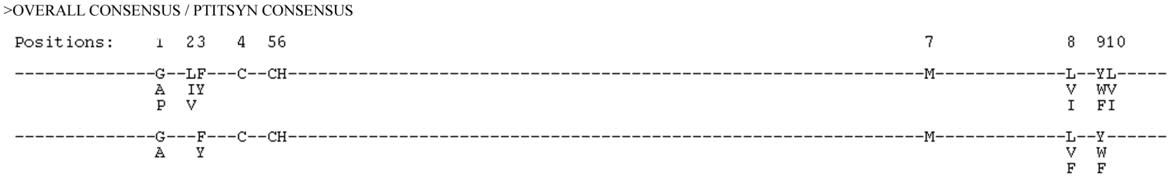
Figure 72. Comparison of alignments of two distant proteins with and without intermediates. (a) Alignment of the two domain produced by FASTA 3.3. (b) The progressive alignment generated by including one intermediate. (c) The progressive alignment generated by including two intermediates.

Likewise, I selected closely clustered domains from each of the four SCOP protein groups (*mitochondrial cytochrome*, *cytochrome c_2*, *cytochrome c_{551}* and *cytochrome c_6*) to make a progressive alignment. These groups were used due to the fact that they represent most of the members of the superfamily. From the progressive alignment made for each of the protein groups, I derived a consensus (Figure 73). This consensus was then used to derive an overall consensus shown in Figure 74. The figure shows that there are 10 invariable residues in the consensus and it agrees with the consensus derived by Ptitsyn by aligning 164 sequences from the cytochrome c superfamily (Ptitsyn, 1998). His alignments were generated using the PileUP program and manually edited taking functional residues into consideration.



```
>CONSENSUS MITO C/ CONSENSUS C2/ CONSENSUS C551/ CONSENSUS C6

---------G---KG--IF--KCAQCHTVE-GG-HK--GPNL-GLFGR-SGQ--GYSYTDA---K-V-W-E--L-EYL-NPKKYIPGTK-M-F-GLKK--ER-DLI-YLK-A---
             A   L   R      ID  A N         I    T T   FT ST    M I   N  M D                   I    D    V  MT

---------GDAA-GE--FN--C--CH-----G--K--GPNLYGVVGR-------F-Y-D--G---I-WTED-L--YV-DP------TK-M-F--L-K-----DV-AYL------
         D PE  A  SK                    V F LFEN       Y   E  N   L  DPE I   I N       SG   Y  M P     NI  FI
            V                           S N           N      P   F    A F     H       G       G        L

----------D---GE-LFK-KC-ACH-ID-----KMVGPA---K--------------DVAAK-AG--GA--LA-HIKNGSQGVWGPIPMPPN-VSEE-EA--LA-WVLS-K
              P             V       L                      E                R                   TDD         I

---------AD---G--VF---C--CH----GG-------------------------------------------Y-----K-------MP--------D--EV-AYL------
              A  LY                                                         I     Q        T        E  QL  WV
```

Figure 73. Consensus sequences derived for the four SCOP protein group in monodomain cytochrome c family



```
>OVERALL CONSENSUS / PTITSYN CONSENSUS

Positions:    1  23   4  56                                                               7             8  910

--------------G--LF---C--CH---------------------------------------------------------------M-------------L--YL-----
              A  IY                                                                                     V  WV
              P  V                                                                                      I  FI

--------------G---F---C--CH---------------------------------------------------------------M-------------L--Y------
              A   Y                                                                                     V  W
                                                                                                        F  F
```

Figure 74. Consensus of consensus for sequences in monodomain cytochrome c family

The conserved residues were involved in heme binding and needed for functional role of the protein. The other conserved residues do not have any functional role and are found to be key residues needed to maintain structural fold of cytochromes. The key residues reported here agree well with the results found in the literature (Ptitsyn, 1998). Figure 74 shows the key residues identified by Ptitsyn. The differences include two additional residues conserved at position 3 (aliphatic residue) and position 10 (aliphatic residue), the presence of a proline at position 1 and a phenyalanine instead of an isoleucine at position 8. These discrepancies might be due to number of sequences compared and the kind of alignment generated. Ptitsyn used 164 sequences whereas here only 19 sequences were used. Although comparatively very few sequences were used, the result seems to be almost the same. This is a promising result opening opportunities in extending the procedure to other superfamilies. However, an attempt on P-loops failed primarily due to the fact that the superfamily is much more diverged and only very few sequences form distinct clusters.

B.4 Structural homologs

I did an all-against-all structural comparison of the domains using PrISM. Then I used the alignment from PrISM as input to another program called MSARMS (Hubbard, 1994) that measures the distance in Angstrom between the matched residues in the superposition. These RMSD values from PrISM and MSARMS programs were used for this study.

B.5 Clustering

With these homologs and their relationship (given as *E-value* for sequences and *RMSD* for structures), I represented proteins as clusters in two-dimensional space. This was done using the procedure given in Figure 75 using sequence/structure distance matrices (or similarity matrices).

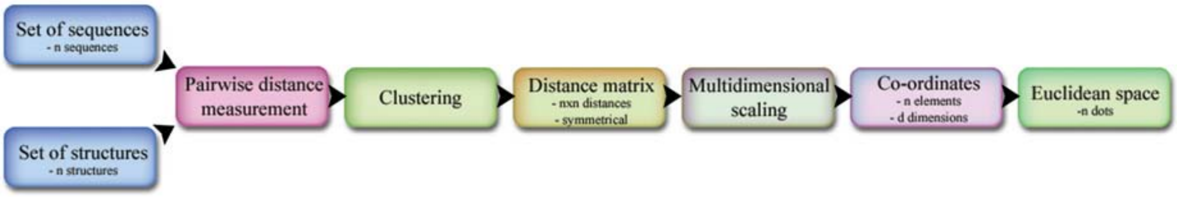


Figure 75. Flow chart describing steps used in clustering and visualisation of data.

I did initial clustering based on the sequence based distance matrix using single and complete linkage methods with a threshold E-value of 0.001 and 0.05 respectively. Then I merged the resulting sets of clusters based on the RMSD values using the Unweighted Pair Group Method using Arithmetic average approach. A threshold value of 4.00Å was used for the P-loops superfamily and a threshold of 2.00Å was used for the cytochrome c superfamily. I also applied the complete linkage approach to merge the initial set of clusters using a threshold value of 6.00Å for both superfamilies.

To find co-ordinates of the data set in 2D space, I used Principal Co-ordinate Analysis (PCoA). For a problem of *N* objects, there could be *N*(N-1)* distances and displayed in *(N-1)* dimensional space. This *(N-1)* dimensional space was reduced to 2D/3D space and plotted.

A manual plotting of the data gave a cluster map for both cytochrome c (Figure 76) and P-loops superfamilies (Figure 77). Figure 78 shows the demarcation of clusters into family and protein levels based on the SCOP classification for cytochrome c. Similarly, Figure 79 shows the demarcation of family levels in P-loops. The protein levels were not marked in P-loops to avoid the complexity in the figure.

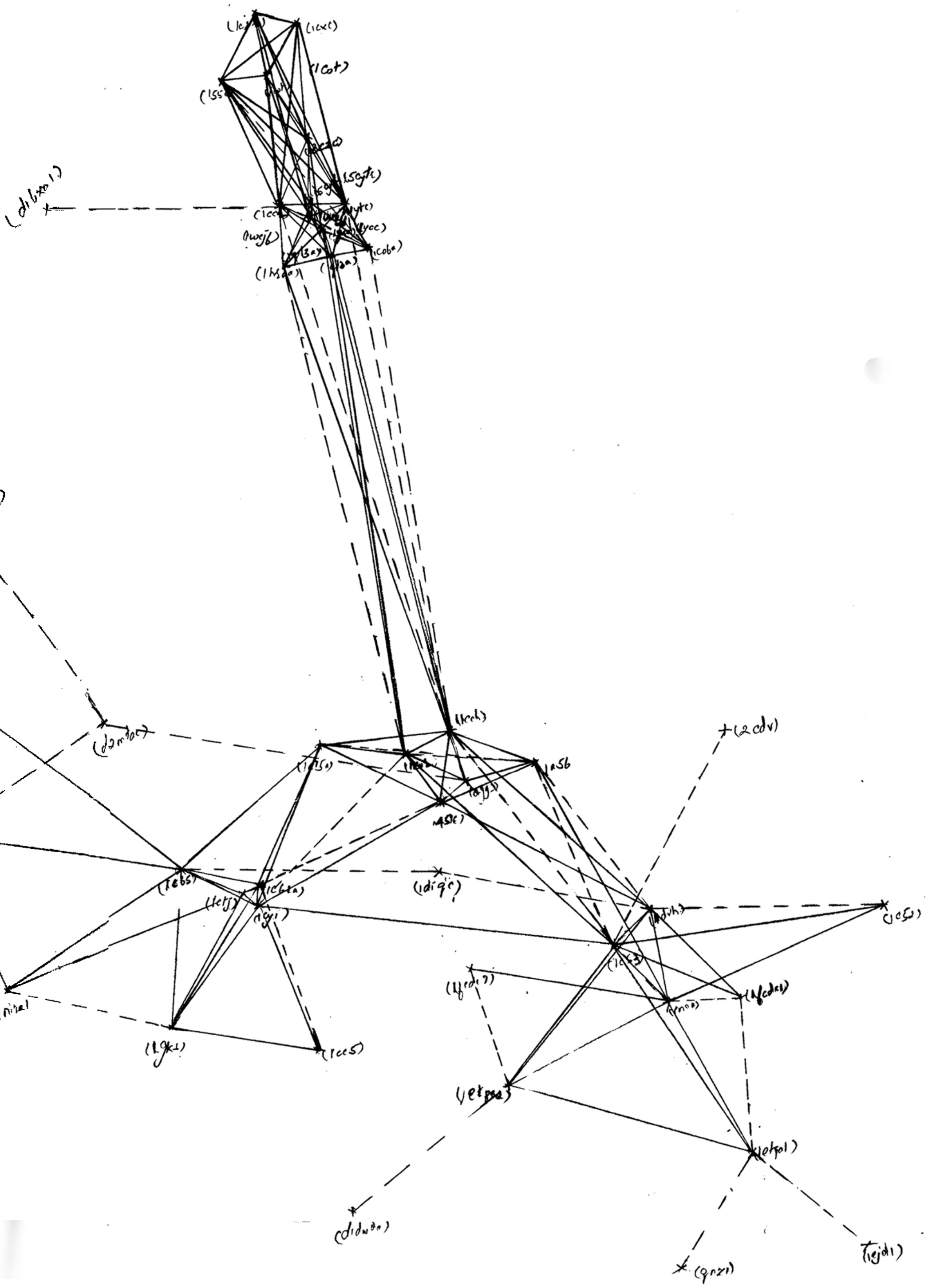


Figure 76. Cluster map of cytochrome c superfamily

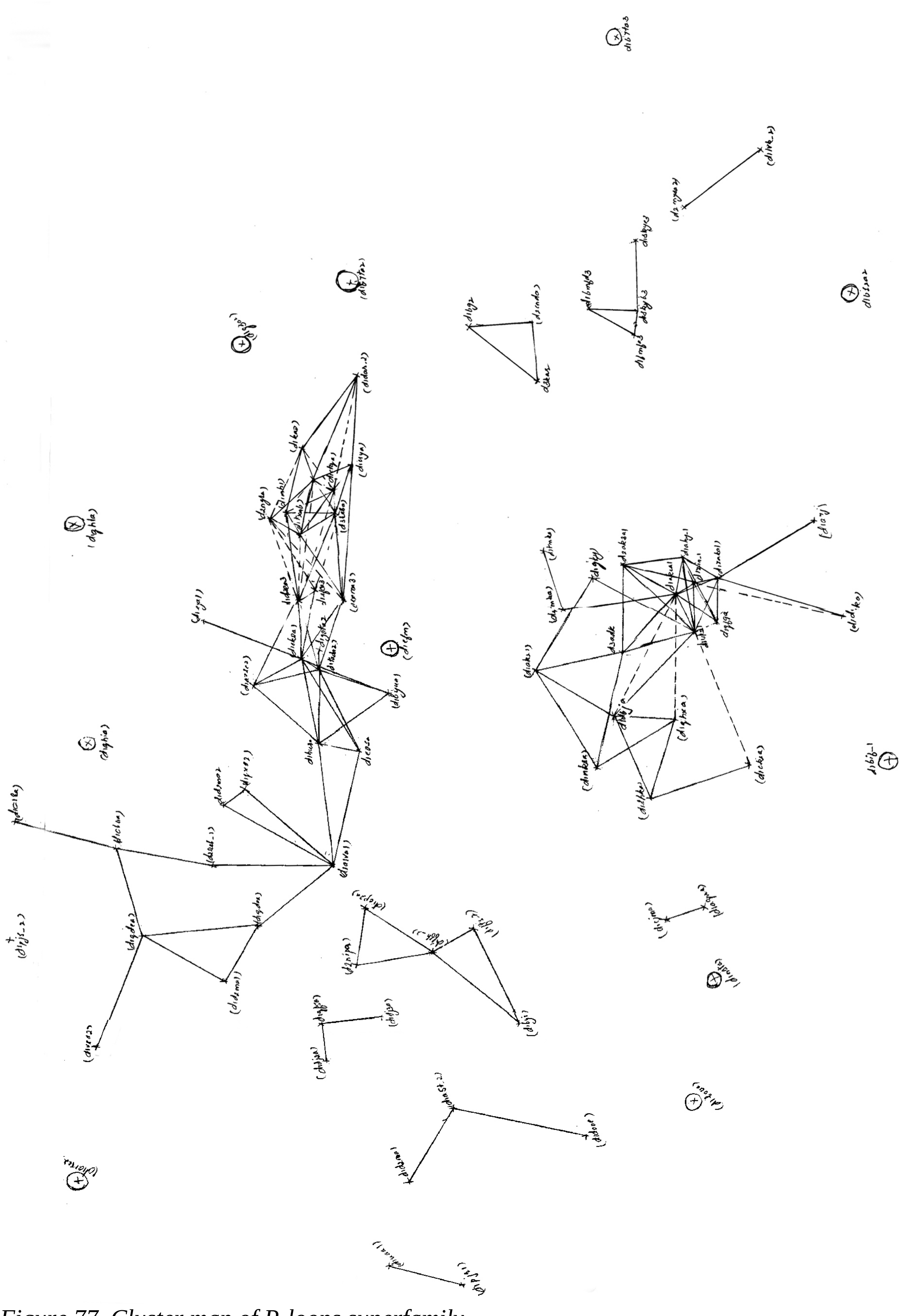


Figure 77. Cluster map of P-loops superfamily

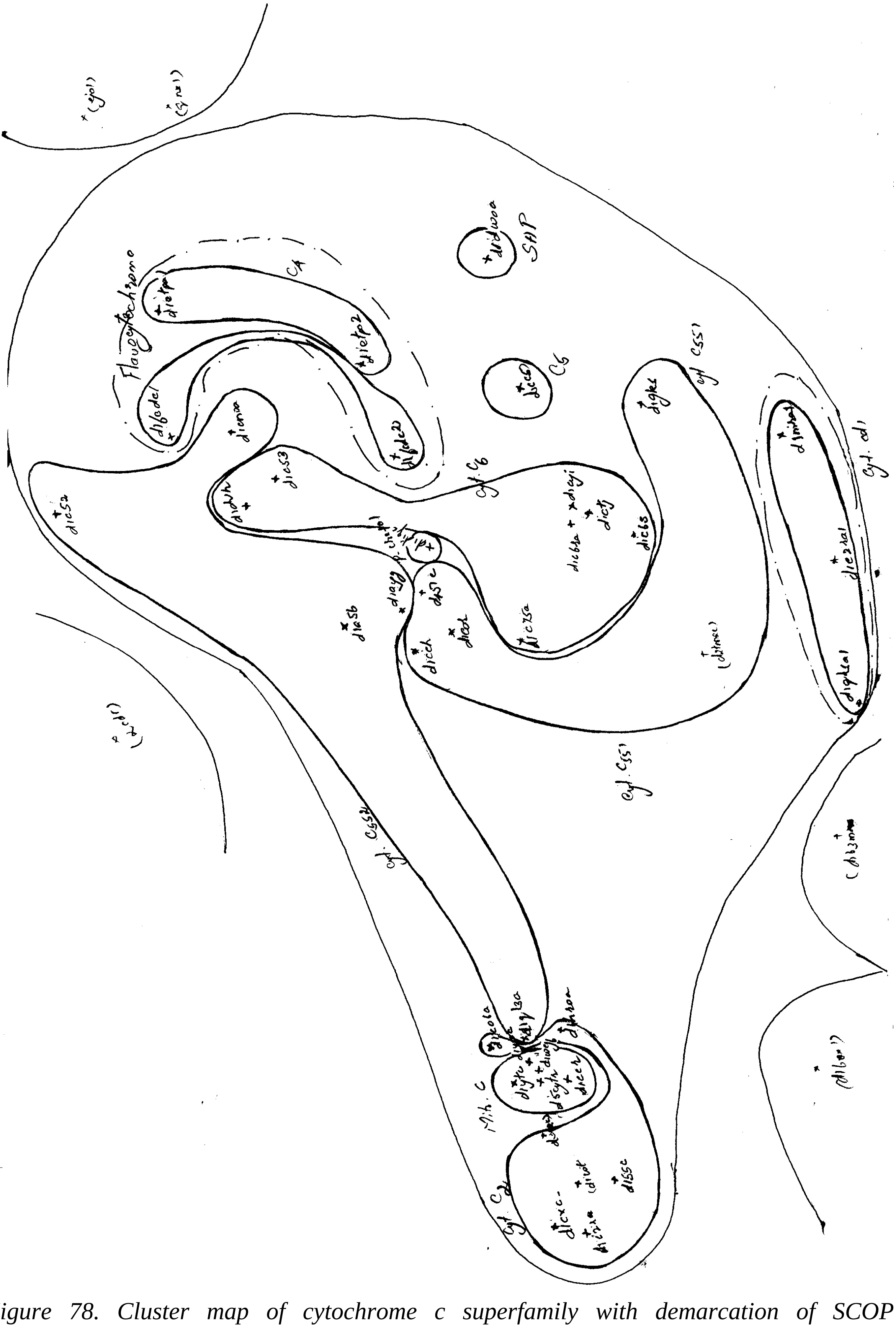


Figure 78. Cluster map of cytochrome c superfamily with demarcation of SCOP superfamily, family and protein levels

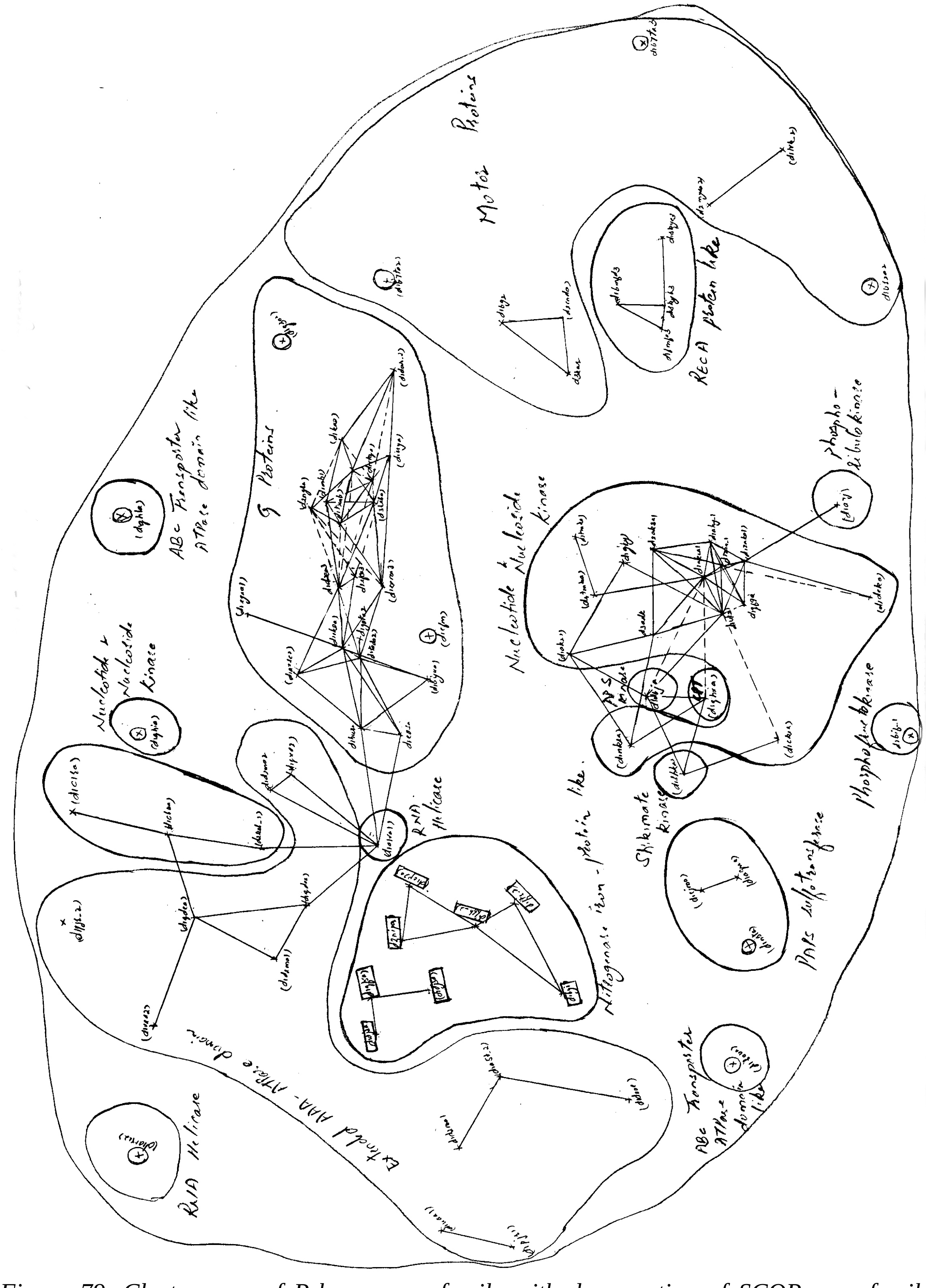

Motor Proteins
RecA protein like
G Proteins
ABC Transporter like ATPase domain
Nucleotide & Nucleoside Kinase
Nucleotide & Nucleoside Kinase
Phospho-ribulokinase
RNA helicase
APS kinase
Shikimate Kinase
Phosphofructokinase
Nitrogenase iron-protein like
PAPS sulfotransferase
ABC Transporter ATPase domain like
Extended AAA-ATPase domain
RNA Helicase

Figure 79. Cluster map of P-loops superfamily with demarcation of SCOP superfamily, family levels

The maps (Figure 76 and Figure 77) show domain relationships either by solid lines or dashed lines. The solid lines indicate domains having strong relationship between them (E-value < 0.4 and RMSD < 4 Å). Also, the length of the solid line represents real Euclidean distance in the cluster map. The dashed lines show there is a relationship between the connected domains. However, the position of domains in the map is not true. This is due to the non-availability of a relationship between the connected domains and its neighbors. Also, the length of the broken line does not represent real Euclidean space in the map.

The cytochrome maps (Figure 76 and Figure 78) show that two SCOP protein groups, *mitochondrial cytochrome c* and *cytochrome* c_2, were well separated from other protein groups. The domains forming the *cytochrome* c_{552} cluster show that they have diverged more than any other SCOP protein group. Also, it can be seen that most of the domains from the *cytochrome* c_6 and *cytochrome* c_{551} SCOP protein groups form closer clusters while some of them get away from this cluster and act as outliers.

P-loops cluster maps (Figure 77 and Figure 79) show that the domains have diverged more when compared to the cytochrome c domains. The maps show a number of domains represented as singletons or as small groups not connected to each other. As stated earlier, absence of a line between domains means no relationship can be identified among them (with score below the threshold limit), although some of the singletons belong to SCOP family. Only members of two families (*Nucleoside and nucleotide kinase* and *G-proteins*) were found to be grouped together on the map. This may be due to more environmental constraints and less active site requirements on P-loop superfamily or may be due to a convergence phenomena as seen in phosphate binding proteins (Bossemeyer, 1994).

These cluster maps are a useful tool to aid in understanding of the relationship between protein members of a family:

(1) It gives an overall picture of the divergence of a protein superfamily.

(2) It shows the relationships between SCOP families.

(3) The method could be used as an initial automated classification procedure of protein structures. A new protein structure can be used as a query to find its sequence or structure homologs. Then based on the sequence and structural relationship (E-value and

RMSD), the protein can be added in the cluster map. Such a map will give a good idea to which of the superfamily or family the new protein belongs. Then with detailed knowledge, the protein can be allocated in a specific family (manual curation). The clustering approach can be exploited to assign function to an unknown protein (Sternberg, 2001), but it cannot be trusted fully as a similar structure does not always represent the same function.

(4) It gives a clear picture about any particular SCOP family and allows the identification of any outliers in it. In the P-loops cluster map (Figure 79), there are two clusters one with domains *d1d2ja__, d1qf5a__ and d1dj3a_* and another with *d2nipa_, d1cp2a_, d1ffh__, d1byi__ and d1fts__* (boxed). But all of these domains are placed in the same family in SCOP. On discussion with Alexey Murzin (the primary curator of SCOP database), he recalled he considered that it might be better to keep these two clusters in two separate groups, say as, two different sub-families/families. He only kept them together due to limitations in the current SCOP classification system.

Likewise the domain *d1qhia_*, classified in the *Nucleotide and nucleoside kinase* family in SCOP, are positioned separately from the main cluster. The outlier was later cross-checked with structural analysis (Morea, 2001). The analysis also agreed that the domain is distinct from its family members. The probable reason for the isolated cluster of *d1qhia__* is that it is a chimeric protein and does not exist naturally i.e. it does not have sequence or structure homology with other *Nucleotide and nucleoside kinase* proteins even though it retains the same function. It was for this reason and since the domain satisfied minimal the P-loop topology, that Alexey Murzin classified the domain under the same family.

Thus, cluster maps might help us to be aware of outliers in a particular superfamily/family classification before starting any kind of detailed analysis on it.

Because of these advantages of the cluster maps, I automated the clustering process to extend the study later for other families. A comparison between manual and automated clustering procedures shows that the automated method performed equally well with the manual method (Figure 80 and Figure 81). Also, the automated methods provide similar results with another automated clustering procedure based on the MCL algorithm (Enright *et al.*, 2002).

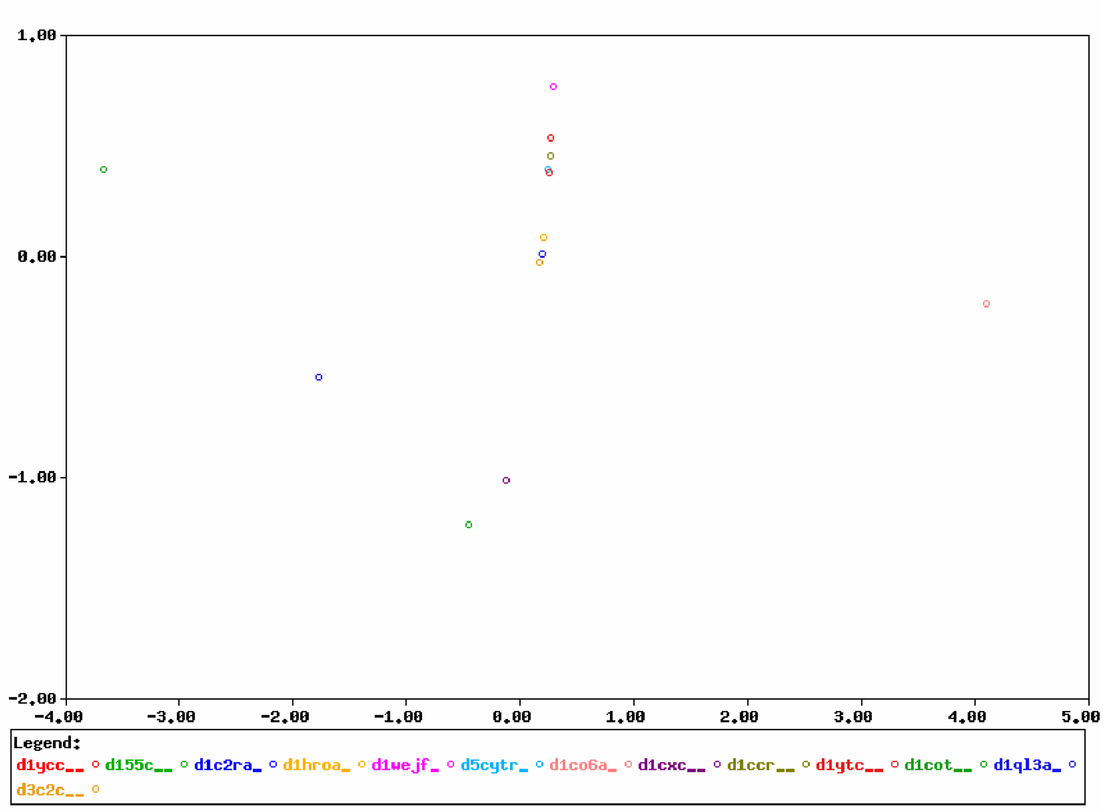


Figure 80. A cluster produced by the automated method for cytochrome c superfamily

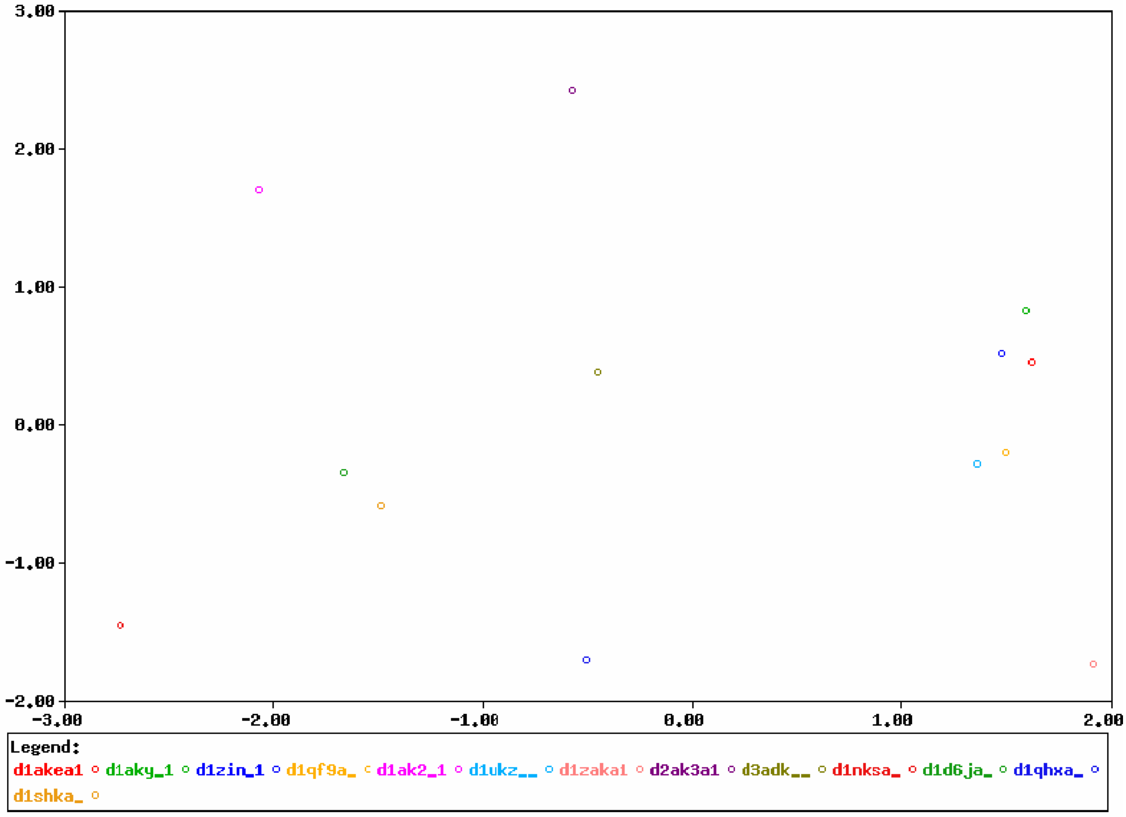


Figure 81. A cluster produced by automated method for P-loops superfamily

In both manual and automated processes, clustering was done using sequence and structural relationships, but it is possible to be done with sequence information alone. However, this will give only the number of clusters that can be formed from the superfamily and members in each cluster. A two dimensional representation of data is difficult with sequence information alone due to the fact that the data needs to undergo significant normalization procedures before it can be used to find co-ordinates.

B.6 Orthology and paralogy

The sequence and structural information, used above to generate cluster maps, can also form the basis for detecting orthologous relationships within protein families in the study of protein evolution. Such a group of ortholog domains was found in P-loops superfamily. The group comprises adenylate kinases from *Escherichia coli, Bacillus stermathermophilus* and *Saccharomyces cerevisiae*. Using species as a time scale, it can be said that adenylate kinase of *Escherichia coli* and *Bacillus stermathermophilus* appeared earlier than yeast protein. However, it does not mean that yeast protein evolved from *Escherichia coli* or *Bacillus* and it would be extremely difficult in assessing the proper time scale for these proteins based on sequence and structure information alone.

All the three adenylate kinases clustered close to each other on the map. So, from tightly clustering domains, it can be presumed that they are possibly to be orthologous to each other.

The TOPS (Westhead *et al.*, 1999) diagrams of these three proteins (Figure 82) shows that *Escherichia coli* and yeast adenylate kinases are identical whereas in *Bacillus*, there is an extra β strand and its orientation is reversed. Interestingly, this part of the protein is not under SCOP domain definition, which means that there is no functional or structural role for this part of the protein. Since this part does not have structural or functional constraints, it is more likely to be subject to mutations and may be influenced by environmental factors of *Bacillus* compared with yeast or *Escherichia coli*. From this, I conclude that the evolution of adenylate kinase would have more likely started from a common ancestor and given rise to *Escherichia coli* and or *Bacillus* and later to yeast protein. Later, *Bacillus* adenylate kinase would have acquired some changes in its protein.

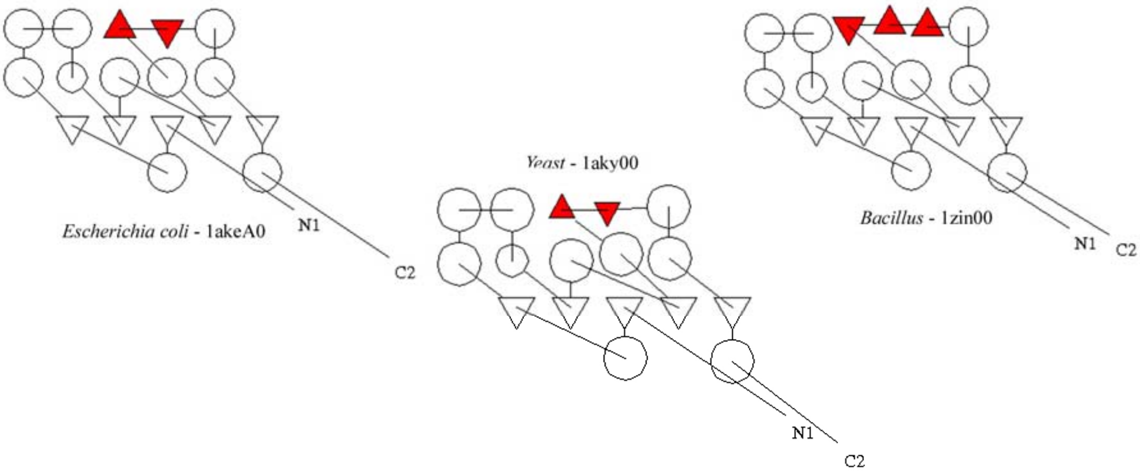


Figure 82. Topology diagram for adenylate kinase

Likewise, from the cytochrome map, two SCOP protein groups form distinct clusters from the rest of the cytochrome members. The overall topology of the cytochrome superfamily members were analyzed using TOPS (Figure 83). Generally, cytochrome c fold has 5 helices. However, some members of *cytochrome* c_{551} group have 6 helices and *cytochrome* c_2 group has 5 helices and 2 β strands except *d3c2c__*, which has only 5 helices. The topology of *cytochrome* c_{552} group (5 helices) remains the same, although its sequence has diverged greatly. However, the domains of this group (*cytochrome* c_{552}) forms close cluster with domains of different cytochrome c protein groups than among itself. It might be one of the typical cases, where orthology/homology cannot be resolved based on sequence identity because an extensive sequence divergence has occurred. However, it can also be argued that *cytochrome* c_{552} proteins were actually formed from convergence of different cytochrome c proteins. But this is highly unlikely to occur given the clear picture of overall divergence of cytochrome proteins and absence of any convergence reports in the cytochrome c fold.

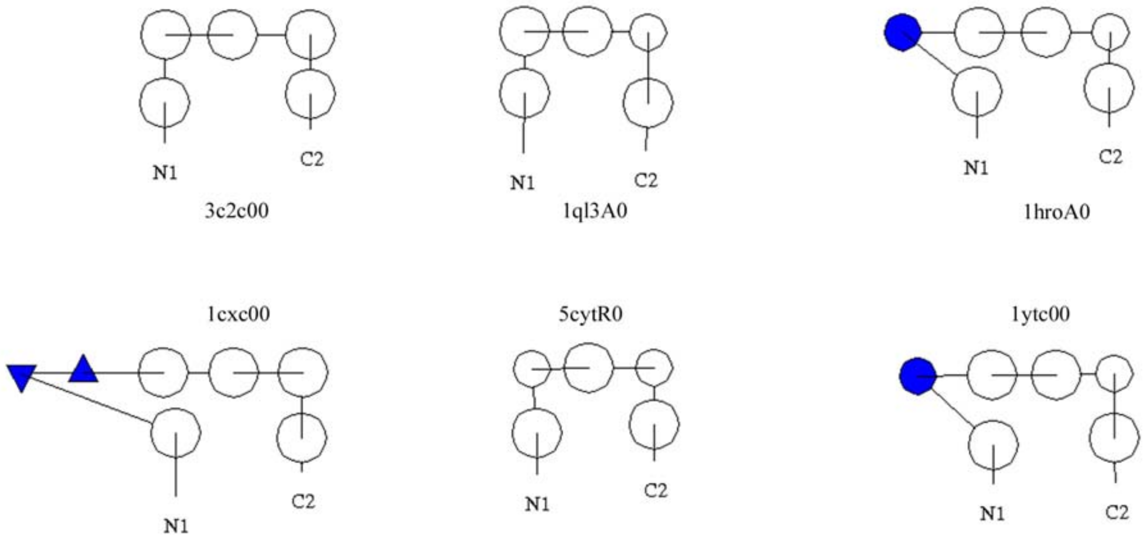


Figure 83. Topology diagram for cytochrome c proteins

Mitochondrial cytochrome c was seen later in the time-scale when compared to bacterial cytochrome c. Given the endosymbiotic hypothesis, it is likely that any bacterial cytochrome c would have given rise to *mitochondrial cytochrome*. Here, it can be seen that *cytochrome c_2* clustered closely with *mitochondrial cytochrome* (Figure 78). So it is likely that *cytochrome c_2* would have been the ancestral protein for *mitochondrial cytochrome*. This was confirmed with expertise knowledge of Alexey Murzin. The topology study of these two SCOP protein groups also confirmed this. The general topology of *cytochrome c_2* and *mitochondrial cytochrome* are 5 helices + 2 β strands and 5 or 6 helices respectively. However, some of the domains of *cytochrome c_2* (e.g., *d3c2c__*), clustering near to *mitochondrial cytochrome* lack the two β strands, confirming that the earlier forms of *cytochrome c_2* with β strands, later lost the β strands and have given rise to *mitochondrial cytochrome*.

Thus, cluster maps made with sequence and structural homology is useful in understanding the ancestry of proteins.

B.7 Conclusions

Protein evolution, driven by structural and functional constraints, may leave a trail of homologs. Homologs are identified using sequence comparison methods like BLAST, FASTA, psi-BLAST and ISS. A comparison of ISS with psi-BLAST was made in two protein superfamilies: cytochrome c and P-loops. The result showed that psi-BLAST detected all the remote homologs identified by ISS in P-loops and only half in cytochrome c superfamily. Although, I cannot generalize using these limited results, it can be said that ISS performs better in some cases than psi-BLAST. The advantage ISS has in some cases might be due to the match score it gains by producing longer alignments around conserved regions of the protein. Intermediate search conducted using structural information revealed that more remote homologs that could not be identified with sequence information alone. So structures might be useful in intermediate search when sequence information is inadequate in detection. From the progressive alignments generated using most of the domains in four SCOP protein groups (*mitochondrial cytochrome, cytochrome* c_2, *cytochrome* c_{551} and *cytochrome* c_6), an overall consensus was generated. The highly conserved residues found in the overall consensus are in tandem with the key structural and functional residues needed for the cytochrome c fold (Ptitsyn, 1998). Thus ISS alignments might be useful in understanding highly conserved residues in a protein fold.

Along with sequence information, I used structural comparisons by PrISM to produce a manual cluster map. The cluster map showed a useful representation of the general evolutionary relationships within P-loops and cytochromes. These might be helpful in depicting the relationship between SCOP families, assigning hierarchies to a new protein structure in the existing structural classification and understanding the likely ancestor of a protein. For example, in cytochrome c superfamily, it was shown that the *cytochrome* c_2 protein is likely to be an ancestor for *mitochondrial cytochrome*. The manual process has been automated and can now be used as a tool in exploring evolutionary relationships of any protein family.

APPENDIX C

C.1 Eponine transcription termination parameters

The parameters used to create Eponine transcription termination model –

```
<? xml version="1.0" ?>

<app xmlns=http://www.sanger.ac.uk/Users/td2/specs/epoapps/0/2
jclass="eponine.TrainingCore">

        <bean name="dataSource" jclass="eponine.datasource.XMLDataSource">
                <string name="fileName" value="Datasets/trainingdata.xml " />
        </bean>

        <bean name="basisSource" jclass="eponine.model.MultiplexedBasisSource">
                <int name="reweightFrequency" value="15" />
                <double name="reweightPseudocounts" value="10.0" />

                <child jclass="eponine.model.NewBasisSource">
                     <boolean name="maximize" value="false" />
                     <double name="stringency" value="0.55" />
                     <double name="stringencyVariance" value="0.03" />
                     <int name="minLength" value="4" />
                     <int name="maxLength" value="8" />
                     <double name="minDistWidth" value="2.5" />
                     <double name="maxDistWidth" value="200.0" />
                     <boolean name="reversible" value="false" />
                     <string name="name" value="nbs1_narrow" />
                     <int name="minPos" value="-190" />
                     <int name="maxPos" value="1990" />
                </child>

                <child jclass="eponine.model.SampleWMBasisSource">
                     <double name="nullModelWeighting" value="7.0" />
                     <double name="nullModelPerMarginalColumn" value="1.0" />
                     <int name="sampleCounts" value="203" />
                     <double name="nullModelWeightingN" value="9.0" />
                     <int name="sampleCountsN" value="120" />
                     <string name="name" value="samplewm2" />
                </child>

                <child jclass="eponine.model.DropColumnBasisSource" />

                <child jclass="eponine.model.DistributionBasisSource">
                     <double name="distChangeWidth" value="3.0" />
                     <double name="distChangeGamma" value="3.0" />
                     <double name="distChangeScale" value="25.0" />
                     <double name="distChangeBias" value="0.06" />
                     <!-- double name="shapeChangeProbability" value="0.05" / -->
                     <double name="flipEnvelopeProbability" value="0.00" />
                     <string name="name" value="distwidth" />
                </child>

                <child jclass="eponine.model.PositionBasisSource">
                     <double name="shiftWidth" value="4" />
                </child>

                <child jclass="eponine.model.CrossWMBasisSource" />

                <child jclass="eponine.model.AppendColumnBasisSource" />

                <child jclass="eponine.model.FlipMaxBasisSource" />
        </bean>

        <bean name="trainer" jclass="stats.glm.VRVMTrainer">
                <int name="numThreads" value="4" />
                <int name="maxCycles" value="11000" />
                <!--int name="cleaningCycles" value="0" /-->
                <int name="maxWorkingSet" value="28" />
                <int name="minWorkingSet" value="25" />
                <int name="initialWorkingSet" value="50" />
                <double name="initialAlpha" value="1.0" />
                <boolean name="unityHack" value="true" />
                <double name="unityHackThreshold" value="1.0" />
                <boolean name="resetAlphaHack" value="true" />
                <boolean name="insertUnity" value="true" />
        </bean>

        <bean name="retrainer" jclass="stats.glm.VRVMTrainer">
                <int name="maxCycles" value="100" />
                <!--int name="cleaningCycles" value="0" /-->
                <double name="initialAlpha" value="1.0" />
                <boolean name="unityHack" value="true" />
                <double name="unityHackThreshold" value="1.0" />
        </bean>

        <string name="fileName" value="Models/terminationmodel.xml" />
        <int name="checkpointFrequency" value="500" />
</app>
```

C.2 GAZE gene structure models

The configuration file explaining the gene model with translation features for predicting genes using GenePred –

```
<? xml version="1.0" encoding="US-ASCII" ?>

    <gaze>
        <declarations>
            <feature id="tss" st_off="0" en_off="1" />
            <feature id="tis" st_off="0" en_off="3"/>
            <feature id="5ss" st_off="1" en_off="1" />
            <feature id="3ss" st_off="1" en_off="1" />
            <feature id="tts" st_off="3" en_off="0"/>
            <feature id="polyA" st_off="1" en_off="1"/>

            <feature id="tss_rev" st_off="1" en_off="0" />
            <feature id="tis_rev" st_off="3" en_off="0" />
            <feature id="5ss_rev" st_off="1" en_off="1" />
            <feature id="3ss_rev" st_off="1" en_off="1" />
            <feature id="tts_rev" st_off="0" en_off="3" />
            <feature id="polyA_rev" st_off="1" en_off="1"/>

            <!--lengthfunction id="intron_pen" />
            <lengthfunction id="intergene_pen" />
            <lengthfunction id="inital_exon_pen" />
            <lengthfunction id="internal_exon_pen" />
            <lengthfunction id="terminal_exon_pen" />
            <lengthfunction id="single_exon_gene_pen" /-->
        </declarations>

        <gff2gaze>
            <!-- Features -->
            <gfffeat feature="TSS" strand="+" source="Eponine">
                <feat id="tss"/>
            </gfffeat>

            <gfffeat feature="TSS" strand="-" source="Eponine">
                <feat id="tss_rev"/>
            </gfffeat>

            <gfffeat feature="TIS" strand="+" source="Eponine">
                <feat id="tis"/>
            </gfffeat>

            <gfffeat feature="TIS" strand="-" source="Eponine">
                <feat id="tis_rev"/>
            </gfffeat>

            <gfffeat feature="5SS" strand="+" source="Eponine">
                <feat id="5ss"/>
            </gfffeat>

            <gfffeat feature="5SS" strand="-" source="Eponine">
                <feat id="5ss_rev"/>
            </gfffeat>

            <gfffeat feature="3SS" strand="+" source="Eponine">
                <feat id="3ss"/>
            </gfffeat>

            <gfffeat feature="3SS" strand="-" source="Eponine">
                <feat id="3ss_rev"/>
            </gfffeat>

            <gfffeat feature="TTS" strand="+" source="Eponine">
                <feat id="tts"/>
            </gfffeat>
            <gfffeat feature="TTS" strand="-" source="Eponine">
                <feat id="tts_rev"/>
            </gfffeat>

            <gfffeat feature="POLYA" strand="+" source="Eponine">
                <feat id="polyA"/>
            </gfffeat>

            <gfffeat feature="POLYA" strand="-" source="Eponine">
                <feat id="polyA_rev"/>
            </gfffeat>
        </gff2gaze>

        <dna2gaze>
            <!--dnafeat pattern="tataaa">
                <feat id="tss" />
            </dnafeat>

            <dnafeat pattern="atg" score="0.001">
                <feat id="tis" />
            </dnafeat>

            <dnafeat pattern="taa" score="0.001">
                <feat id="tts" />
            </dnafeat>

            <dnafeat pattern="tag" score="0.001">
                <feat id="tts" />
            </dnafeat>

            <dnafeat pattern="tga" score="0.001">
                <feat id="tts" />
            </dnafeat>

            <dnafeat pattern="aataaa" score="0.001">
                <feat id="polyA" />
            </dnafeat>

            <dnafeat pattern="tttata">
                <feat id="tss_rev" />
            </dnafeat>

            <dnafeat pattern="cat" score="0.001">
                <feat id="tis_rev" />
            </dnafeat>

            <dnafeat pattern="tta" score="0.001">
                <feat id="tts_rev" />
            </dnafeat>

            <dnafeat pattern="cta" score="0.001">
                <feat id="tts_rev" />
            </dnafeat>
            <dnafeat pattern="tca" score="0.001">
                <feat id="tts_rev" />
            </dnafeat>

            <dnafeat pattern="tttatt" score="0.001">
                <feat id="polyA_rev" />
            </dnafeat-->

            <!--takedna id="5ss_1" st_off="0" en_off="1"/>
            <takedna id="3ss_1" st_off="1" en_off="-1"/>
            <takedna id="5ss_2" st_off="-1" en_off="1"/>
            <takedna id="3ss_2" st_off="1" en_off="0"/>
            <takedna id="5ss_1_rev" st_off="1" en_off="0"/>
            <takedna id="3ss_1_rev" st_off="-1" en_off="1"/>
            <takedna id="5ss_2_rev" st_off="1" en_off="-1"/>
            <takedna id="3ss_2_rev" st_off="0" en_off="1"/-->
        </dna2gaze>

        <model>
            <target id="END">
                <source id="BEGIN" out_feat="No_genes"/>
                <source id="polyA" out_feat="GEN_DNA" />
                <source id="tss_rev" out_feat="GEN_DNA"/>
            </target>

            <!--Forward strand gene-->

            <target id="tss">
                <source id="BEGIN" out_feat="GEN_DNA"/>
                <source id="polyA" mindis="1" out_feat="intergenic"/>
                <source id="tss_rev" mindis="1" out_feat="intergenic"/>
            </target>

            <target id="tis">
                <source id="tss" mindis="1" out_feat="5UTR" out_str="+"/>
            </target>

            <target id="5ss">
                <!--killfeat id="tts"/-->
                <source id="tis" out_feat="inital_exon" mindis="3" maxdis= "10000" out_str="+" />
                <source id="3ss" out_feat="internal_exon"  mindis="6" maxdis= "10000" out_str="+"
            />
            </target>

            <target id="3ss">
                <source id="5ss" out_feat="intron"  mindis="6" out_str="+"/>
            </target>

            <target id="tts">
                <!--killfeat id="tts" /-->
                <!--source id="tis" out_feat="single_exon_gene" mindis="60"out_str="+"/-->
                <source id="3ss" out_feat="terminal_exon"  mindis="3" out_str="+"/>
            </target>

            <target id="polyA">
```

```
            <source id="tts" out_feat="3UTR" mindis="1" out_str="+"/>
        </target>

        <!--Reverse strand gene-->

        <target id="polyA_rev">
            <source id="BEGIN" out_feat="GEN_DNA"/>
            <source id="polyA" out_feat="intergenic" mindis="1"/>
            <source id="tss_rev" out_feat="intergenic" mindis="1"/>
        </target>

        <target id="tts_rev">
            <source id="polyA_rev" out_feat="3UTR" mindis="1" out_str="-"/>
        </target>

        <target id="3ss_rev">
            <!--killfeat id="tts_rev"/-->
            <source id="tts_rev" out_feat="terminal_exon" mindis="3" maxdis= "10000"
        out_str="-"/>
            <source id="5ss_rev" out_feat="internal_exon" mindis="6" maxdis= "10000"
        out_str="-"/>
        </target>

        <target id="5ss_rev">
            <source id="3ss_rev" out_feat="intron" mindis="6" out_str="-"/>
        </target>

        <target id="tis_rev">
            <!--killfeat id="tts_rev" phase="0"/-->
            <!--source id="tts_rev" out_feat="single_exon_gene" mindis="60" out_str="-"/-->
            <source id="5ss_rev" out_feat="initial_exon" mindis="3" out_str="-"/>
        </target>

        <target id="tss_rev">
            <source id="tis_rev" out_feat="5UTR" mindis="1" out_str="-"/>
        </target>
    </model>

    <lengthfunctions>
        <!-- lengthfunc id="intron_pen" file="./tables/intron_penalty"/>
        <lengthfunc id="initial_exon_pen" file="./tables/exon_penalty.initial"/>
        <lengthfunc id="terminal_exon_pen" file="./tables/exon_penalty.terminal"/>
        <lengthfunc id="internal_exon_pen" file="./tables/exon_penalty.internal"/-->

        <!--lengthfunc id="single_exon_gene_pen">
            <point x="500" y ="0.001"/>
            <point x="20000" y="0.2"/>
        </lengthfunc-->

        <!--lengthfunc id="intergene_pen">
            <point x="200000" y ="0.01"/>
            <point x="200001" y="0.01"/>
        </lengthfunc -->
    </lengthfunctions>
</gaze>
```

The configuration file explaining the gene model without translation features for predicting genes using GenePred –

```
<?xml version="1.0" encoding="US-ASCII"?>

    <gaze>
        <declarations>
            <feature id="tss" st_off="0" en_off="1" />
            <!--feature id="tis" st_off="0" en_off="3"/-->
            <feature id="5ss" st_off="1" en_off="1" />
            <feature id="3ss" st_off="1" en_off="1" />
            <!--feature id="tts" st_off="3" en_off="0"/-->
            <feature id="polyA" st_off="1" en_off="1"/>

            <feature id="tss_rev" st_off="1" en_off="0" />
            <!--feature id="tis_rev" st_off="3" en_off="0" /-->
            <feature id="5ss_rev" st_off="1" en_off="1" />
            <feature id="3ss_rev" st_off="1" en_off="1" />
            <!--feature id="tts_rev" st_off="0" en_off="3" /-->
            <feature id="polyA_rev" st_off="1" en_off="1"/>

            <!--lengthfunction id="intron_pen" />
            <lengthfunction id="intergene_pen" />
            <lengthfunction id="inital_exon_pen" />
            <lengthfunction id="internal_exon_pen" />
            <lengthfunction id="terminal_exon_pen" />
            <lengthfunction id="single_exon_gene_pen" /-->
        </declarations>

        <gff2gaze>
            <!-- Features -->
            <gfffeat feature="TSS" strand="+" source="Eponine">
                <feat id="tss"/>
            </gfffeat>

            <gfffeat feature="TSS" strand="-" source="Eponine">
                <feat id="tss_rev"/>
            </gfffeat>

            <!--gfffeat feature="TIS" strand="+" source="Eponine">
                <feat id="tis"/>
            </gfffeat>

            <gfffeat feature="TIS" strand="-" source="Eponine">
                <feat id="tis_rev"/-->
            </gfffeat>

            <gfffeat feature="5SS" strand="+" source="Eponine">
                <feat id="5ss"/>
            </gfffeat>

            <gfffeat feature="5SS" strand="-" source="Eponine">
                <feat id="5ss_rev"/>
            </gfffeat>

            <gfffeat feature="3SS" strand="+" source="Eponine">
                <feat id="3ss"/>
            </gfffeat>

            <gfffeat feature="3SS" strand="-" source="Eponine">
                <feat id="3ss_rev"/>
            </gfffeat>

            <!--gfffeat feature="TTS" strand="+" source="Eponine">
                <feat id="tts"/>
            </gfffeat>

            <gfffeat feature="TTS" strand="-" source="Eponine">
                <feat id="tts_rev"/-->
            </gfffeat>

            <gfffeat feature="POLYA" strand="+" source="Eponine">
                <feat id="polyA"/>
            </gfffeat>

            <gfffeat feature="POLYA" strand="-" source="Eponine">
                <feat id="polyA_rev"/>
            </gfffeat>
        </gff2gaze>

        <dna2gaze>
            <!--dnafeat pattern="tataaa">
                <feat id="tss" />
            </dnafeat>

            <dnafeat pattern="atg" score="0.001">
                <feat id="tis" />
            </dnafeat>

            <dnafeat pattern="taa" score="0.001">
                <feat id="tts" />
            </dnafeat>

            <dnafeat pattern="tag" score="0.001">
                <feat id="tts" />
            </dnafeat>

            <dnafeat pattern="tga" score="0.001">
                <feat id="tts" />
            </dnafeat>

            <dnafeat pattern="aataaa" score="0.001">
                <feat id="polyA" />
            </dnafeat>

            <dnafeat pattern="tttata">
                <feat id="tss_rev" />
            </dnafeat>
            <dnafeat pattern="cat" score="0.001">
                <feat id="tis_rev" />
            </dnafeat>

            <dnafeat pattern="tta" score="0.001">
                <feat id="tts_rev" />
            </dnafeat>

            <dnafeat pattern="cta" score="0.001">
                <feat id="tts_rev" />
            </dnafeat>

            <dnafeat pattern="tca" score="0.001">
                <feat id="tts_rev" />
            </dnafeat>

            <dnafeat pattern="tttatt" score="0.001">
                <feat id="polyA_rev" />
            </dnafeat-->

            <!--takedna id="5ss_1" st_off="0" en_off="1"/>
            <takedna id="3ss_1" st_off="1" en_off="-1"/>
            <takedna id="5ss_2" st_off="-1" en_off="1"/>
            <takedna id="3ss_2" st_off="1" en_off="0"/>
            <takedna id="5ss_1_rev" st_off="1" en_off="0"/>
            <takedna id="3ss_1_rev" st_off="-1" en_off="1"/>
            <takedna id="5ss_2_rev" st_off="1" en_off="-1"/>
            <takedna id="3ss_2_rev" st_off="0" en_off="1"/-->
        </dna2gaze>

        <model>
            <target id="END">
                <source id="BEGIN" out_feat="No_genes"/>
                <source id="polyA" out_feat="GEN_DNA" />
                <source id="tss_rev" out_feat="GEN_DNA"/>
            </target>

            <!--Forward strand gene-->
            <target id="tss">
                <source id="BEGIN" out_feat="GEN_DNA"/>
                <source id="polyA" mindis="1" out_feat="intergenic"/>
                <source id="tss_rev" mindis="1" out_feat="intergenic"/>
            </target>

            <!--target id="tis">
                <source id="tss" mindis="1" out_feat="5UTR" out_str="+"/>
            </target-->

            <target id="5ss">
                <!--killfeat id="tts"/-->
                <!--source id="tis" out_feat="inital_exon" mindis="3" maxdis= "10000" out_str="+"
            /-->
                <source id="tss" mindis="1" out_feat="initial_exon" out_str="+"/>
                <source id="3ss" out_feat="internal_exon"  mindis="6" maxdis= "10000"
            out_str="+" />
            </target>

            <target id="3ss">
                <source id="5ss" out_feat="intron"  mindis="6" out_str="+"/>
            </target>

            <!--target id="tts"-->
                <!--killfeat id="tts" /-->
                <!--source id="tis" out_feat="single_exon_gene" mindis="60" out_str="+"/-->
                <!--source id="3ss" out_feat="terminal_exon"  mindis="3" out_str="+"/>
            </target-->

            <target id="polyA">
                <!--source id="tts" out_feat="3UTR" mindis="1" out_str="+"/-->
                <source id="3ss" out_feat="terminal_exon"  mindis="3" out_str="+"/>
            </target>

            <!--Reverse strand gene-->

            <target id="polyA_rev">
                <source id="BEGIN" out_feat="GEN_DNA"/>
                <source id="polyA" out_feat="intergenic" mindis="1"/>
                <source id="tss_rev" out_feat="intergenic" mindis="1"/>
            </target>

            <!--target id="tts_rev">
                <source id="polyA_rev" out_feat="3UTR" mindis="1" out_str="-"/>
            </target-->

            <target id="3ss_rev">
                <!--killfeat id="tts_rev" phase="0"/-->
                <source id="polyA_rev" out_feat="terminal_exon" mindis="3" maxdis= "10000"
            out_str="-"/>
                <source id="5ss_rev" out_feat="internal_exon" mindis="6" maxdis= "10000"
            out_str="-"/>
            </target>

            <target id="5ss_rev">
                <source id="3ss_rev" out_feat="intron" mindis="6" out_str="-"/>
            </target>

            <!--target id="tis_rev"-->
                <!--killfeat id="tts_rev" phase="0"/-->
                <!--source id="tts_rev" out_feat="single_exon_gene" mindis="60" out_str="-"/-->
                <!--source id="5ss_rev" out_feat="initial_exon" mindis="3" out_str="-"/>
            </target-->

            <target id="tss_rev">
                <source id="5ss_rev" out_feat="initial_exon" mindis="1" out_str="-"/>
            </target>
        </model>
        <lengthfunctions>
            <!-- lengthfunc id="intron_pen" file="./tables/intron_penalty"/>
```

```
      <lengthfunc id="initial_exon_pen" file="./tables/exon_penalty.initial"/>
      <lengthfunc id="terminal_exon_pen" file="./tables/exon_penalty.terminal"/>
      <lengthfunc id="internal_exon_pen" file="./tables/exon_penalty.internal"/-->

      <!--lengthfunc id="single_exon_gene_pen">
         <point x="500" y ="0.001"/>
         <point x="20000" y="0.2"/>
      </lengthfunc-->

      <!--lengthfunc id="intergene_pen">
         <point x="200000" y ="0.01"/>
         <point x="200001" y="0.01"/>
      </lengthfunc -->
   </lengthfunctions>
</gaze>
```