Prediction of Intron-Exon Boundaries And Transcription Start Sites of *Entamoeba histolytica* Genes

A thesis submitted in partial fulfillment of the requirements

for the award of the degree of

Master of Technology

Under the Guidance of

Prof. Alok Bhattacharya

Ravindra Sambhaji Patake

School of Information Technology

Jawaharlal Nehru University

New Delhi – 110067

2009

CENTRE FOR COMPUTATIONAL BIOLOGY AND BIOINFORMATICS
SCHOOL OF INFORMATION TECHNOLOGY
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI – 110067.

CERTIFICATE

Certified that this M.Tech. dissertation titled "**Prediction of intron-exon boundaries and transcription start sites in Entamoeba histolytica genes**" has been carried out by **Mr. Ravindra Sambhaji Patake** under my guidance and supervision. This work is original and has not been submitted elsewhere for award of any degree or diploma.

Prof. Indira Ghosh
Dean
School of Information Technology
Jawaharlal Nehru University

Prof. Alok Bhattacharya
Supervisor
School of Information Technology
Jawaharlal Nehru University

Dedicated To

My family and to my fiancee Dipali.

Acknowledgment

Now the time has come to give thanks to all of them without whom, this work could not be possible for me to do.

I will always remain indebted to my guide, my mentor, Prof. Alok Bhattacharya for his suggestions, advice and encouragement.
I am very thankful to Dr. A .Krishnamachari for his kind suggestions. I would like to express hearty gratitude to the Dean of SIT, Prof. Indira Ghosh, all faculty members and teachers for their constant encouragement. I am also thankful to Ms. Candida Vaz, Mr. Sarbashish Das and other senior SIT fellows for their kind help.
I am very thankful to Mr. Siteshu (SLS), for providing me data for my work.

I have no words to thanks all my friends-my batch mates Varun, Anmol, Ravi[2], Ashutosh, Sandip, Arvind, Chetna, Shirin, Lokesh and Sarwar for their support and encouragement. I would also like to give thanks to Ms. Jyoti (SIT trainee fellow) and Ms. Mridula (SLS) for their kind help. I am also thankful to all non teaching staff of SIT for their kind help.

An last but not least, I would express my gratitude to DBT for financial support.

Ravindra Sambhaji Patake

R.S.Patake

20/7/09

INDEX

Introduction And Review:

Basic biology of protist parasite *Entamoeba histolytica:*

Entamoeba histolytica was named for its ability to destroy "Host Tissues". It is a protozoan parasite and capable of invading intestinal mucosa and causing amebic dysentery. It is now clear that tissue destruction is due to both amebic factors and to inflammatory response of the host. (Huston C. D. (2004), Alfonso Olivos-Garcia (2009)). *E. histolytica* shows two-staged life cycle. When a person ingests infectious cyst via contaminated food or water, "excystation" (breaking of the cyst) occurs in the lumen of the intestine, resulting in formation of motile trophozoites. The majority of infected persons (about 90%) do not show the characteristic symptoms of amebic dysentery, such as sub-acute watery or bloody diarrhea and weight loss, but remaining do show the same. The proposed model of mechanism of invasion and virulence is as follows: *E. histolytica* first attaches to the mucous layer with the help of lectins. It secretes amebic cysteine proteases to degrade mucus layer. As it comes into contact with the luminal epithelial host cells, it secretes amebapore. This protein shares homology with the pore forming proteins granulysin and NK-lysin produced by cytotoxic T cells and natural killer (NK) cells respectively. Then *E. histolytica* triggers contact dependent apoptotic and necrotic killing of host cells. Amebic cysteine proteases activate pre-IL-1beta in neighboring cells, which after converting into IL-1beta activates NF- kappaB resulting secretion of inflammatory cytokines. Inflammatory cytokines recruits neutrophils and other leukocytes. They all contribute to the tissue damage. After tissue damage, deep invasion and lateral spread of *E. histolytica* occur (Huston C. D. (2004)).

Entamoeba dispar, a sibling species of *E. histolytica*, is a non pathogenic species. There are a number of differences between the two species including sequence differences. However, precise cause of lack of pathogenicity is not known. (Alfonso Olivos-Garcia, (2009)).

Genome of *E. histolytica*:

A draft sequence of the genome of *E. histolytica* has been published by an International consortium (Fraser and Hall et al (2005)). The sequence is not finished and it consists of 23,751,783 base-pairs. The number of annotated genes are 9,938. Phylogenetic analysis of *E. histolytica* genome provides evidence for the lateral gene

transfer of bacterial genes into *it's* genome, thus widening spectrum of metabolic pathways. The predicted genes show both prokaryotic and eukaryotic features. *Entamoeba* lacks distinct cell-organelles such as Golgi and rough ER but strikingly shows an extensive vesicular transport system including phagocytosis and pinocytosis.

Introns and Exons of *E. histolytica*:

Introns evolve much more rapidly than exon because of lack of selective pressure to produce protein with useful sequences. The positions of introns are usually conserved when homologous genes are compared between different organisms but the length of the corresponding introns may vary greatly.

Mutations that affect splicing are usually deleterious. The majority are single base substitution at the junctions between introns and exons. These may cause presence of an unspliced intron or splicing occuring at an aberrant site. The most common result is to introduce a termination codon that results in the truncation of the protein sequence. About 15% of the point mutation that causes human diseases are caused by disruption of splicing (Benjamin Lewin, 2004).

Only a fraction of genes in *E. histolytica* are predicted to have introns. Average length of *E. histolytica* introns is about 60 nucleotides, very few but considerable number of introns exceed length up to 150. *E. histolytica* introns are AT rich in nature.

Transcription start and termination sites of *E. histolytica*

The minimal Pol II promoter has a TATA box ~25 base pairs upstream of the initiator (Inr) sequence. The TATA box has the consensus sequence of TATAA. The Inr has pyrimidines (Y) surrounding the CA at the start point. At the start point, there is no extensive homology of sequence but there is tendency for the first base of mRNA to be "A" flanked either side by pyrimidines (the description is also valid for the CAT start sequence of bacterial promoters).

CpG islands mostly surround the promoters of constitutively expressed genes where they are un-methylated. They are also found at the promoters of some tissue regulated genes. Methylation of CpG islands prevents activation of promoter within it. Repression is caused by proteins that bind to methylated CpG duplexes (Benjamin Lewin 2004).

Transcription factor (TF) binding sites are typically 5 to 15 nucleotides long and there consensus is generally described by position weight matrix (PWMS) that assign a weight to each possible nucleotide in each position of the putative binding sites based on the observed occurrence of that nucleotide in the known promoter elements (Graziano Pesole et al. (2003)).

The transcription start site is the position from which RNA polymerase actually start transcribing genes. It is normally present upstream to the "translation start site (generally Met)". The region between transcription start site and translation start site is called an Untranslated Region (5' UTR) of the gene. Identification of correct transcription start site and UTR is inevitably important due to the importance of these in mRNA processing, transport and translation. The size of UTRs generally vary from species to species. On the average amebic UTRs are much smaller than that of mammalian ones based on analysis of only a few sequences. In general only a few UTRs have been experimentally mapped or identified in *E. histolytica.*

E. histolytica have many unusual features at transcription level such as "alpha amanitin" resistant RNA POL II enzyme (Lioutas and Tannich et al (1995)), Divergent TATA Binding Protein, Short 5' and 3' untranslated regions. It has been shown that amebic promoter sequences do not function in a mammalian system; and that the viral promoters (Cytomegalo virus, HIV, SV40) and promoters from other systems (*Dictyostelium*) are not functional in amebic trophozoits (Singh U. and Rogers B., (1998)). Amebic unusual core promoters show non-consensus TATA , Inr regions, and consensus GAAC element. In the *hgl5* gene promoter the GAAC element (AATGAATC) is speculated to determine the site and rate of transcription initiation. The amebic TATA (GTATTTAA(G/C)) and INR (AAAAATTCA) elements appear to be functional in a classical manner despite their sequence divergence from metazoans (Petri W. A. Jr. et al, (2001)). The 5' sequence of the ferredoxin gene (*fdx*) contains the motif TATTCTATT (URE3). This sequence is also present in the 5' sequence of the lectin genes *hgl3* and *hgl5* and in those of alkaly-hydroperoxidase reductase (Petri W. A. Jr. et al, (1998)). Ribosomal genes have unusual feature e. g. transcription of SSU ribosomal gene starts 2447 bp upstream, at an adenosine residue (Zurita M. et. al. (1995)).

Transcription termination site (TTS) is a site where RNA Pol actually stops transcription and mRNA cleavage occurs along with the process of poly adenylation (Poly A). Poly A signals are usually embedded in the DNA sequence generally at the

end of the 3' UTR. Simplest mechanism found in bacteria where occurrence of the"Hairpin Loop" followed by poly U terminates the transcription. From yeast "*Saccharomyces cerevisiae* " to higher mammals, complexity of 3' end signal sequences increases gradually. Marchat L. A. et. al.'(2005) show that *E. histolytica* pre-mRNA 3' end processing signals differ from those described in human and yeast, but in contrast pre-mRNA 3' end processing factors are well conserved suggesting a high conservation of the mechanisms through evolution.

Experimental approaches for identification of introns and transcription start sites

Some of the criteria used for identification of exonic regions:

1. It must have an open reading frame;
2. It is likely to have related sequence in other species;
3. Uninterrupted gene map using restriction enzymes exactly corresponds to its cDNA restriction map.

While designing experimental approaches for detection of introns the above criteria are used. One of the simplest approaches is to use a polymerase chain reaction based method with and without reverse transcription (RT-PCR) with genomic DNA and cellular RNA. If introns are present, RT-PCR products will be shorter.

There are other more sophisticated methods, such as exon trapping. Here cloning vector containing intron with known restriction enzyme sites is used. Long nucleotide fragment, which is to be tested for presence of exon is inserted in the intronic region of the vector. That vector is transformed in appropriate host having the splicing machinery in it, for expression. If nucleotide fragment contains exon with its junction then splicing pattern is different otherwise it gets spliced out as usual intronic sequence.

Determining complete c-DNA sequence is the key method in order to hunt transcription start sites. In one approach, "foot prints" of transcription regulating proteins which bind to promoter and other regulatory sequences just upstream to transcription start site, are determined on the DNA sequence. Then entire gene sequence along with full length c-DNA sequence is analyzed in order to determine transcription start site. Recursive deletion of upstream sequence from negative (-ve)

residue/s to suspected TSS is helpful to determine minimal promoter as well as the transcription start site of the gene (Benjamin Lewin 2004).

Computational Methods for Intron/Exon prediction:

Intronic splice junctions are short sequences of seven to eight nucleotides. Splice sites are the sequence immediately surrounding the exon-intron boundaries. They are named for their position related to the introns. The 5' splice site at the 5' (left) end of the intron includes the consensus sequence GU. The 3' splice site at the 3' (right) end of the intron includes the consensus sequence AG. The GU-AG rule describes the requirement for these constant di-nucleotides at the first two and last two positions in pre mRNAs.

One of the major obstacles to accurately predict gene structures in eukaryotes is the presence of spliceosomal introns. Spliceosomal introns show large variations in intron numbers per gene, typically show no length or sequence conservation either within or between species and afford opportunities for alternative splicing, farther complicating accurate prediction of total proteome of an organism (Roy and Penny, 2007). Errors in genome assembly can lead to over prediction of introns, one such example involves the genome of *E. histolytica.*

Roy and Penny used intron length distribution to determine over prediction or under prediction of introns in given genome sequence. As introns are spliced out, and there length are not expected to respect coding frames, thus the number of introns that are multiple of three bases (3n introns) should be similar to the number that are multiple of three plus one bases (3n+1) or plus two bases. Skewed predicted intron length distribution thus suggested systematic error in intron prediction. That is genome wide excess of 3n introns suggest that many internal exonic sequences have been incorrectly called introns, whereas deficit of 3n introns suggest that many 3n introns that lack stop codons have been mistaken for exonic sequences (Roy and Penny, 2007).

Several programs to locate promoter elements in DNA sequence have been developed but they do not show a satisfactory level of accuracy because of the remarkable level of degeneracy of single promoter elements that contaminate prediction with huge number of false positive (Graziano Pesole et al. 2003). Valery Shepelev and Alexei (2008) reviewed advances in Exon- Intron Database (EID) and its newly added features. How one can use this database and which features are useful for model

building and farther predictions.

Gene Prediction:

Computational identification of complete gene model in eukaryote remains a challenging task till today. Difficulties in creating annotation arise for a variety of reasons. Sometimes the evidence for a gene is week consisting of just one gene prediction but no sequence homology, or just a single EST (Expression Sequence Tag) match. In other cases the evidences are plentiful but contradictory: different gene finders and protein sequence alignment may indicate many overlapping candidate genes and more than one of these models may in fact be correct (Steven L. Salzberg, 2004).

Pretty handy annotation tool (Phat) model Genomic DNA with generalized hidden Markov model (GHMM) similar to existing GHMM gene models such as GENSCAN, Genei and HMMgene, Glimmer M, but it was found that Glimmer M tends consistently to under annotate while Phat tends consistently to over annotate. Also Phat server returns abnormally short introns and long exons (Cawley and Speed, 2001).

One quarter of *E. histolytica* genes are predicted to contain introns, with 6% of genes containing multiple introns. *E. histolytica* genome was annotated with the Glimmer HMM, Phat and tran-scan (Fraser and Hall et al, 2005).

One approach is used for correct annotation is combiner approach in which these statistical algorithms take output of different '*ab initio*' gene finders as an input and annotate genome with greater specificity and sensitivity. Open source linear combiners as well as non linear combiners are available but better pipeline work and statistical approach is needed to use the same and selection of the '*ab initio*' gene finders (Steven L. Salzberg, 2004).

Pierre Rouze and Thomas Schiex (2002) reviewed many gene prediction methods, their strengths and weaknesses; but they focused on methods that only predict protein coding genes. They also summarized the EST based approach like EbEST, ESgenome, TAP and PAGAN, which commonly use the approach in which EST clusters are made based on similarity searches and then further used to shortlist most informative EST.

Some of the problems of using EST-based approaches are: a) ESTs are very redundant in nature and large number of them may be retrieved when performing BLAST search

against dbEST; b) ESTs are naturally error prone since they are generated from single read sequences; c) The prediction of "ENDs" of genes is difficult task as EST represents partial mRNAs.

Some novel statistical computational approaches, such as Variable Order Bayesian Network (VOBN) that generalizes PMW models, Markov models and Bayesian Network models for gene prediction. But these models worked well with sigma factor binding sites in "*E. coli*" but not that much well with eukaryotes (Grosse et al, 2005). Singh U. et. al. (2007) attempted use of naive Bayesian inference to identify genome wide transcriptional regulatory networks in *E. histolytica*.

Michael R. Brent and Roderic Guigo (2004) reviewed the *de novo* gene predictors that uses single genome sequence, two genome sequences and more than two genome sequences and usefulness of pipeline for the annotations. Single genome predictors are based on machine learning approaches such as HMM, for example, HMMgene. Whereas, duel and multi- genome predictors, are based on the conservation and the rate of evolution among them. But such approaches are quite useful only for compact eukaryotic genomes not for mammalian one.

Objectives of the thesis

1. To develop a method for detection of intron-exon boundaries in *Entamoeba histolytica* genes.

2. To develop a method for detection of transcription start sites in *E. histolytica* gene.

3. Rescanning present NCBI annotation of *E. histolytica* (HM1:IMSS) for splice junction using developed methods.

Methods

Creation of a database of true Introns

The intron prediction in *E. histolytica* has not been very accurate so far and there is no reported database of true introns so far. Therefore, we created a database of true introns. In order to do this, we analyzed the EST sequences available from *E. histolytica* in the NCBI database. We extracted putative introns based on the current annotation available for ameba (http://www.ncbi.nlm.nih.gov/sites/entrez?db=nucleotide&cmd=Search&term=DS571145%3ADS572673[PACC]). The extracted sequences were matched with the EST database using a local implementation of BLAST. The putative intronic sequences that showed nearly identical match were classified as false introns. The sequences that did not show any match were further analyzed to see if the corresponding genes are represented in the EST database. For these the entire putative coding regions were extracted and used for sequence similarity searches utilizing the EST database. The sequences that showed nearly identical match were further analyzed to see if exon sequences surrounding introns are present in an EST. If yes, then these were classified as "true exons" and sequence parting them shortlisted as "true intron". A description of the computation pipeline is shown in Fig. 1.

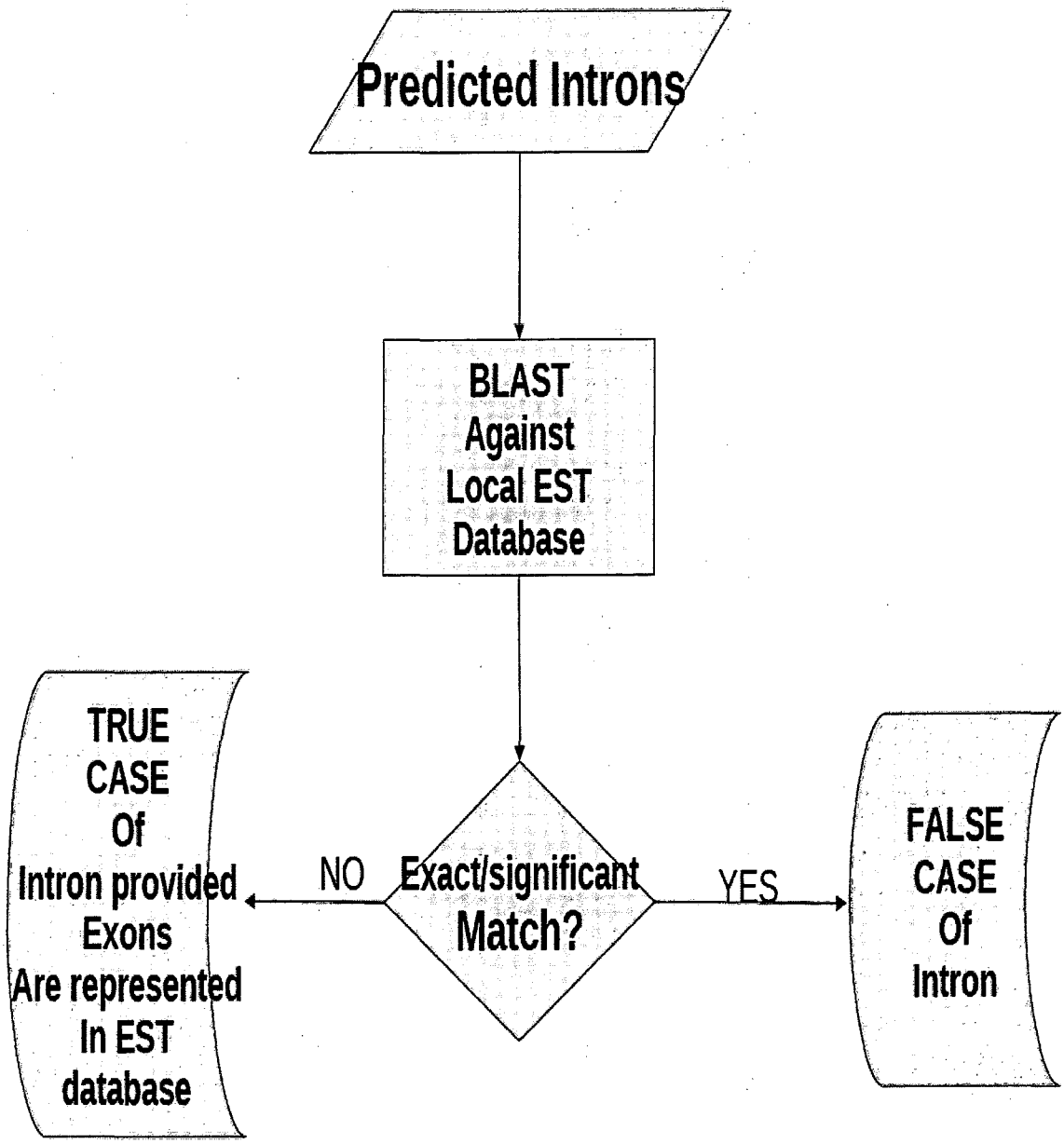

Fig. 1 : Creation of Database of true Introns

Creation of database of true Transcription Start sites

One way to find TSS is to obtain full length cDNAs of genes and then identify the 5'-sequence. There has not been any study carried out so far to find TSS for E. histolytica. This study is aimed at identification of TSS using EST data and computational methods.

Two different computation pipelines for identification of TSS from ESTs, used in this study are shown in Fig. 2 and 3. The rationales for developing these pipelines are based on the fact that ESTs are partial mRNA sequences and that the library may not represent all the transcripts. In the first pipeline the smaller genes (less than150 Amino Acid residues) were selected as the chances of getting their 5'-UTR is much higher. These sequences were aligned with ESTs after translation using tblastn utility of local BLAST. The ESTs showing identical matches on translation with entire protein length and still possessing untranslated "ENDs" were shortlisted as putative full length cDNAs. These were further validated by checking if the ESTs contain the entire gene including start and stop codons and match with annotated putative protein sequence present in NCBI and Pathema databases (give Pathem URL). A description of computational pipeline is shown in figure 2.

In the second approach, 100 nucleotides were extracted from the 5'-end (+1 to +100) of all the annotated genes. These sequences were then used to search E. histolytica EST database. The ESTs that showed sequences beyond the +1 site, that is translation initiation site, were identified and shortlisted for further studies. If these represent truly TSS then the start of the sequence is synonymous with TSS and it is possible that the length of the sequence reads in these clones may be smaller than others. On further analysis it appeared that the average size of these sequences are smaller than others.

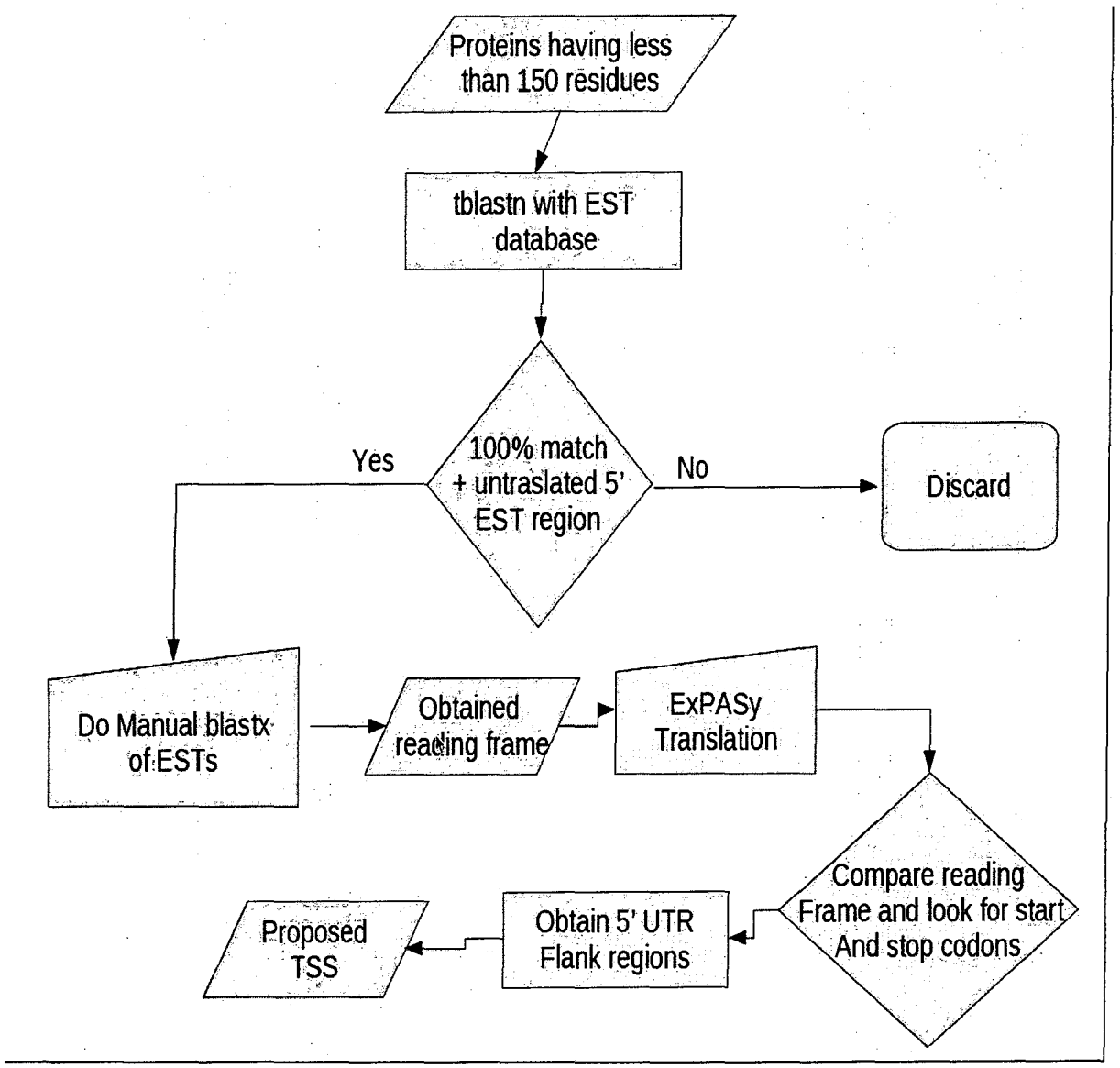

Fig. 2: Creation of Database of TSS

Building PSSM and its implementation

Position Specific Scoring Matrix represents model that is based on the concept that for a given regular pattern of sequence of DNA and/or amino acid, each position has been evolved independent of other and is a way to represent and analyse motifs.

A PSSM is a matrix of score values that gives a weighted match to any given substring of fixed length. For DNA sequence it has rows representing different nucleotides and columns the position in the motif. Given a derived matrix the score of a given motif can be determines as,

$$s = (s_j)_{j=1}^{N}$$

is defined as

$$\sum_{j=1}^{N} m_{s_j,j},$$

where j represents position in the substring, s_j is the symbol at position j in the substring, and $m_{\alpha,j}$ is the score in row α, column j of the matrix.

Here we used log odds for the convenience of additions as follows,

$m_{i,j} = log(p_{i,j} / b_i)$,

where $p_{i,j}$ is the probability of observing symbol i at position j of the motif, and b_i is the probability of observing the symbol i in a background model. Experimentally verified sequences containing the desired characters, that is splice donor and acceptor sequences, TSS etc are aligned in a multiple sequence alignment (MSA) block and used to generate the scoring matrix by using the equation described above. Trained model consists of log - normalized frequencies for each position. A scoring function is then defined to score other test sequences using model. For training PSSM model for Intron/exonic boundaries and for transcription start site, we first aligned our candidate sequences in MSA form. Then MSA blocks were subjected to training procedures, which were consist of determination of frequencies of each nucleotide base for each position, normalizing the frequencies with the background frequencies obtained from entire amoeba Genome, and finally taking logarithm of the normalized frequencies to the base 2 in order to easing computation of the score using scoring function. Then we used trained model to determine specificity sensitivity of the model, p value of the predictions (both described in miscellaneous methods), and to scan the sequence for desired pattern using sliding window algorithm. For the implementation of all these procedures, we wrote Perl codes and pipelines.

Building Markov models and their implementation:

In some patterns of sequences of DNA and/or protein, there is correlation between existence of groups of bases / residues, such coexistence can be modeled using principles of Markov process.

Markov process:

Conceptually, Markov process is the process where the future development is completely determined by the present state and is independent of the way in which the present state has developed. In stochastic processes, the future is not uniquely determined, but we have at least probability relation relating to future depends on the present state, but not on the manner in which present state has emerged from the past. In other words, if two independent systems subject to transition probabilities happen to be in the same state, then all probabilities relating to their future development are identical.

Markov chain:

A sequence of the trials with the possible outcomes E1, E2, ..., En is called a Markov chain if the probabilities of sample sequence are defined by

$$P\{(Ei_0, Ei_1, ..., Ei_n)\} = ai_0pi_0i_1 ..... pi_{n-2}i_{n-1}pi_{n-1}i_n$$

in the terms of probability distribution {ak} for Ek at the initial (or zeroth) trial a fixed conditional (joint) probability p_{ik} of Ek give that Ei has occurred at the preceding trial.

Briefly, a Markov chain is a process defined by a set of states (in our case, symbols of nucleotides) and by a transition probability from one state to the next. The transition probabilities are organized in the transition matrix T. The trajectory of the process through the state space defines a sequence, as shown in figure,

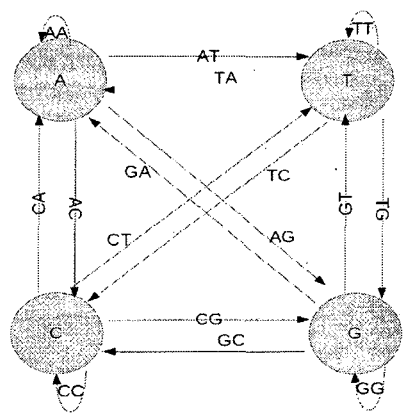

Fig. 3 Markov Trajectory For DNA sequence

We used following consideration for the choice of model, when evolution at a given position in pattern is depended on history of evolution of previous position/s then the evolution of pattern follows Markov process, and the order of Markov process is exactly equal to the number of previous position/s, for example if existence of particular base/residue at position "n" in a sequence is depended on existence of particular base/residue at position "(n-1)", then the pattern can be modeled using 1^{st} order Markov model and that depend on both "$(n-1)^{th}$ and $(n-2)^{th}$ position then 2^{nd} order Markov model is the good choice". Here we attempted to build up to third order Markov model.

First order Markov model, training and implementation:

In our first order Markov model (Fig. 3), we emphasized both on position vise transitions and individual positions. While training the model using aligned sequences, transition to immediate next position from previous one is captured in transition matrices, representing dinucleotide frequency matrix of order 4X4 as shown in the following table.

	A	T	G	C
A	jf(AA)	jf(AT)	jf(AG)	jf(AC)
T	jf(TA)	jf(TT)	jf(TG)	jf(TC)
G	jf(GA)	jf(GT)	jf(GG)	jf(GC)
C	jf(CA)	jf(CT)	jf(CG)	jf(CC)

(jf = Joint Frequency)

In each transition matrix, we did "small sample correction" as we have few number of training sequences (less than 100), and also in order to avoid formation of spare matrix. We achieved this by using following formula,

$SC_{i,j} = (S_{i,j}+1) / ((S_{i,1}+S_{i,2}+S_{i,3}+S_{i,4}) + 4)$,

where "$Sc_{i,j}$" was corrected entry at (i,j) position after correction, "$S_{i,j}$ " was original score at (i,j) th place

We define position vectors corresponding to each position in the sequence, as {1[A], 2[T], 3[G], 4[C]}i

Where "i"represents the index of position. Each element represents distribution of the nucleotide shown in brackets "[ ]" at position i, obtained by Markov process as shown in figure (), except initializing vector (π Vector) shown as RED block, which was obtained by calculating the frequencies of nucleotides at first position of MSA block of training data set.

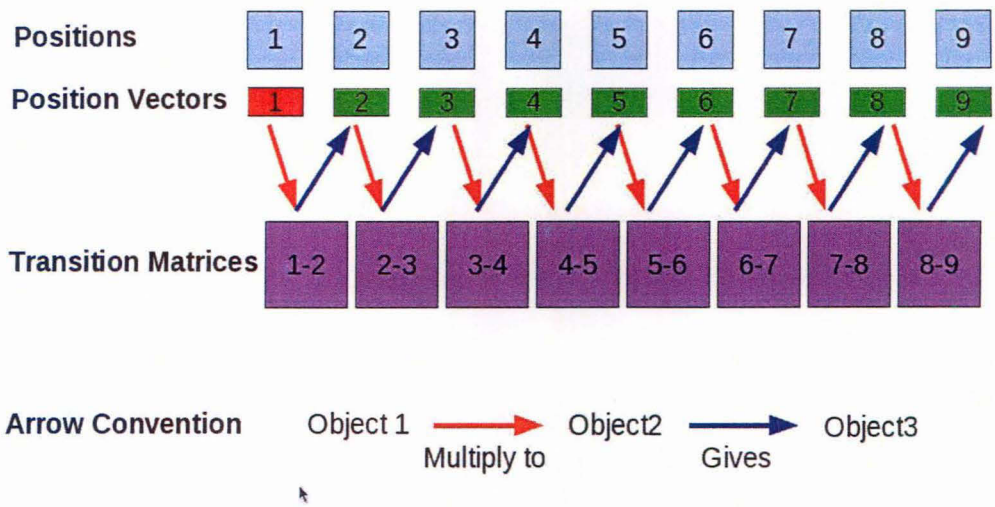

Fig. 4 Architecture of First Order Markov Model

"ith" position vector is obtained by multiplying (i -1)th position vector to the "{(i -1) to i} transition matrix" and so on.

Normalizing and using trained model to score test sequences :

After generation of transition matrices and position vectors by the procedure described above, we normalized the entries both in transition matrices and position vectors as follows. Till up to this stage all entries in data - structure (matrices and vectors) were in the form of frequencies, so we could not use those directly for scoring. To serve the purpose, we normalized transition matrices entries by dividing each entry with the corresponding dinucleotide frequencies from background (that was obtained from whole genomic sequence) and then taking logarithm of the value to the base 2. Entries in position vectors are also log - normalized to the base 2 with the background frequencies of mono nucleotides obtained from the genome of the parasite.

Once the model got trained, we defined scoring function to score test sequences. We consider a window whose size was equal to the length of multiple sequence alignment with which we trained the model. Then for a sub-sequence of the test sequence having length equal to the size of window was scored as follows

$$score = \left(\sum_{i=1}^{i=n} Vs_{i,j(A \vee T \vee G \vee C)}\right) + \sum_{i=2}^{i=n} \sum_{k=(i-1)}^{k=i} Tmat_{k,l(A \vee T \vee G \vee C),m(A \vee T \vee G \vee C)}$$

Where, "i" denotes the index of position numbers in the window of size "n". "Vs" is an array of position vectors where "j" represents column index from where log - normalized score is to be picked according to the nucleotide being recognized at position "i". "Tmat" represents the transition matrix and "k" is the index of transition matrix as there will be "n - 1" number of transition matrices for a given sub - sequence of size "n". "l" is a "row" index and m is a "column" index of "k^{th}" transition matrix where the log normalized values are stored and those row and column indices will be depend on dinucleotide recognized at transition.

Specificity and sensitivity curve and ROC curves were obtained then according to procedure mentioned in the miscellaneous methods.

Second order Markov model, its training and implementation:

In our second order Markov model, we modeled trinucleotide occurrence in the target type sequence. We had following considerations to build our Second order Markov model. Our model consisted of three data structures namely array of trinucleotide transition frequency matrices, another array of dinucleotide frequency position vectors and finally an array of "third" position frequency vectors. We used ungapped multiple sequence alignment to train second order Markov model. We captured trinucleotide transition frequencies in matrices of order 16 X 4, while training the model in a following manner:

	A	T	G	C
AA	AAA	AAT	AAG	AAC
AT	ATA	ATT	ATG	ATC
AG	AGA	AGT	AGG	AGC
AC	ACA	ACT	ACG	ACC
TA	TAA	TAT	TAG	TAC
TT	TTA	TTT	TTG	TTC
TG	TGA	TGT	TGG	TGC
TC	TCA	TCT	TCG	TCC
GA	GAA	GAT	GAG	GAC
GT	GTA	GTT	GTG	GTC
GG	GGA	GGT	GGG	GGC
GC	GCA	GCT	GCG	GCC
CA	CAA	CAT	CAG	CAC
CT	CTA	CTT	CTG	CTC
CG	CGA	CGT	CGG	CGC
CC	CCA	CCT	CCG	CCC

If training sequences forming ungapped MSA were each of length equal to "n", then there would be (n - 3 + 1) overlapping trinuceotide transitions in each sequence. Thus we created n - 3 + 1 transition matrices as shown in figure.

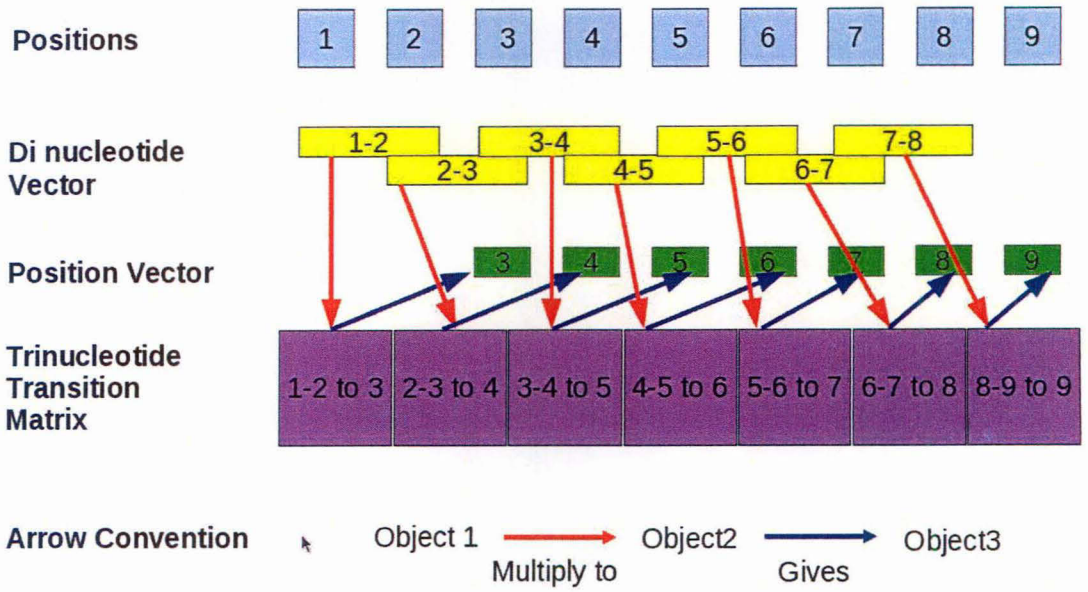

Fig 5. Architecture of Second Order Markov model

We also calculated dinucleotide frequencies for each dinucleotide overlapping positions from the same MSA and for each dinucleotide overlapping position we created dinucleotide frequency vector consisting sixteen elements in each. As we were modeling transition to given state (here nucleotide alphabet), from given pair of nucleotide previous to it, the resulting third alphabet had to be given importance. We derived "third position vectors" by taking "cross product" or vector product of each dinucleotide position vector and corresponding trinucleotide transition matrix. Now all entries in data structures were frequencies, thus, to ease the calculation of score, we normalized the entries by taking log odds. To normalize transition matrices, we generated background trinucleotide frequencies from parasite genome and using them we log normalized the frequencies by taking logarithm to the base 2 of the ratio of observed trinucleotide frequencies to the expected one from the background. Then we also log normalized the dinucleotide position vectors and third position vectors using di and mono nucleotide frequencies from background respectively. For all these procedures we wrote our own Perl programs.

Scoring test sequences using trained second order Markov model:

In order to score test sequences using trained model, we define a window of size that is equal to the length of MSA we used to train the model. For the sub sequence of a test

sequence having length "n" that was equal to the window length we calculated score using following scoring function,

$$Score = \sum_{i=3}^{i=n} TPV_{i,j(A \vee T \vee G \vee C)} + \sum_{i=2}^{i=n} \sum_{k=(i-1)}^{k=i} DNPV_{k,l(A \vee T \vee G \vee C),m(A \vee T \vee G \vee C)} + \sum_{i=3}^{i=n} \sum_{a=(i-2)}^{a=i} TNTM_{a,p(A \vee T \vee G \vee C),q(A \vee T \vee G \vee C),r(A \vee T \vee G \vee C)}$$

where, "i" is the index of the any of the "n" positions. TPV is the array of "third position vectors" in the 2D matrix form, in which each row corresponds to the one "third position vector". "j" represents the columns of vector correspond to four nucleotide alphabet and consisting log normalized frequencies of them for "i^{th}" position, out of them, which value is going to be considered for calculation is depend on which nucleotide symbol is being recognized at that position in windowed test sequence. "DNPV" is the array of dinucleotide position vector. While training the model, our Perl code generates these dinucleotide vectors in linear 1D format, so as to ease multiplication with transition matrices. Once trained, our one subroutine transforms each dinucleotide array to 2D matrix and stacks each matrix into one single array to ease the scoring procedures. Thus, DNPV, being 3D array, shows three layers, outer layer "k" indicates the position in the window from which dinucleotide in consideration starts, middle layer "l" and inner one "m" correspond to the first and second nucleotide respectively and point out the place in the k^{th} matrix where log normalized joint frequency is stored for the dinucleotide under consideration. "TNTM" is the array of trinucleotide transition matrices. Being 4D in nature, TNTM shows 4 layers, outer "a" indicates the position in the sequence from which trinucleotide in consideration starts, inner three layers "p", "q" and "r" correspond to the first, second and third nucleotide respectively and point out the place in the "a^{th}" matrix where the log normalized joint frequency of the trinucleotide under consideration is stored.

We, then obtained specificity an sensitivity curve, as well as ROC curves for the model.

Third order Markov model, its training and implementation:

here we modeled tetra nucleotide occurrence in the parasite genome using third order Markov models. We used ungapped MSA of target type sequence to train our model. Our model consists of an array of tetra nucleotide transition matrices. While training the model with MSA, our code generate transition matrices of dimension 64 X 4 for each overlapping transition. These matrices had following formats,

	A	T	G	C
AAA	AAAA	AAAT	AAAG	AAAC
AAT	AATA	AATT	AATG	AATC
--	--	--	--	
CCC	CCCA	CCCT	CCCG	CCCC

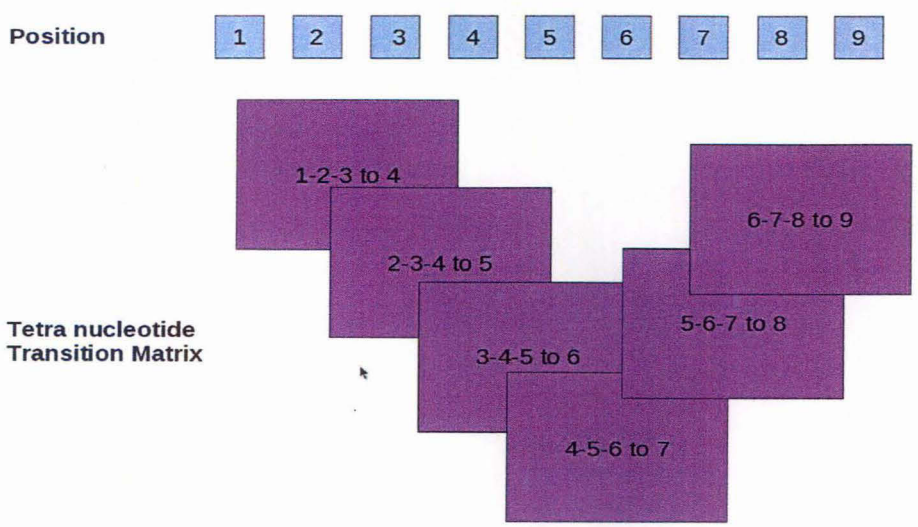

Fig. 6. Architecture of Third Order Markov Model.

We filled matrices with the tetra nucleotide frequency occurrence at each overlapping transitions. If we have sequence of length "n", then there will be "(n - 4 +1)" possible transitions, thus we generated (n - 4 + 1) transition matrices to build the model. Then in order to use model to score the test sequences for our target pattern, we normalized the tetra nucleotide frequencies with the background frequencies that we obtained from the

parasite genome. Then we took logarithm of each normalized ratio to the base to in order to ease calculation of the score. These converted values in data structure are then restructured in single 5D matrix architecture to ease the scoring. We used the following scoring formula to score test sequences. We decided a window of length "n" that was equal to the length of MSA we used to train.

$$Score = \sum_{i=4}^{i=n} \sum_{k=(i-3)}^{k=i} TRNTM_{k,l(A \vee T \vee G \vee C),m(A \vee T \vee G \vee C),n(A \vee T \vee G \vee C),o(A \vee T \vee G \vee C)}$$

Where, "i" is the index of the any position among the "n" positions in the sequence. Being 5D matrix architecture, TRNTM has 5 layers in it. Outer layer "k" indicate the index of transition, or the place from which transition is to be considered. Inner four layers correspond to the tetra nucleotide combination, and the combination under consideration at given instant, point out that place in 5D matrix, where the log - normalized value for that tetra nucleotide combination is stored.

Then we obtained specificity - sensitivity curves and ROC curves for the model for different target patterns.

Integrated use of model to cross-verify the NCBI annotations:

We scanned available NCBI annotation for splice junction with our models. For a given annotated splice site if at least three of the four models were predicting as true splice junction (DONOR/ACCEPTOR) , we labeled sequence as "true". Annotations which were predicted as false by all four models, we labeled them as false.

Training And Testing Stochastic Models for TSS:

Since We had very few (16) verified sequences of 5' UTR, training set and testing set for "true" were same. 16 "true" sequences and 16 sequences from background "false" were the candidates of validation set, using those we obtained specificity – sensitivity curves and ROC curves for our models

Integrating models in order to use SVM classifier based on the scores:

SVM:

Support vector machines (SVMs) are a general class of learning architectures, inspired by the statistical learning theory that performs *structural risk minimization* on a nested set structure of separating hyperplanes. Given a training data, the SVM learning algorithm generates the optimal separating hyperplane in terms of generalization error. SVMs have been found to be very useful in handling data mining problems. The basic idea is to construct a hyperplane as the decision surface such that the margin of separation between positive and negative examples is maximized. The *structural risk minimization* principle is used for the purpose. Here the error rate of a learning machine is considered to be bounded by the sum of the training error rate and a term depending on the Vapnik-Chervonenkis (VC) dimension. The VC dimension of a system is defined as the largest set S of data samples for which the system can implement all possible 2^s dichotomies on S, where dichotomy implies a two-class categorization.

Here we attempted to use scores generated by all the stochastic models described previously (PSSM, Markov 1st, 2nd and 3rd order models) for preverified patterns like "DONOR - NON DONOR", "ACCEPTOR - NON ACCEPTOR", to build SVM model. We wrote subroutines to integrate all models and to generate input vectors in the prescribed format of "lib - SVM" software. In training vectors, we did not only include the scores of the models, but also the dinucleotide frequencies as well as mono nucleotide frequencies. Then we rescaled the data between -1 and +1 using SVMscale utility of lib - SVM. To decide suitable SVM parameters such as kernel type function (linear, radial, polynomial etc), and other parameters such as "gamma of kernel function - g", and "cost parameter - c" we used pre - written python script (provided by lib - SVM). We used the resultant parameters to train the SVM models with our scaled data. We farther did 5 fold cross-validation and obtained ROC for the integrated model using lib - SVM toolbox.

Integrating models in order to use j48 (decision tree) classifier based on scores of the model:

Decision tree classifier is one of the most widely used supervised learning methods used for data exploration. It is easy to interpret and can be re-represented as *If-then-else* rules. It approximates a function by piecewise constant regions and does not require any prior knowledge of the data distribution. This classifier works well on noisy data. A decision tree aids in data exploration in the following manner:
It reduces a volume of data by transformation into a more compact form, that preserves the essential characteristics and provides an accurate summary. It discovers whether the data contains well-separated classes of objects, such that the classes can be interpreted meaningfully in the context of a substantive theory. Also, it maps data in the form of a tree so that prediction values can be generated by backtracking from the leaves to its root (Susmita Mitra and Tinku Acharya, (2003)). This may be used to predict the outcome for a new data or query. For example, if one have data of accidents for families and sports persons those driving cars and their ages, a simple decision tree was built out of all those as follows,

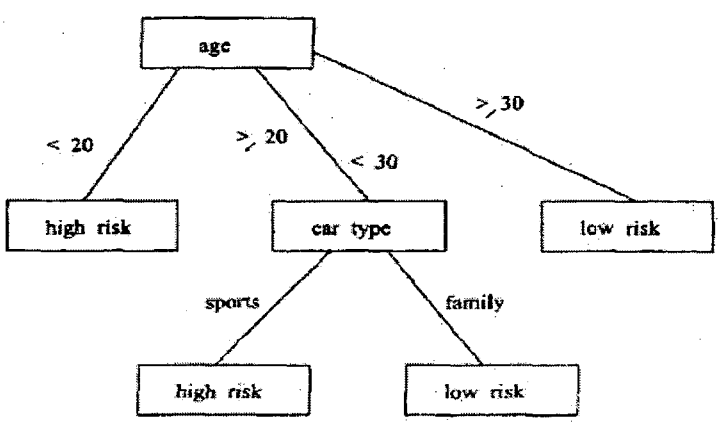

Fig. 7 Example of Decision Tree

(Figure adapted from Susmita Mitra and Tinku Acharya (2003))

we used WEKA tool to perform j48 classification. First of all we integrate all the models (PSSM, 1st , 2nd and 3rd order Markov models) to generate scores and other attributes in the "Attribute - Relation File Format " (.arff file format), then we trained the model with verified true and false cases using above attributes and cross - validated with 5 fold cross validation, accuracy, ROC and "confusion matrix" of the classification were determined using utilities provided with WEKA.

Miscellaneous Methods:

Specificity and Sensitivity analysis of the models:

Trained models were subjected to specificity and sensitivity analysis. Pre - verified true cases and false cases were labeled as true and false respectively and subjected for testing model on them. Prediction of model and tagged labeled were compared and number of True Positive (TP), False Positive (FP), True Negative (TN) and False Negative were determined using Perl codes. Specificity and sensitivity were calculated as follows;

Sensitivity = TP / (TP + FN);

Specificity = TN / (TN + FP);

Sensitivity and specificity plot is then obtained using output data.

ROC curve:

To obtain ROC (Receiver Operator Characteristic), above specificity and sensitivity values were used, plotting of **Sensitivity Vs (1 - specificity)** gives ROC curve.

P Value for Models:

If test sequence is of length "n" and model is using window size of W then for a given cutoff score value, p value can b calculated as follows;

p Value = number of hits at given cutoff / (n - W + 1);

we run our models over range of cutoffs and then we plotted p value against cutoff.

Results and discussion:

Computational annotation of *E. histolytica* is a really challenging job as no heuristic or supervised learning method for sequence annotation has yet been proved accurate method so far. One of the major problem in carrying out automated annotation of E. histolytica genomic sequences is the absence of experimentally verified true "genome structures" like transcription start sites, splicing junctions, transcription termination sites and poly A signals. We also have very rudimentary knowledge of the mechanisms of gene expression and regulation in *E. histolytica.*

This study has been designed to create a database of true splice junctions (Donors and Acceptors flanking intronic sequences) and transcription start sites using the EST database of *E. histolytica.* Subsequently, these data sets were used to build stochastic models (PSSM and Markov models) for correct prediction.

Preparation of splice site data sets:

The computation pipeline for identification of true and false introns, has been described in "methods". The prediction based on the principle that intronic sequences are not likely to be present in EST sequences as these get splice out. Presence of sequences in EST database means that these sequences are not introns. Sequences, that were annotated as introns previously (http://www.ncbi.nlm.nih.gov/sites/entrez?db=Nucleotide&cmd=Search&term=DS571145:DS572673[PACC]) and showed significant match with an EST sequence were sorted out and labeled as "False cases"(Table 1). The true introns were identified on the basis of EST hits that do not include predicted intronic regions, that is, the predicted intronic regions were absent in EST database. About 72 cases of true introns were identified as shown in table 2.

Table 1

False Cases:

Scaffold number	annotation
DS571145	(<404713..404807,404866..405451>)
DS571145	(<382939..382949,383033..383513>)
DS571152	(<14512..14694,14742..14936>)
DS571152	(<15041..15046,15102..15341>)
DS571192	(<59714..59925,60062..60493>)
DS571207	(<56533..56870,56918..57140>)
DS571232	(<32276..32316,32390..32861>)
DS571286	(<30559..30837,30898..31206>)
DS571297	(<9820..10450,10521..10870>)
DS571314	(<35041..35607,35649..35675>)
DS571538	(<4594..4801,4895..5487>)
DS571600	(<8591..8946,9066..9090>)
DS571869	(<673..821,845..1271>)
DS571977	(<566..1003,1078..1302>)

Table 1

Table 2
True Cases:

Scaffold	position
DS571145	(<404591..404665,404713..404807>)
DS571145	(<415062..415286,415342..415928>)
DS571145	(<279142..279257,279396..279997>)
DS571145	(<284512..284673,284736..285174>)
DS571146	(<38488..38490,38554..38854>)
DS571147	(<21669..21887,21937..22438>)
DS571148	(<170978..171678,171723..171876>)
DS571148	(<121860..122331,122405..122445>)
DS571150	(<84257..84316,84366..84909>)
DS571150	(<84177..84206,84257..84316>)
DS571152	(<11041..11234,11296..11738>)
DS571155	(<22372..22798,22855..23037>)
DS571155	(<22180..22296,22372..22798>)
DS571164	(<103785..103787,103903..104367>)
DS571166	(<71641..71691,71740..71928>)
DS571174	(<31999..32095,32150..32685>)
DS571174	(<31761..31940,31999..32095>)
DS571174	(<21238..21248,21315..21777>)
DS571175	(<70540..70770,70824..71240>)
DS571175	(<70478..70480,70540..70770>)
DS571180	(<49841..50433,50483..50665>)
DS571183	(<38654..39346,39408..39528>)
DS571184	(<86188..86669,86726..86728>)

DS571189	(<35353..35665,35718..36035>)
DS571190	(<8887..9024,9081..9543>)
DS571192	(<71572..72026,72096..72250>)
DS571195	(<46222..46755,46842..47135>)
DS571210	(<37618..38293,38341..38456>)
DS571210	(<38341..38456,38513..38549>)
DS571211	(<40355..40690,40750..41152>)
DS571219	(<26301..26765,26881..26889>)
DS571219	(<41659..42106,42173..42371>)
DS571220	(<43526..43918,43972..44224>)
DS571224	(<16761..17086,17156..17396>)
DS571224	(<53694..53920,53971..54428>)
DS571224	(<19720..19863,19937..20417>)
DS571226	(<27366..27466,27560..27857>)
DS571226	(<27088..27277,27366..27466>)
DS571233	(<38583..38716,38786..39395>)
DS571244	(<45045..45364,45481..45820>)
DS571246	(<16637..16786,16842..17250>)
DS571248	(<30975..31146,31243..31490>)
DS571260	(<27929..28447,28577..28696>)
DS571275	(<20217..20275,20338..20920>)
DS571286	(<29965..30267,30320..30390>)
DS571300	(<7208..7502,7555..7815>)
DS571301	(<24679..24985,25038..25113>)
DS571301	(<25038..25113,25178..25443>)

DS571316	(<5076..5288,5338..5592>)
DS571316	(<29477..29609,29662..30267>)
DS571323	(<27652..27748,27852..28434>)
DS571332	(<8195..8209,8257..9093>)
DS571347	(<16029..16367,16465..16752>)
DS571347	(<16465..16752,16810..16821>)
DS571393	(<5124..5393,5440..5960>)
DS571414	(<10581..10813,10872..11498>)
DS571416	(<5256..5517,5573..5959>)
DS571416	(<5573..5959,6008..6049>)
DS571422	(<6886..7182,7251..7556>)
DS571422	(<6793..6807,6886..7182>)
DS571441	(<12297..12424,12482..13113>)
DS571481	(<9160..9667,9774..9817>)
DS571487	(<9711..9716,9770..10432>)
DS571489	(<7327..7656,7715..8003>)
DS571506	(<33..127,200..362>)
DS571528	(<7666..7954,8018..8276>)
DS571678	(<105..306,358..830>)
DS571718	(<3283..3891,3947..3980>)
DS571748	(<2059..2753,2799..2966>)

Table 2

Transcription start site data sets:

In E. histolytica TSS of only a handful of genes have been experimentally identified. Due to the partial nature of ESTs and the fact that these are all derived from 5'-end of mRNAs it is quite often difficult to identify TSS from EST database. In order to find out 5' UTR, we have to find full length EST. We used two approaches to find out TSS and 5' UTR as described in methods. Our first pipeline was time taking and tedious. We had chosen those proteins of E. histolytica possessing about 150 or less amino acid residues, as chances of finding full length ESTs were more. We got about 25 genes that are likely to be full length. Careful blastx analysis of shortlisted ESTs and comparison with the conceptual translated products helped to identify true TSS and 5'-UTR of 16 genes (Table 3).

In our second approach we used existing annotation and retrieved upstream sequences (-100 to 0) of every annotated genes for blast analysis using EST database as subject. EST sequences that showed 100% matching with alignment length more than 50, were short listed. These were subjected to conceptual translation and blastx to confirm UTR region. About 80 such EST matches were observed (Table 4). It is clear from the results that most of the E. histolytica UTRs are small in length varying between 44 to 110 and unlike the ones found in higher eukaryotic organisms.

Table 3
5' UTR:

Sequence	source EST
AATTCTTTTTTAAATTTAATTAATT	CX089020.1
CTTAAAGATGTCAACTTATGAAGTA	CX085796.1
CTTCAAAGACACTATTAGTTAATAT	CX096076.1
TTTAATTCACTTCATTAGTATGAGT	CX093591.1
TTTTCTTATTCTTTATTCAACAGAA	CX081590.1
TTAAGTTGGTGTTTTGATTTAATTG	CX087915.1
TAGCAGCAACAGAAGCTACCGAGGC	CX079632.1
TTAGCAGCAACAGAAGCTACCGAGG	CX081289.1
TAACTTGATATCAAGTGGTTACCGA	CX082230.1
TCACTTTTTGTCAATATAGGACAAA	CX082773.1
AGCAACAGAAGCTACCGAGGCTTCA	CX086686.1
ACAGAAGCTACCGAGG	CX084873.1
TAGCAGCAACAGAAGCTACCGAGGC	CX087994.1
AATTTGAAAAGAAGAATAAGAACTG	CX087751.1
TGATTGATAAGTGTTTAATTTTTTA	CX087421.1
TTAAGAAAAAATTTAAATTAATTAA	CX082080.1

Table 4

(format >scaffold id_position)

>DS571147_69666_upstream
>DS571149_139148_upstream
>DS571150_112325_upstream
>DS571150_150537_upstream
>DS571151_145688_upstream
>DS571152_15041_upstream
>DS571153_59211_upstream
>DS571154_20607_upstream
>DS571154_37738_upstream
>DS571154_60516_upstream
>DS571156_31576_upstream
>DS571156_52120_upstream
>DS571157_62283_upstream
>DS571158_39382_upstream
>DS571159_49091_upstream
>DS571159_63376_upstream
>DS571159_78953_upstream
>DS571160_49214_upstream
>DS571160_63726_upstream
>DS571160_105154_upstream
>DS571160_108495_upstream
>DS571162_37922_upstream
>DS571164_120201_upstream
>DS571165_34756_upstream
>DS571168_42219_upstream
>DS571169_18502_upstream
>DS571169_106178_upstream
>DS571170_91118_upstream
>DS571170_93468_upstream
>DS571172_74910_upstream
>DS571173_85064_upstream
>DS571177_20754_upstream
>DS571178_13864_upstream
>DS571183_39464_upstream
>DS571186_3632_upstream
>DS571188_76919_upstream
>DS571195_17703_upstream
>DS571195_50053_upstream
>DS571197_63887_upstream
>DS571200_46287_upstream
>DS571200_51551_upstream
>DS571208_5151_upstream
>DS571209_52450_upstream
>DS571229_31715_upstream
>DS571238_43531_upstream
>DS571241_35369_upstream
>DS571256_44450_upstream
>DS571260_15804_upstream
>DS571266_30851_upstream
>DS571272_10454_upstream
>DS571284_14708_upstream
>DS571287_30979_upstream
>DS571292_17277_upstream
>DS571296_29420_upstream
>DS571300_13997_upstream
>DS571301_24679_upstream
>DS571304_11608_upstream
>DS571322_21475_upstream
>DS571330_10796_upstream
>DS571340_24755_upstream
>DS571341_22063_upstream
>DS571345_15803_upstream
>DS571346_689_upstream
>DS571347_8318_upstream
>DS571355_4225_upstream
>DS571388_6598_upstream
>DS571390_19590_upstream
>DS571419_1282_upstream
>DS571449_1663_upstream
>DS571458_3352_upstream
>DS571518_4517_upstream
>DS571547_2084_upstream
>DS571548_10129_upstream
>DS571578_860_upstream
>DS571615_5382_upstream
>DS571708_4489_upstream
>DS571869_673_upstream
>DS571885_1610_upstream
>DS571910_2080_upstream
>DS571950_934_upstream

Stochastic models and splicing junctions (DONOR / ACCEPTOR):

From the true intron sequences of Table 2 we retrieved their 5' flanking junctions (labeled as DONOR) and 3' flanking junctions (labeled as ACCEPTOR). Then we aligned all "DONORS" and all "ACCEPTORS" separately in the form of ungapped columns in order to train stochastic models. At a time, we could train our models for one type of pattern only, either for DONORs or for ACCEPTORs.

Results for PSSM:

The PSSM models were generated as described in "methods", separately for DONOR and ACCEPTOR using only part of the dataset. These were then used for testing on positive and negative datasets to determine specificity, sensitivity and p value for different cutoffs.

The model that trained on DONOR sequences showed 62% optimum specificity sensitivity at cutoff score of 0.5, with p value 0.2. Significant p value of 0.05 or 5% level of significance can only be obtained if the cutoff score is kept between 3 to 5. At this level of cutoff score, specificity was increasing up to 85% at the cost of loss of sensitivity (40%) . Thus PSSM model for DONOR does not appear to be able to capture the splice donor signal effectively. Interestingly, the model for ACCEPTOR showed optimum specificity and sensitivity at cutoff score 3.5. Analysis of p values suggested that cutoff range, 3.5 to 4 is optimum to achieve 5% level of significance. ROC curve result obtained for ACCEPTOR was also better than that of DONOR. Further, Mann Whitney rank test were done to obtain single numerical value that is equivalent to Area Under Curve of ROC. For donor Mann Whitney "U" was 83.5 where as that of for ACCEPTOR was 111.5.

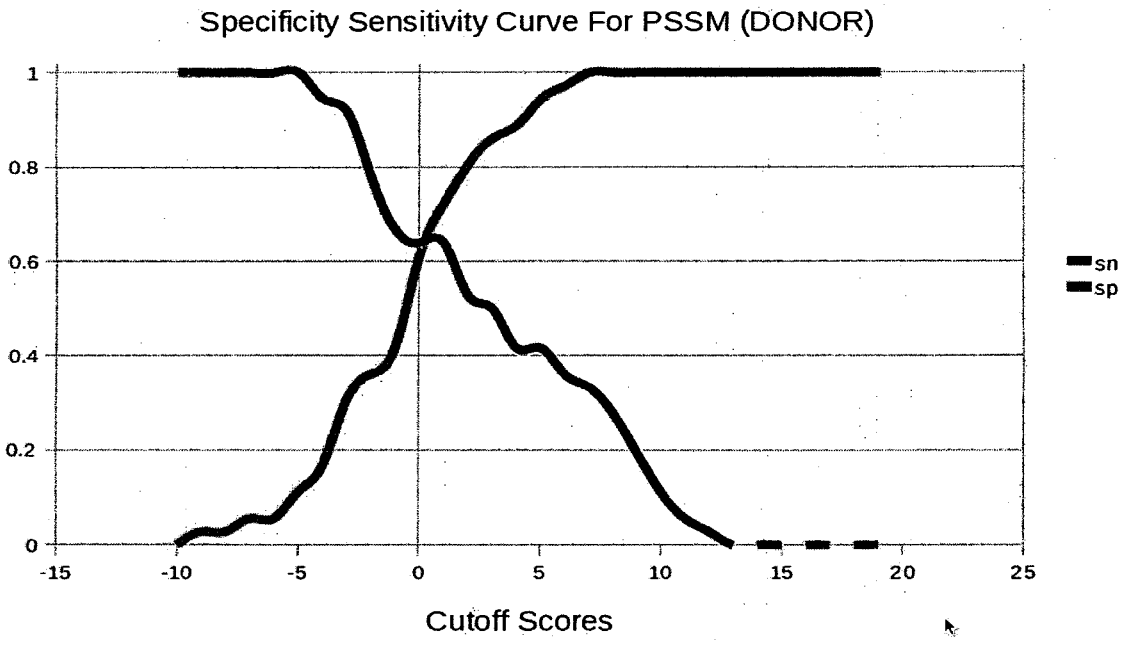

Figure. 8

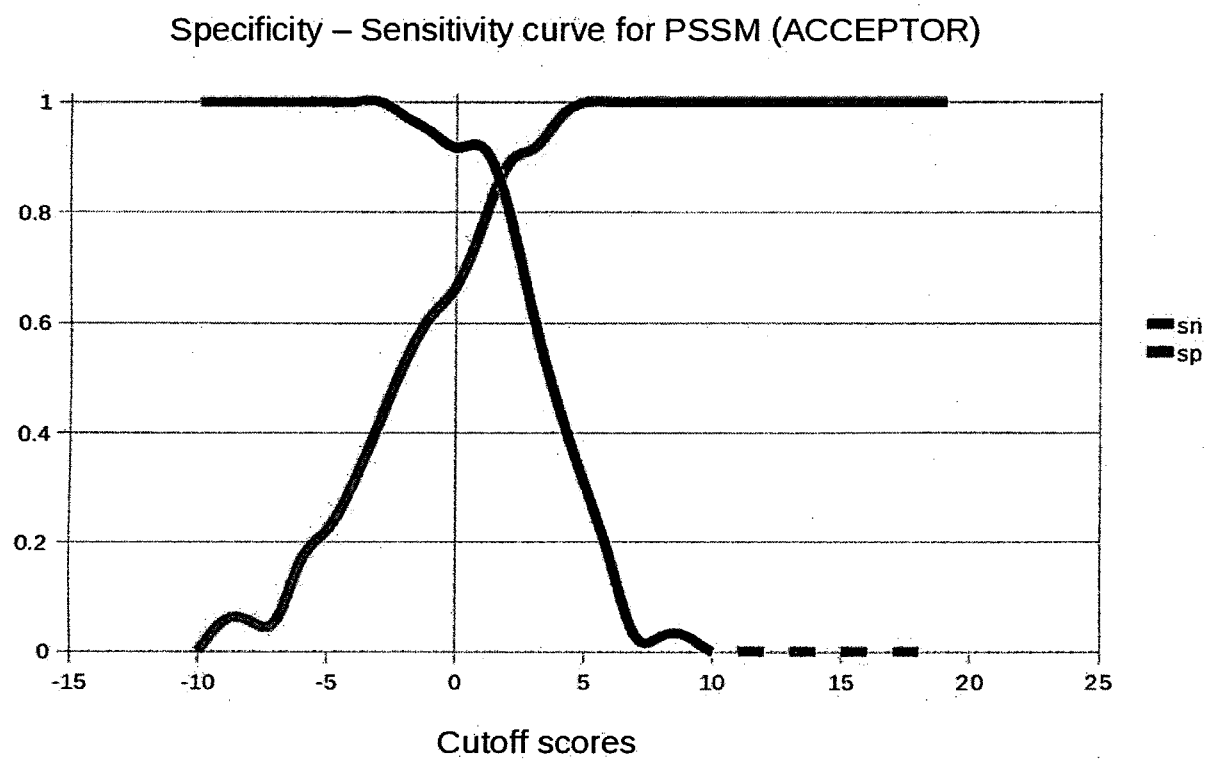

Figure 9

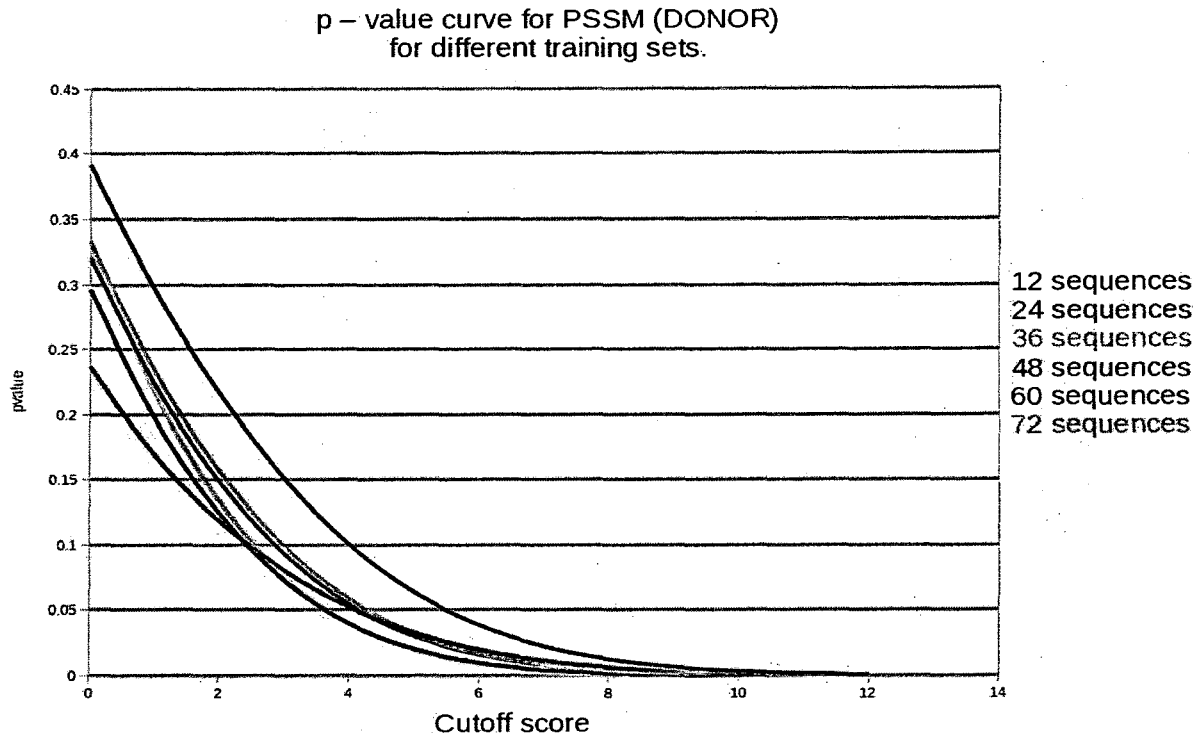

Figure 10

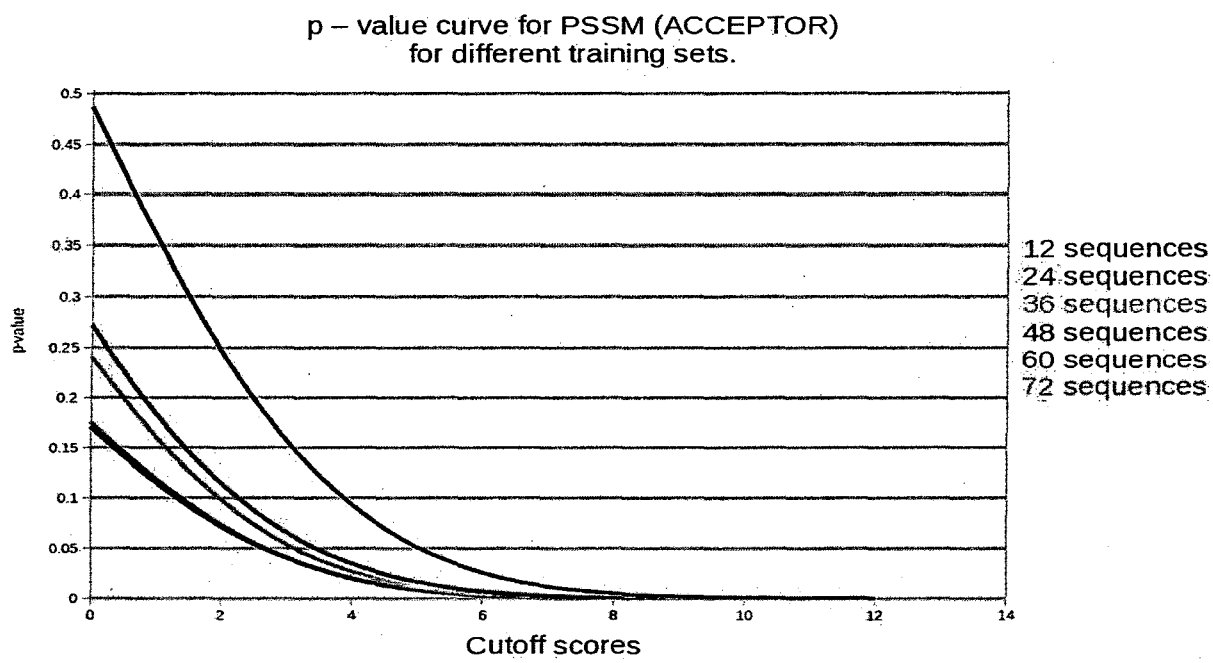

Figure 11

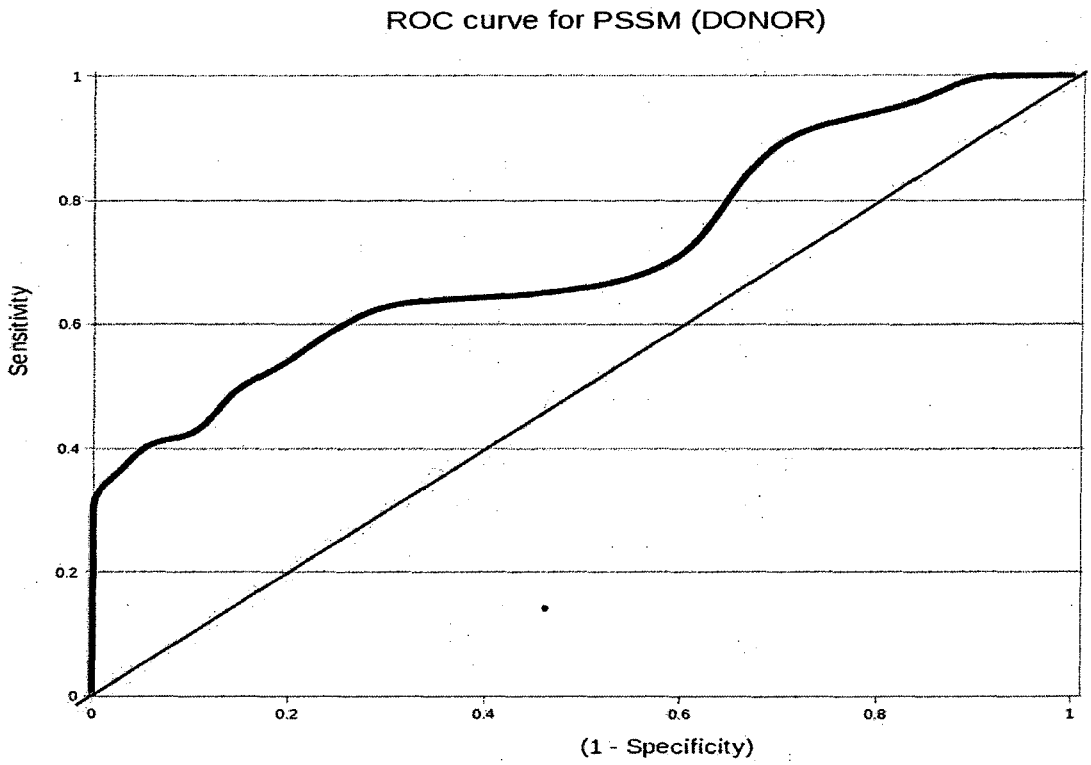

Figure 12

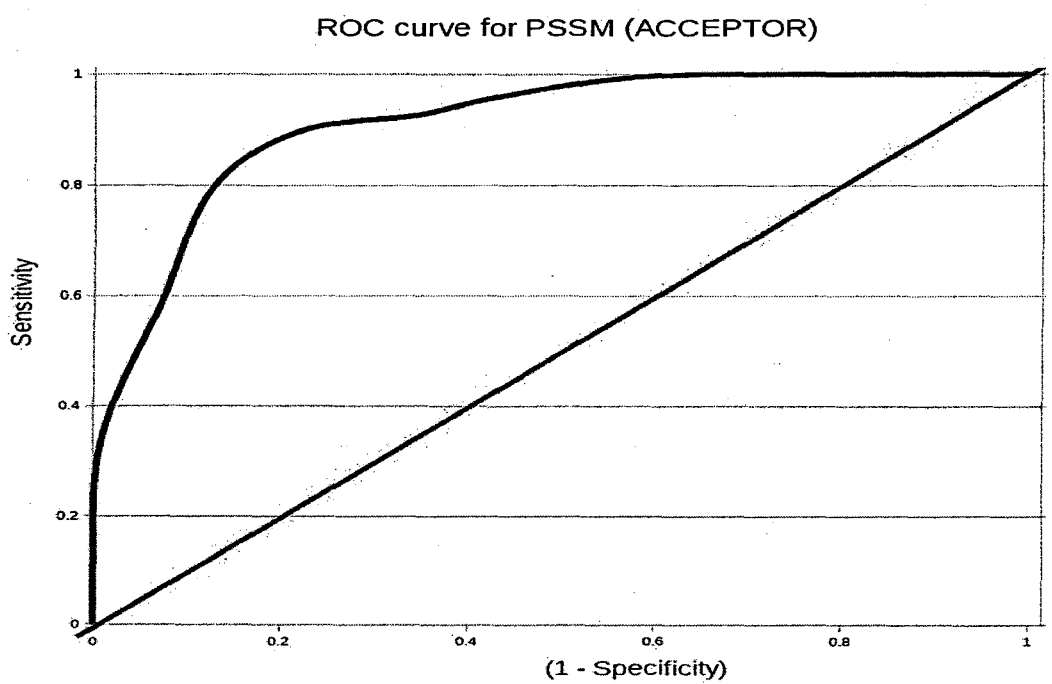

Figure 13

Markov 1st order model:

The sequences and multiple sequence alignments used for generating PSSM models were also used for training Markov models for splice DONOR and ACCEPTOR sites. Positive and negative datasets were used to obtained specificity and sensitivity curves. The p value curves for both DONOR and ACCEPTOR were also obtained as described previously. Specificity and sensitivity (68%) were not very high for the Markov 1st order model, trained with splice DONOR sequences at optimum cutoff score of 35, with p value 0.18. The p value analysis over a range of cutoffs showed that the cutoff score should be between 42 to 44 in order to achieve 5% level of significance. However, at this value of cutoff, sensitivity decreased to 42%. Markov 1st order model, trained on ACCEPTOR sequences displayed better discrimination power, with optimum sensitivity and specificity (81%) at cutoff score of 42 with p value 0.045. ROC curves also showed good performance of Markov 1st order model trained on ACCEPTOR sequences. AUC, Mann Whitney "U" for DONOR ROC was 91.345 and that was of ACCEPTOR was 103.45.

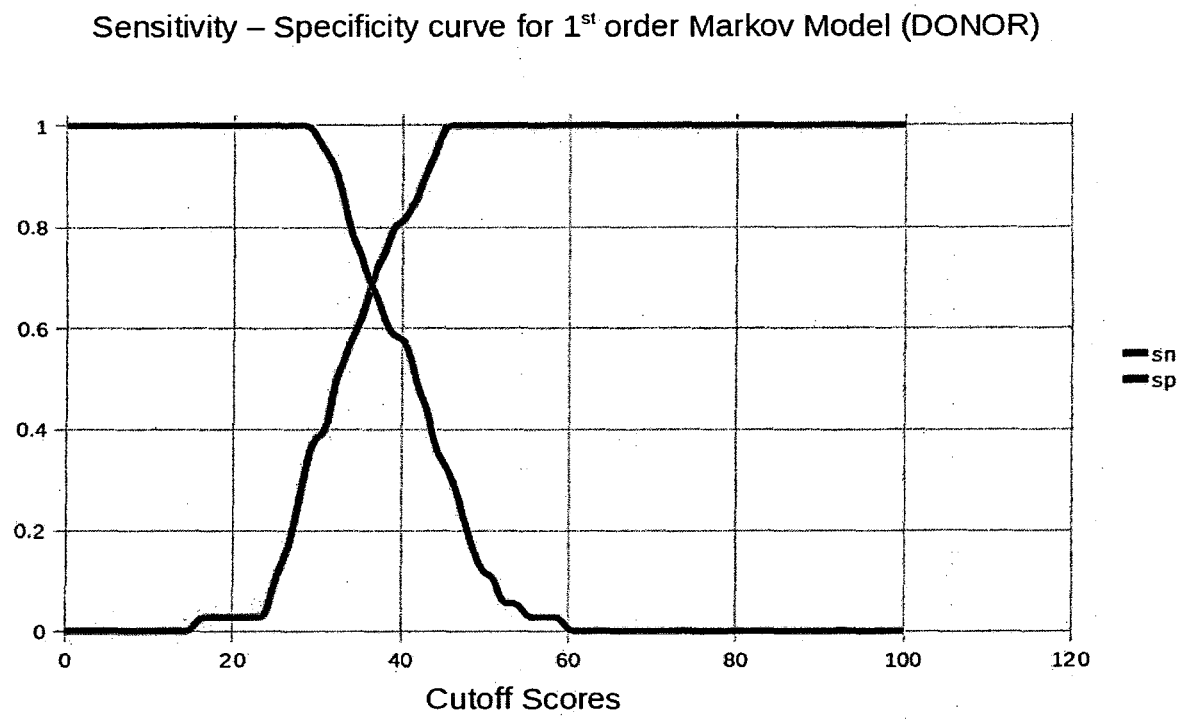

Figure 14

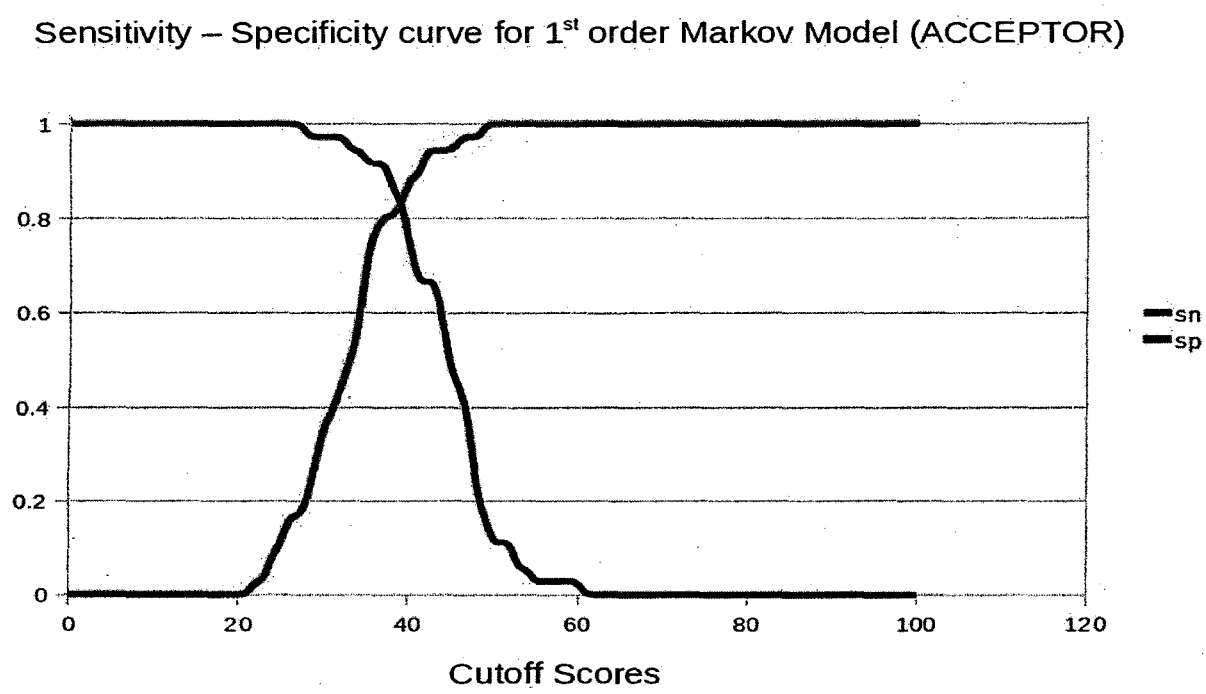

Figure 15

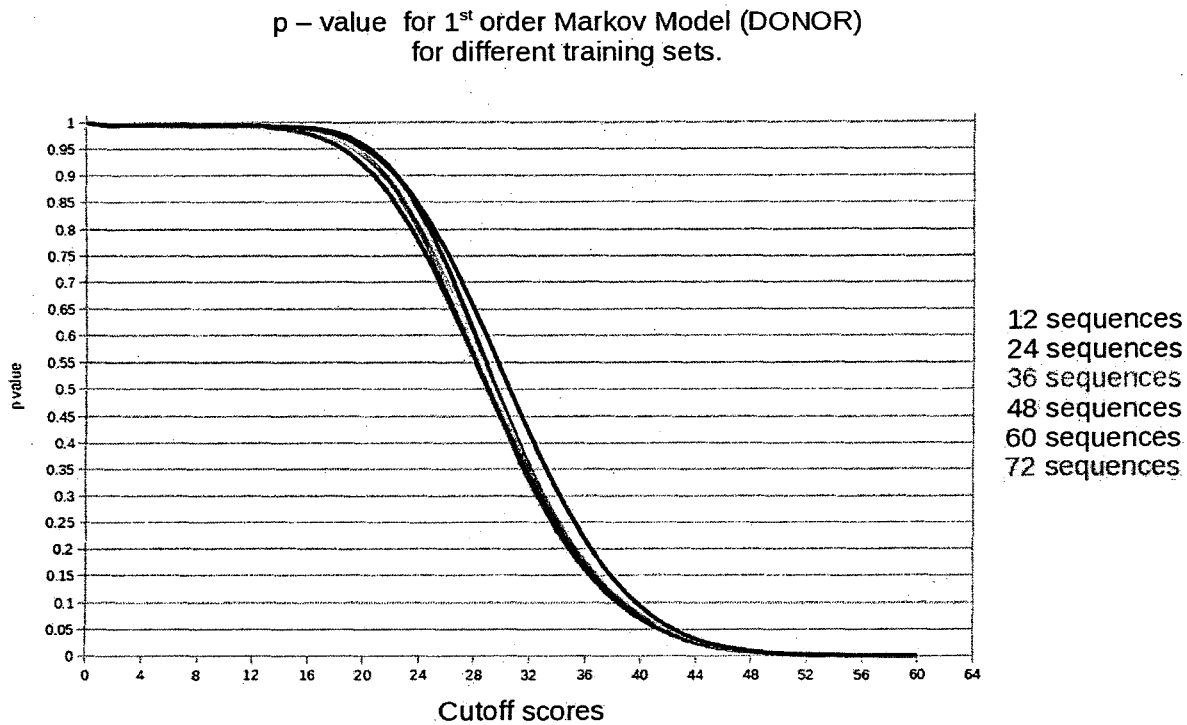

Figure 16

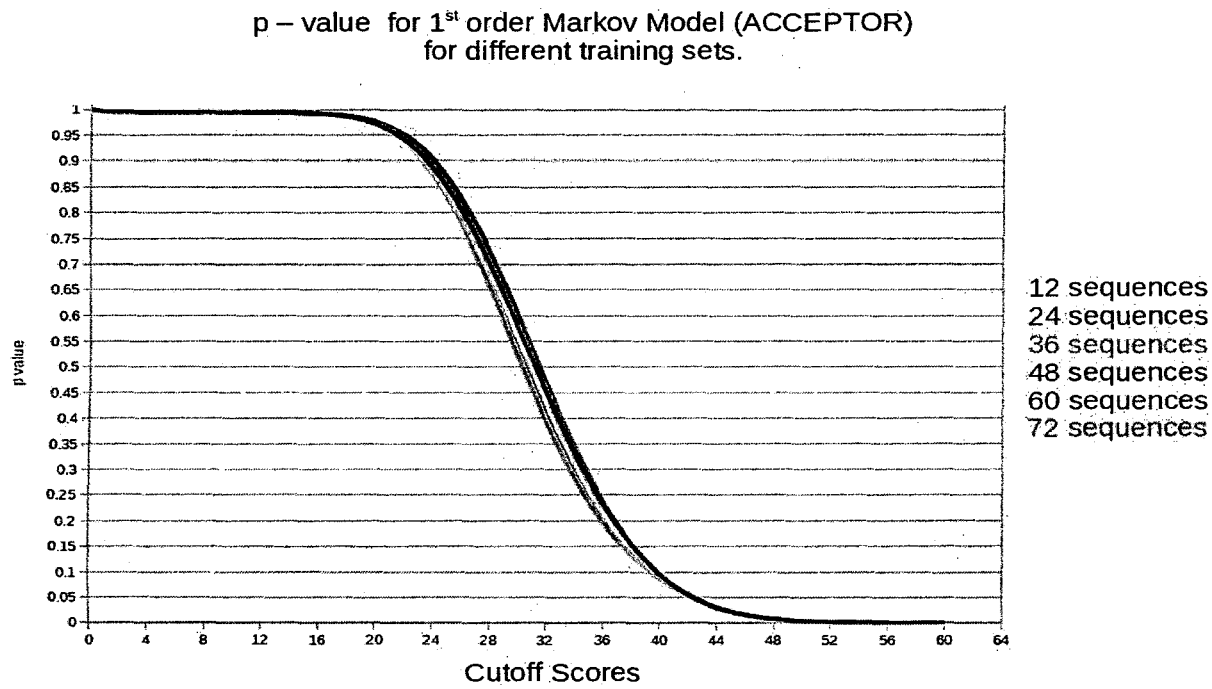

Figure 17

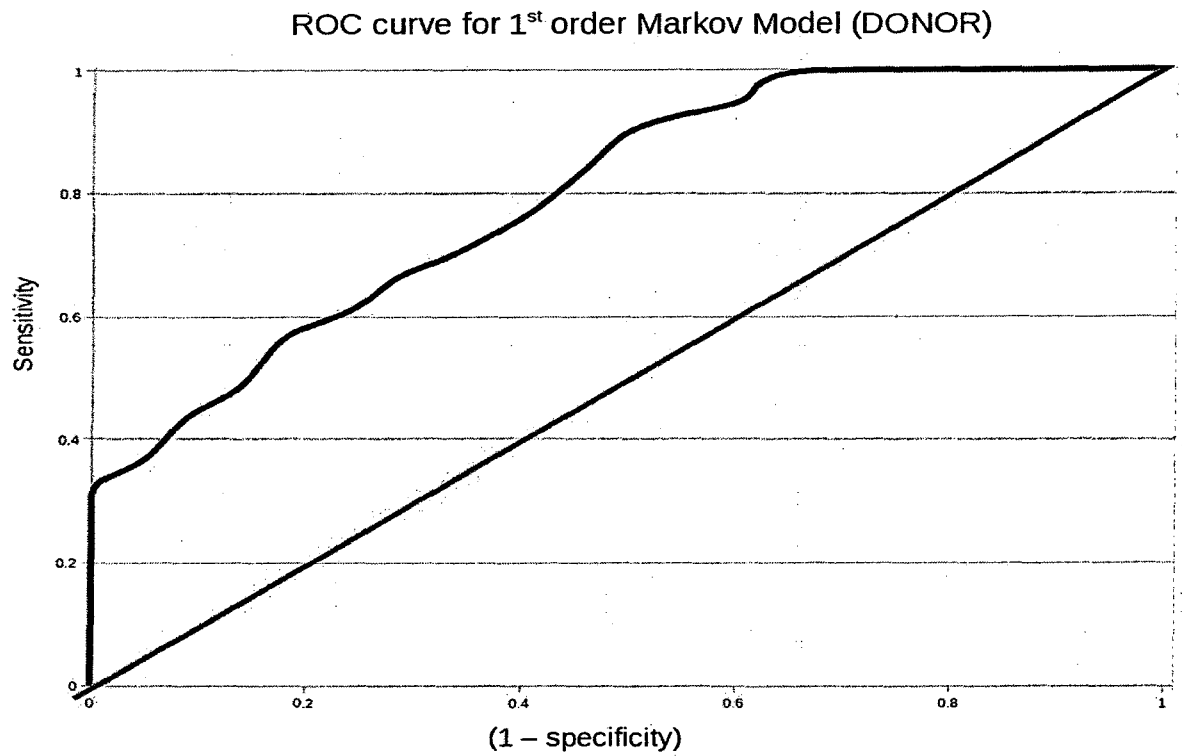

Figure 18

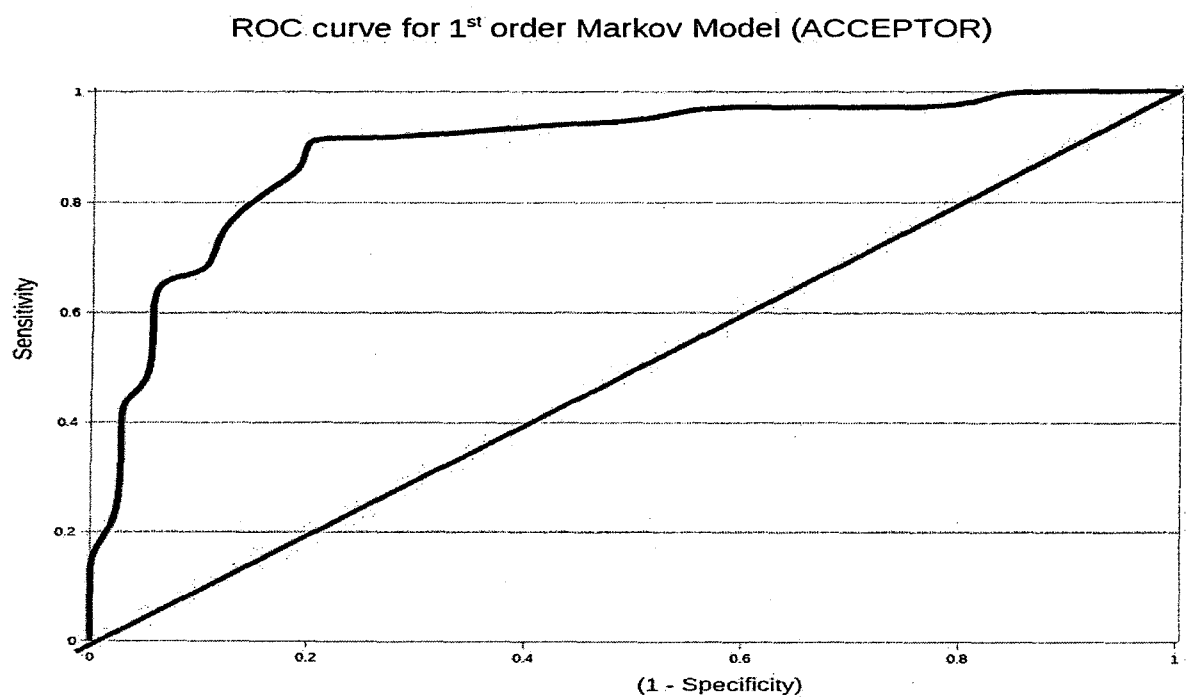

Figure 19

For second and third order Markov Models:

The second and third order models, trained on DONOR and ACCEPTOR sequences separately, were tested with positive test set and negative test set to obtain specificity, sensitivity and ROC curves. As the models are yet to be develop for sliding window operations and data structure is too computationally expensive for running different cutoffs using large sequences, the p value analysis both for second order and third order Markov models were not carried out. The second order Markov model appears to discriminate between non sites against true sites for both DONOR and ACCEPTORs. The optimum specificity and sensitivity were nearly 80% for both. The ROC curves for both DONOR and ACCEPTOR displayed good separations. Interestingly the third order Markov model dis not show high discrimination power with optimum specificity and sensitivity of about 40% for DONOR and about 50% for ACCEPTOR. This may be due to lack of enough sequences for training tetra nucleotide transitions in third order Markov model. AUC, Mann Whitney "U" for ROC obtained by second order Markov model was 92.63 for DONOR and 98.9 for ACCEPTOR that of for ROC obtain by third order Markov model was 13.11 for DONOR an 29.26 for ACCEPTOR.

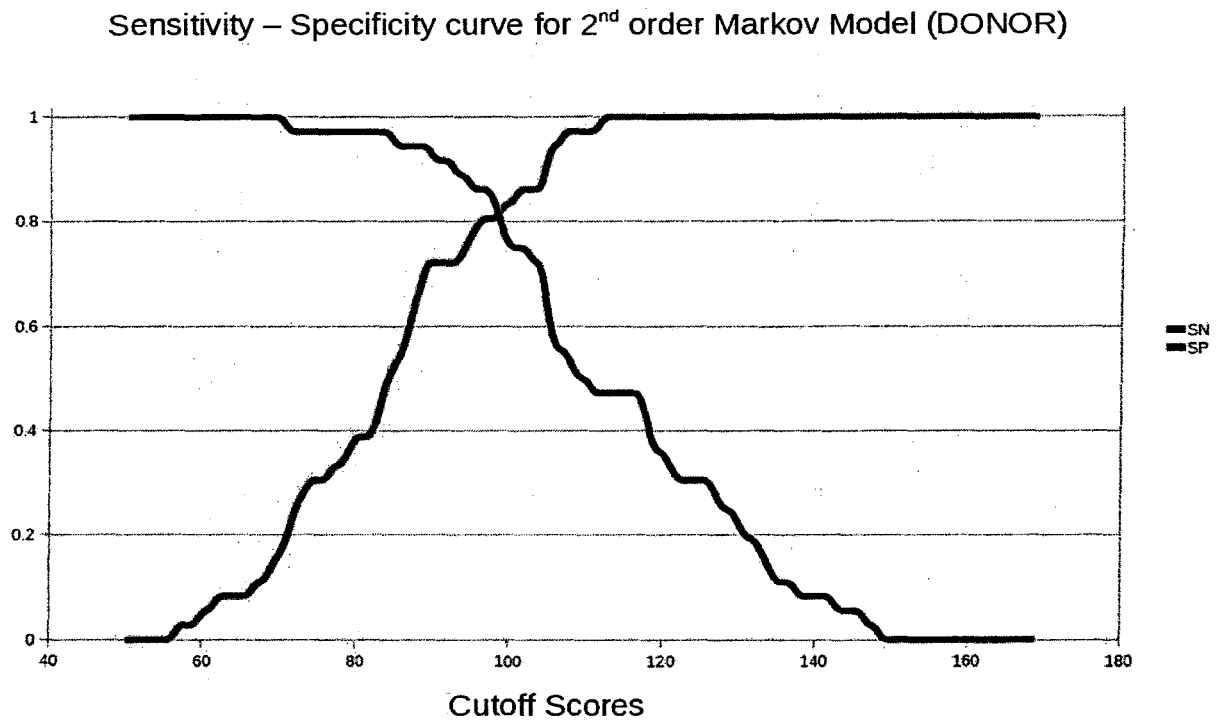

Figure 20

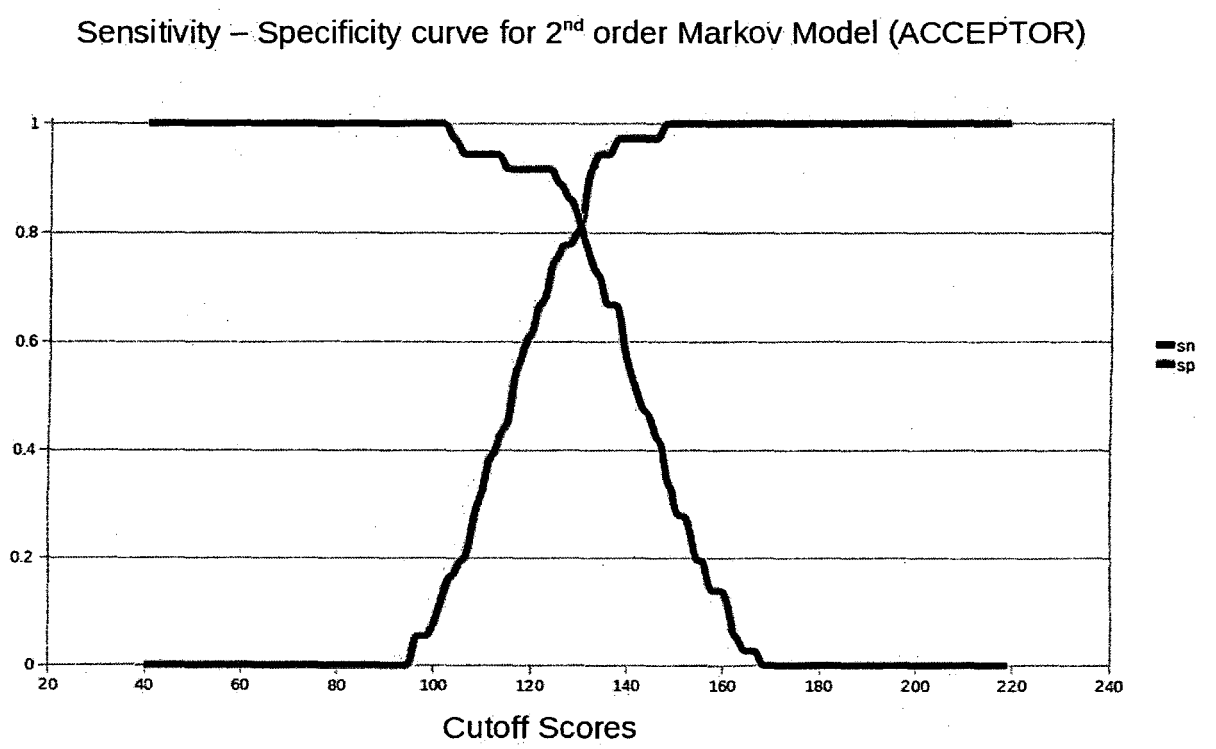

Figure 21

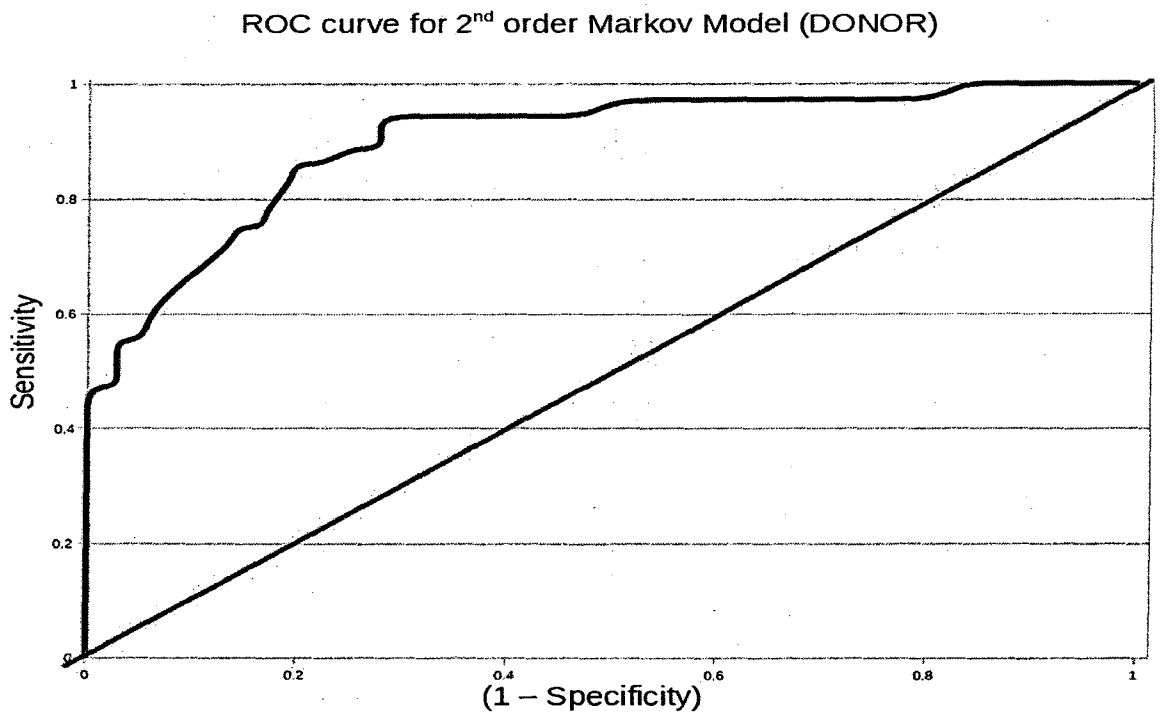

Figure 22

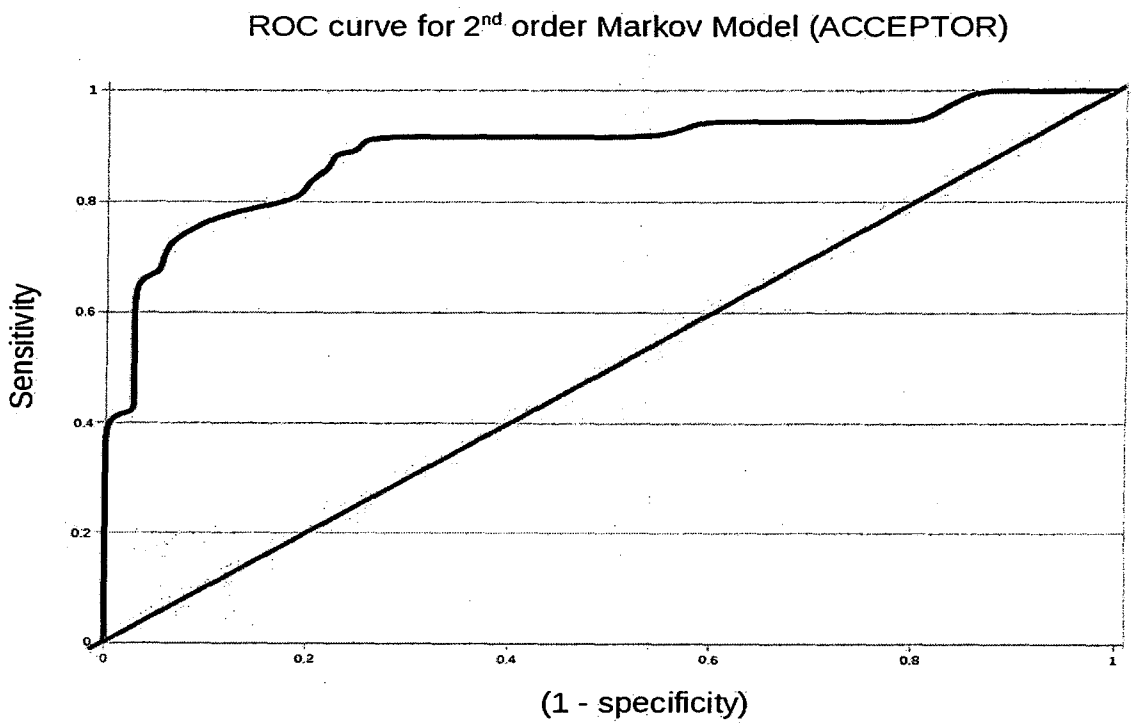

Figure 23

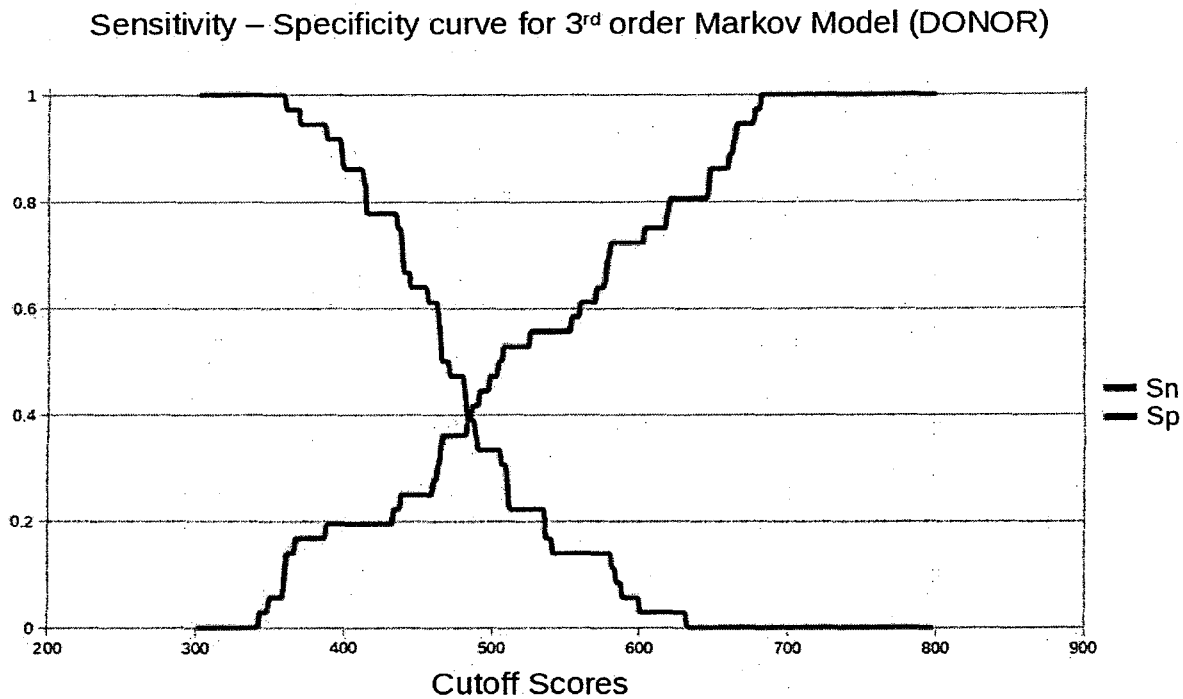

Figure 24

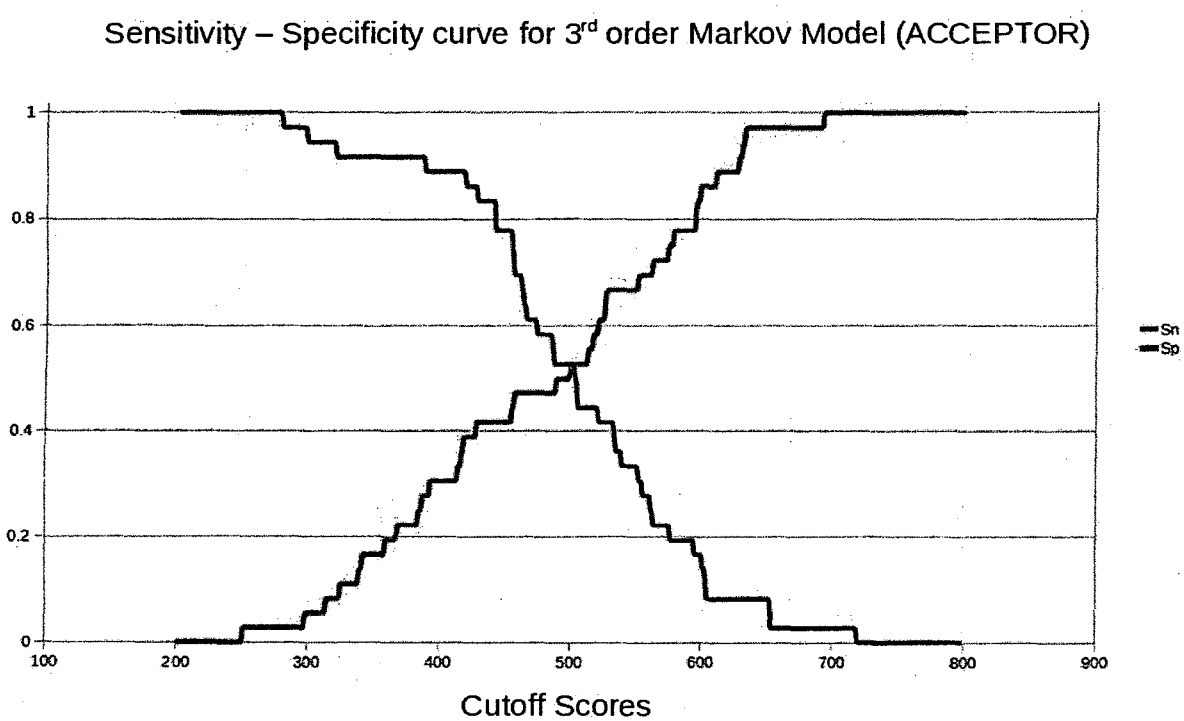

Figure 25

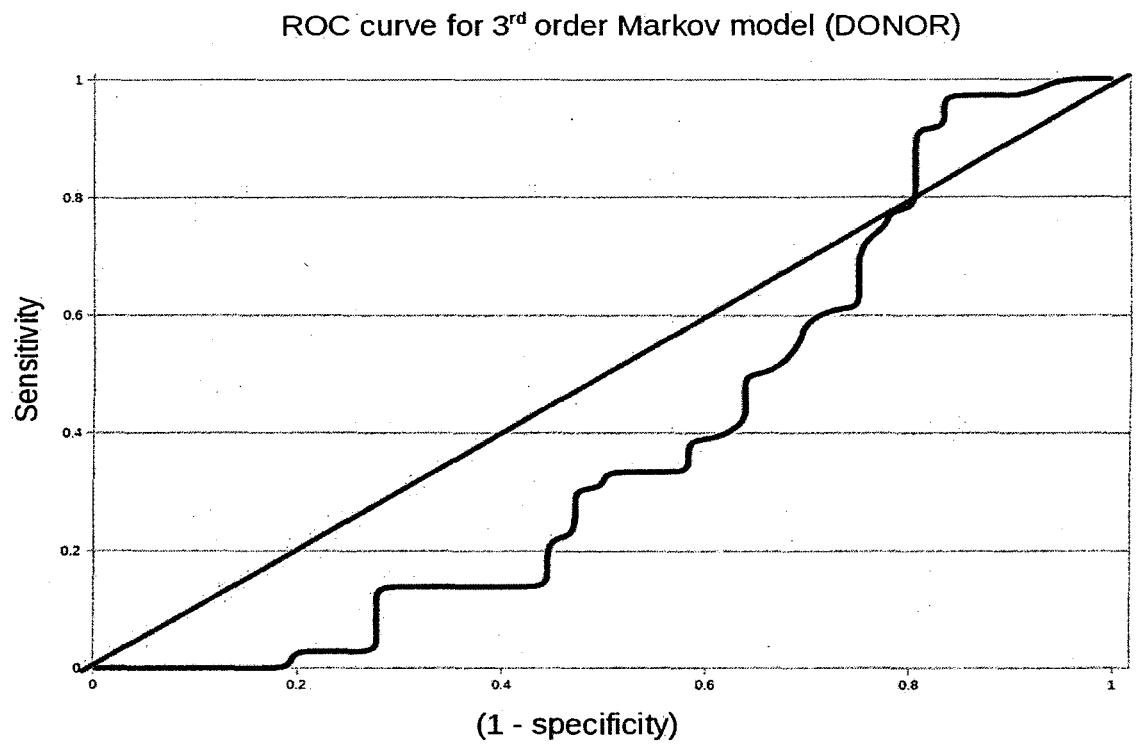

Figure 26

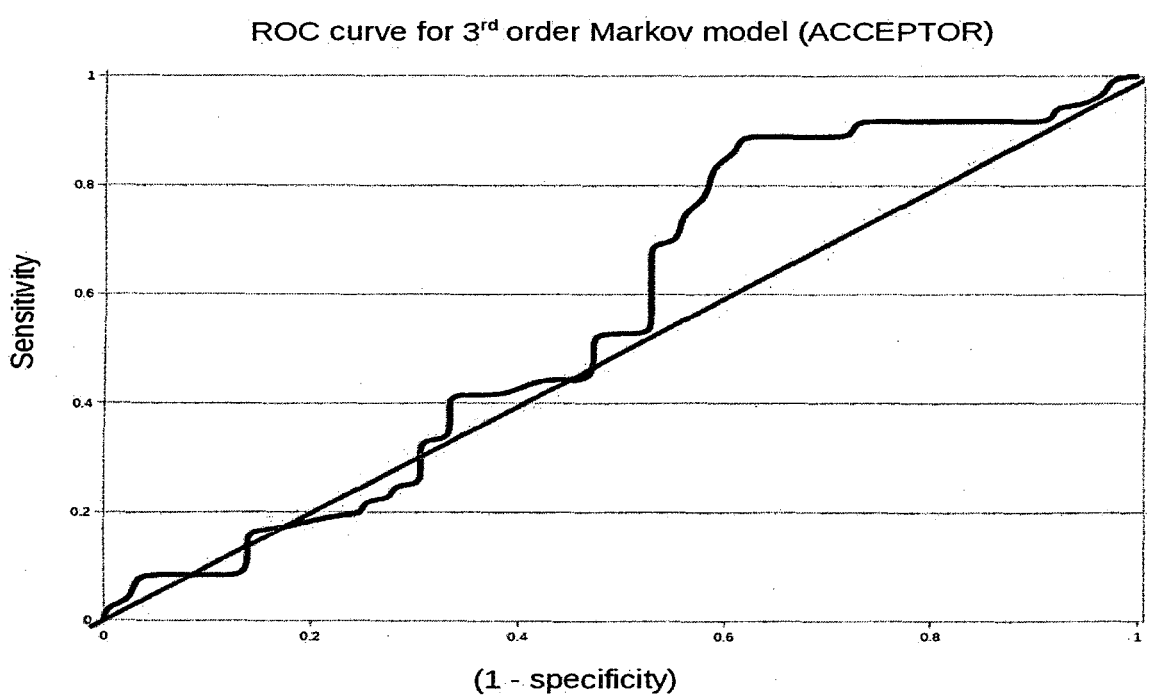

Figure 27

Integrated Use Of Stochastic models:

Cross-verification of NCBI annotation:

It is important to integrate results from all models so that a decision can be made regarding true or false donor or acceptor sites. For this, all annotated splicing junctions from *E. histolytica* genome were extracted and scores were given to each sequence one by one using stochastic models. There were four separate scores for each sequence generated by the four models. For integration two different approaches were used, one based on majority rule and a machine learning tool SVM was used as second approach. For simplicity a majority rule was used for prediction, that is, positive prediction was based on any three out of the four models predicting a sequence as positive and a negative decision is based on all the models predicting a sequence as negative. The results showed 480 "positive" annotations where both DONOR and ACCEPTORS were cross-verified as positive and in other hand we got 679 "Negative" annotation where DONOR and ACCEPTOR both were predicted as negative (see Appendix 1 for detailed results). The results clearly indicate that the current annotation of E. histolytica genomic sequences shows large scale error and that there is a need for a reannotation of the genome.

An SVM model was generated with 72 support vectors of dimension 29. Prewritten python script was used to decide parameters for classification. After 5-fold cross validation, 43.48% accuracy was obtained. Area Under Curve (AUC) of ROC was 0.435 for the DONORs. An accuracy of 33.91% with AUC 0.34 of ROC was observed for ACCEPTORs. One advantage of SVM model is its byproduct "Generated Support Vectors". These can be used for further statistical analysis. One can apply other parameters to classify them using different rules. For example one can do "principle component analysis" to shorten the number of dimension and then again do the SVM, or can apply Genetic Algorithm to choose parameters as well as kernel function to draw perfect hyperplane.

Decision Tree j48 analysis of integrated use of models also shows 39.35% accuracy. Detailed result is given in appendix 2.

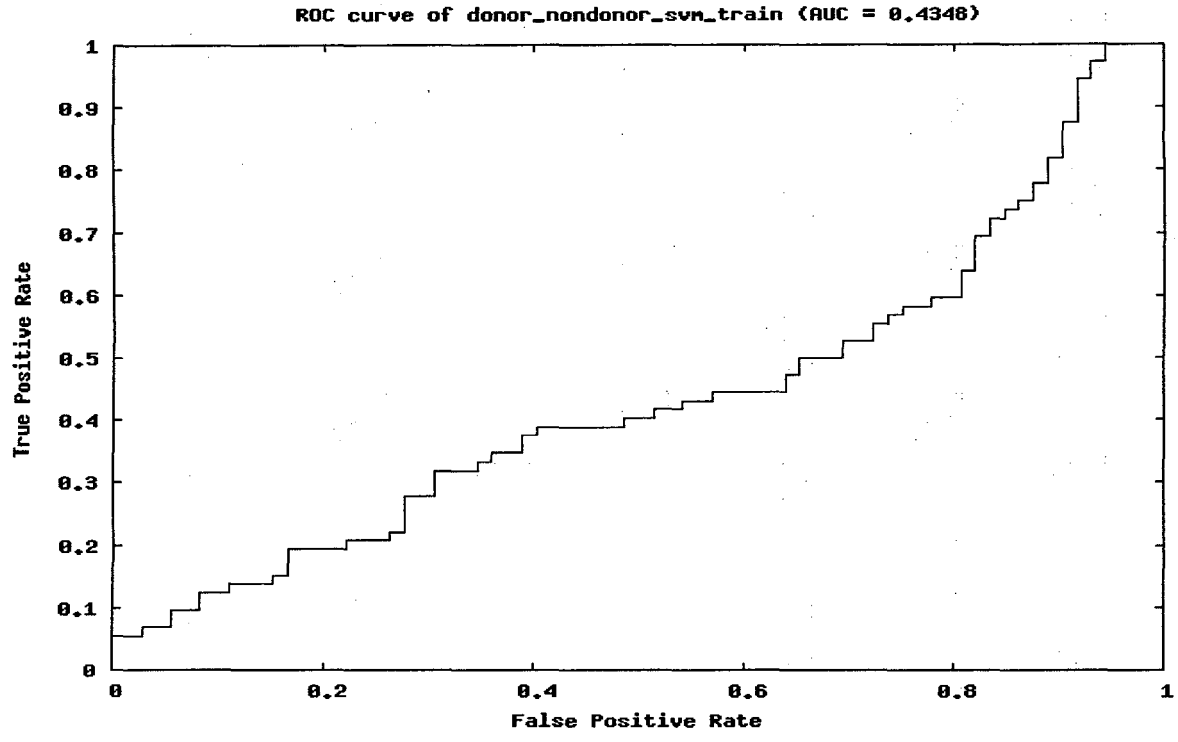

Figure 28

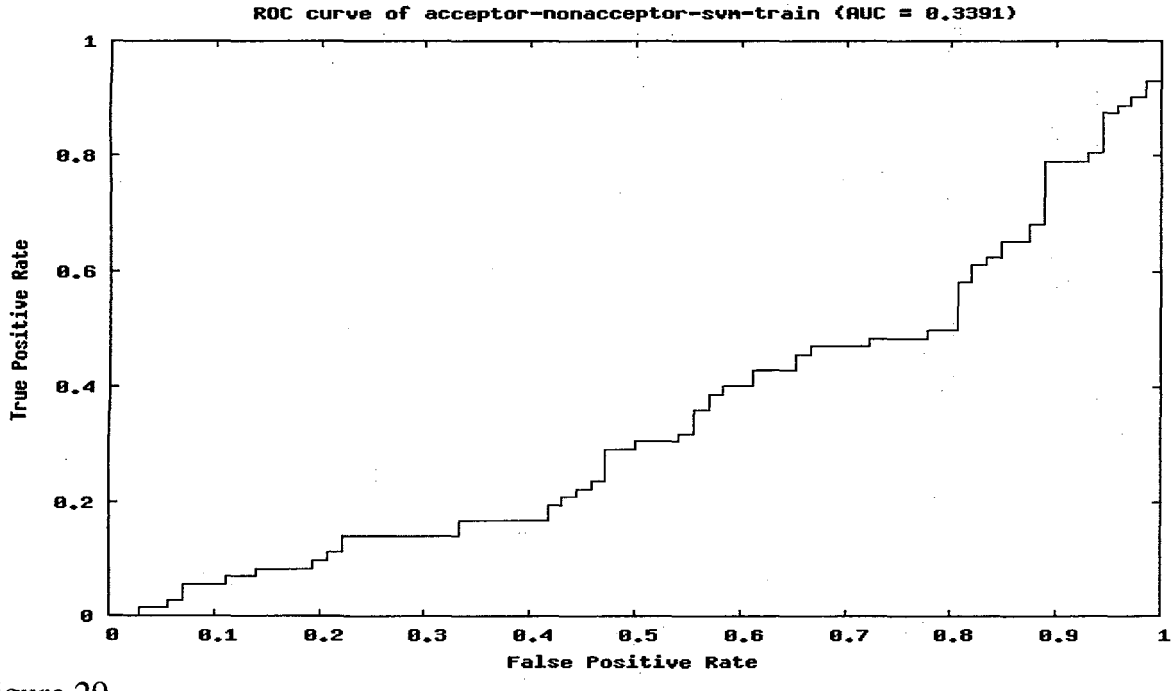

Figure 29

Though the stochastic models gave high prediction accuracy when used independently, on integration the results were found to be error prone. Following reasons may be put forward to explain such unexpected behavior.

Firstly, we are using scores of models as criteria for classification using classifiers such as SVM and j48, but it may be possible that the standard deviation of scores about "mean" is different for different models, thus rescaling of the scores are also not helping to overcome this difficulty. Secondly, selection of other attributes than the score in j48 and SVM were not based on prior statistical analysis like PCA, for example, we used dinucleotide and mono nucleotide frequencies along with the scores. Finally, we used pre written python script provided along with lib-SVM tool box, to optimize parameters and to choose kernel functions.

To overcome above obstacles in future we can write our own Genetic Algorithm code to optimize SVM parameters and to choose kernel function. PCA may be done to select attributes both for j48 and SVM. We can do relative normalization of model scores according to their distributions, in order to integrate them.

Stochastic Models And Transcription Start Sites:

For transcription start site prediction, only PSSM and Markov 1st order model were developed. Since only 16 sequences of shortlisted TSS were available for training the models the models generated are unlikely to capture the signals present in the motif. The 2nd order and 3rd Markov would not work as we had less number of training data. The 80 UTR sequences identified as 5'- UTRs using second approach (see page...) have not been analysed fully for identifying TSS. Trained PSSM and first order Markov model on 16 sequences displayed 52% and 60% optimum specificity and sensitivity respectively indicating that these models do not have high discrimination power. ROC was also not satisfactory. It is likely that addition of extra 80 sequences in the training set will improve prediction by the models. For PSSM ROC cure, AUC, Mann Whitney "U" was 17.34 and that of for Markov 1st order ROC was 26.3.

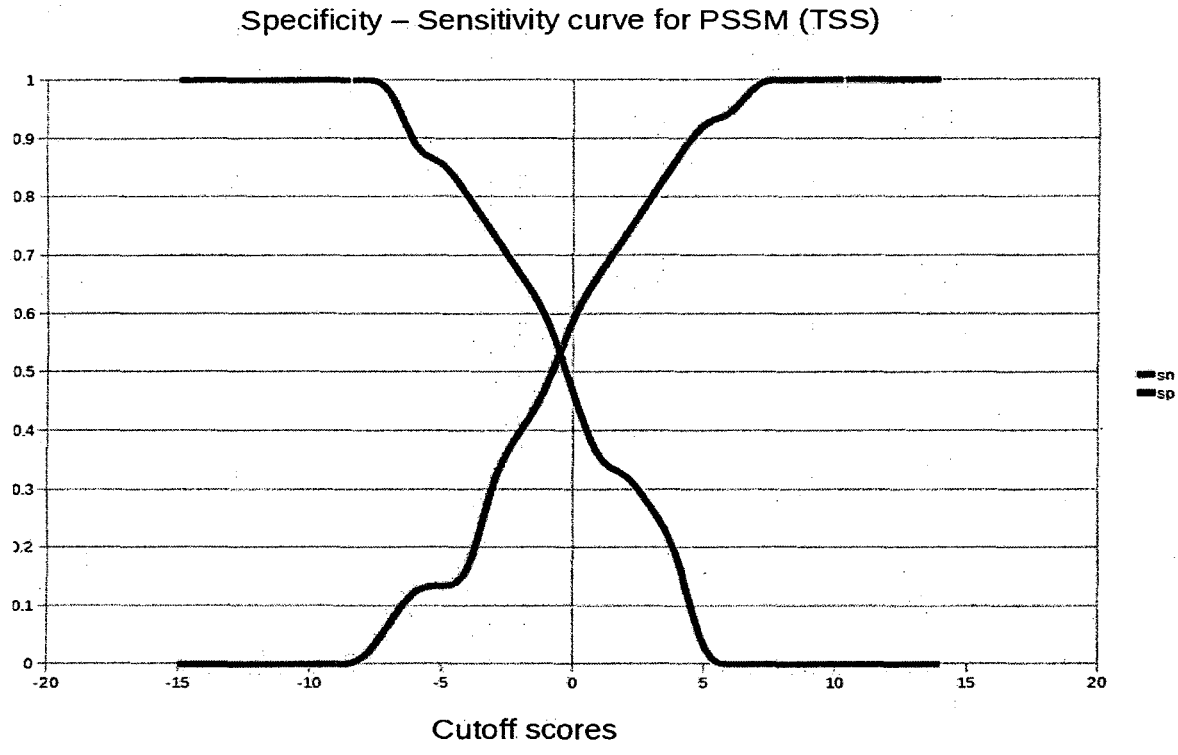

Figure 30

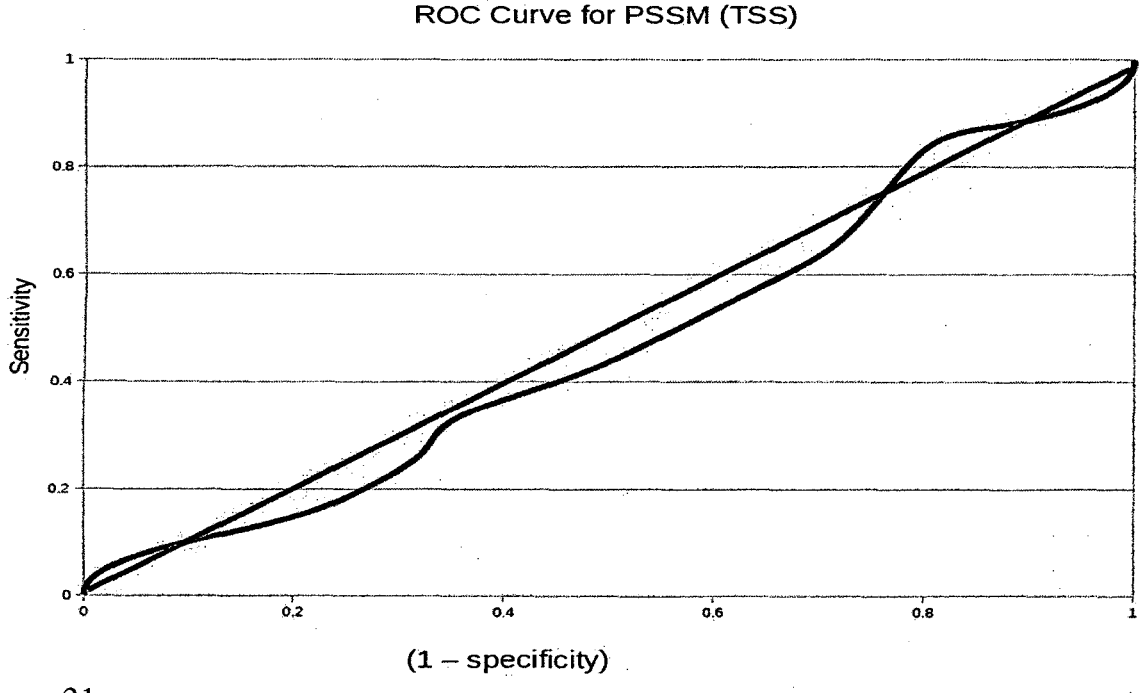

Figure 31

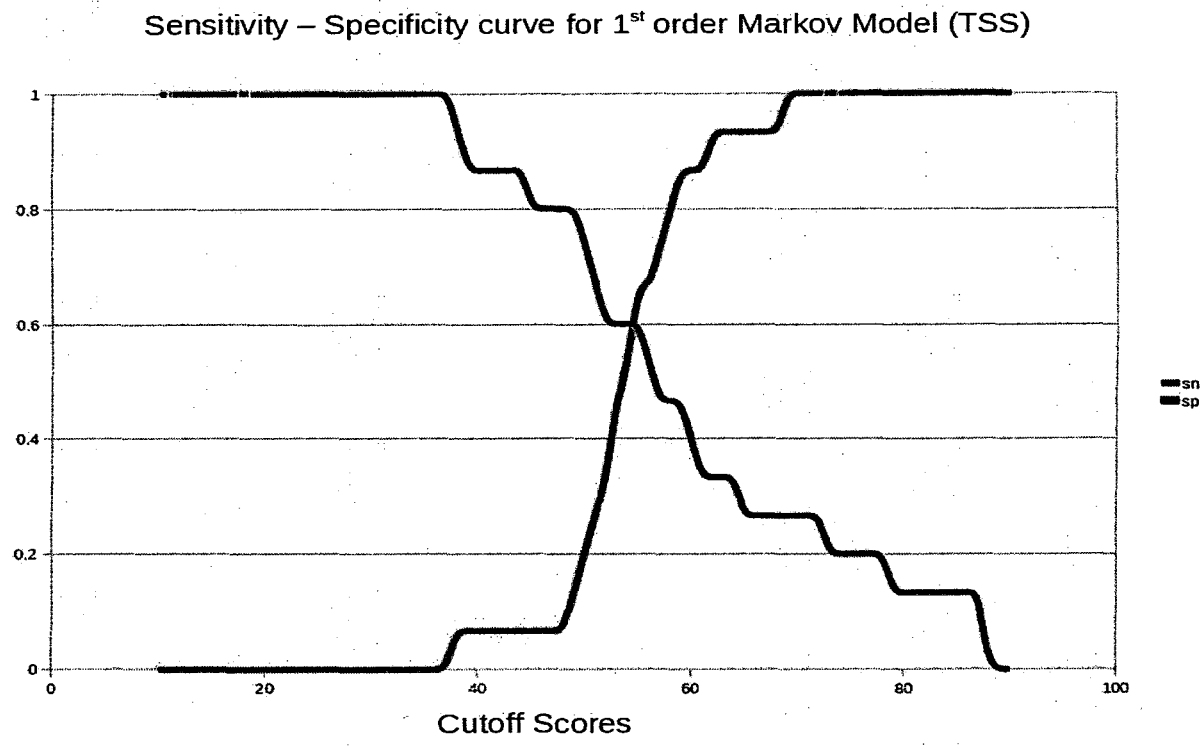

Figure 32

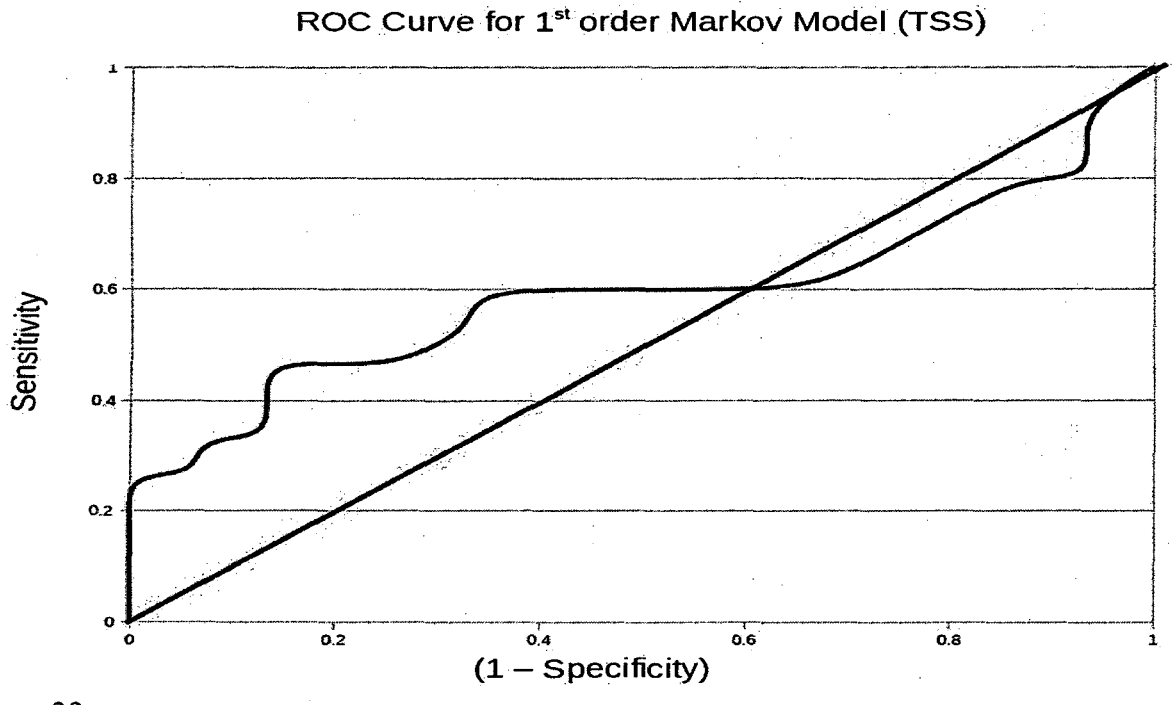

Figure 33

Conclusion:

In this study attempts have been made to develop suitable computational models for prediction of splice junctions (both donor and acceptors) and transcription start sites in the protozoan parasite *E. histolytica*. The current annotation displays errors in prediction of the above mentioned motifs and there is a need to develop newer methods for accurate annotation. One of the major problems for making better prediction is lack of enough sequences in *E. histolytica* where correct annotation is available for training any stochastic model. One of the first goals of this study was to make databases of accurate splice donor and acceptor sites and TSS. For this EST database was used. It did help to identify a few more sequences but probably not enough to train models properly.

A number of different models were used in order to remove bias that may be present in a model. In most of the cases suitable accuracy was observed for discrimination of splice acceptor sequences. But prediction of splice donor sites was quite inaccurate. However, an attempt to have a single decision based on either SVM or J48 classification using multiple models did not give higher accuracy suggesting that all the models may be seeing similar features. One of the major problems of predicting intron/exon or TSS in *E. histolytica* is that both intronic and UTR sequences are short (in fact much shorter than other organisms) and it is difficult to identify regions that may carry the appropriate signals. It is possible that the whole introns or UTRs may contain the signals though we have taken only a small part of these. Moreover, naive methods of scoring, for example, log-odds in stochastic models may work when a model is used independently, when integrated with other models, we have to consider rescaling of scores, incorporation of other parameters like nucleotide frequencies, biasness, information entropy etc.

Computational prediction of Transcription start site is difficult as computational models require experimentally verified true cases for the training the model. EST database proved to be useful for *in - silico* experimental determination of transcription start site. Finding full length cDNA among available ESTs is the key to identify 5' UTR. Overall it is necessary to do further work for developing truly highly accurate methods for signals in E. histolytica genome.

References:

Benjamin Lewin (2004); *Genes VIII*, Pearson, Chapter 2, The interrupted genes, pp 33-50.

Bertha Michel, Paul M. Lizardi, Alejandro Alagon, Mario Zurita (1995); Identification and analysis of the start site of ribosomal RNA transcription of Entamoeba histolytica, Molecular and Biochemical Parasitology 73 : 19-30.

Brendan Loftus, Iain Anderson, Rob Davies, U. Cecilia M. Alsmark,John Samuelson, Paolo Amedeo, Paola Roncaglia, Matt Berriman,Robert P. Hirt, Barbara J. Mann, Tomo Nozaki, Bernard Suh, Mihai Pop, Michael Duchene, John Ackers, Egbert Tannich, Matthias Leippe, Margit Hofer, Iris Bruchhaus, Ute Willhoeft, Alok Bhattacharya, Tracey Chillingworth, Carol Churcher, Zahra Hance, Barbara Harris, David Harris, Kay Jagels, Sharon Moule, Karen Mungall, Doug Ormond, Rob Squares, Sally Whitehead, Michael A. Quail, Ester Rabbinowitsch, Halina Norbertczak, Claire Price, Zheng Wang, Nancy Guille´n, Carol Gilchrist, Suzanne E. Stroup, Sudha Bhattacharya, Anuradha Lohia, Peter G. Foster, Thomas Sicheritz-Ponten, Christian Weber, Upinder Singh, Chandrama Mukherjee, Najib M. El-Sayed, William A. Petri Jr, C. Graham Clark, T. Martin Embley, Bart Barrel, Claire M. Fraser & Neil Hall, (2005); The genome of the protist parasite *Entamoeba histolytica*, *Nature*, 433:865-868.

Carol A. Gilchrist, Barbara J. Mann, AND William A. Petri, Jr., (1998); Control of Ferredoxin and Gal/GalNAc Lectin Gene Expression in *Entamoeba histolytica* by a cis-Acting DNA Sequence, *Infection and immunity*, 66/5 : 2383-2386

Catherine MatheÂ , Marie-France Sagot, Thomas Schiex and Pierre RouzeÂ, (2002); Current methods of gene prediction, their strengths and weaknesses, *Nucleic Acids Research*, 30/19: 4103-4117.

Chrissostomos Lioutas, Egbert Tannich, (1995); Transcription of protein-coding genes in Entamoeba histolytica is insensitive to high concentrations of alpha-amanitin, *Molecular and Biochemical Parasitology* 73:259-261.

Christopher D. Huston, (2004); Parasite and host contributions to the pathogenesis of amebic colitis, *TRENDS in Parasitology*, 20 /1:23-25.

Gary D. Stormo, (2000); DNA Binding sites, representation and discovery, *Bioinformatics* 16/1: 16-23.

Giorgio Grillo, Flavio Licciulli, Sabino Liuni, Elisabetta Sbia and Graziano Pesole, (2003); PatSearch: a program for the detection of patterns and structural motifs in nucleotide sequences,

Nucleic Acid Reasearch, 31/13 : 3608-3612.

I. Ben-Gal, A. Shani, A. Gohr, J. Grau, S. Arviv, A. Shmilovici, S. Posch, and I. Grosse (2005); Identification of transcription factor binding sites with variable-order Bayesian networks, *Bioinformatics* 21/11 : 2657-2666.

Jason A. Hackney, Gretchen M. Ehrenkaufer and Upinder Singh, (2007); Identification of putative transcriptional regulatory networks in Entamoeba histolytica using Bayesian inference, *Nucleic Acids Research* 35/7 : 2141-2152.

Jay E. Purdy, Lana T. Pho, Barbara J. Mann, William A. Petri Jr, (1996); Upstream regulatory elements controlling expression of the Entamoeba histolytica lectin, *Molecular and Biochemical Parasitology* 78 : 91-103.

John P Ackers and David Mirelman, (2006); Progress in research on Entamoeba histolytica pathogenesis, *Current Opinion in Microbiology*, 9:367–373.

Jonathan E. Allen, Mihaela Pertea, and Steven L. Salzberg, (2004); Computational Gene Prediction Using Multiple Sources of Evidence, *Genome Res* 14: 142-148.

Lutz Hamann, Heidrun BuB, Egbert Tannich, (1997); Tetracycline-controlled gene expression in Entamoeba histolytica, *Molecular and Biochemical Parasitology* 84 : 83-91.

Manuel Irimia, David Penny and Scott W. Roy (2007); Coevolution of genomic intron number and splice sites,*TRENDS in Genetics* 23/7:321-325.

Michael R Brent and Roderic Guigo, (2004); Recent advances in gene structure prediction, *Current Opinion in Structural Biology,* 14:264–272.

Olivos-Garc´ıa, A., Saavedra, E., Ramos-Mart´ınez, E., Nequiz, M., P´erez-Tamayo, R.,(2008); Molecular nature of virulence in Entamoeba histolytica, *Infection, Genetics and Evolution, doi:10.1016/j.meegid.2009.04.005.*

Scott William Roy and David Penny, (2007); Intron length distributions and gene prediction, *Nucleic acid research* 35/14:4737-4742.

Scott William Roy, Manuel Irimia, and David Penny, (2006); Very Little Intron Gain in *Entamoeba histolytica* Genes Laterally Transferred from Prokaryotes, *Mol. Biol. Evol.* 23/10:1824–1827.

Simon E. Cawley, Anthony I. Wirth, Terence P. Speed (2001); Phat – a gene finding program for Plasmodium falciparum *Molecular & Biochemical Parasitology* 118:167-174.

Susmita Mitra and Tinku Acharya, (2003); *Data Mining*, John Wiley & Sons.

Todd M. Lowe and Sean R. Eddy,(1997); tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence, *Nucleic acid research*, 25/5: 955-964.

Upinder Singh, Carol A. Gilchrist, Joanna M. Schaenman, Joshua B. Rogers, Joel W. Hockensmith, Barbara J. Mann, William A. Petri Jr, (2002); Context-dependent roles of the Entamoeba histolytica core promoter element GAAC in transcriptional activation and protein complex assembly, *Molecular & Biochemical Parasitology* 120 : 107–116.

Upinder Singh, Joshua B. Rogers, (1998); The Novel Core Promoter Element GAAC in the hgl5 Gene of Entamoeba histolytica Is Able to Direct a Transcription Start Site Independent of TATA or Initiator Regions, *The Journal Of Biological Chemistry* 273/34 : 21663-21668.

W. H. Majoros, M. Pertea and S. L. Salzberg,(2004); TigrScan and GlimmerHMM: two open source *ab initio* eukaryotic gene-finders, *Bioinformatics* 20/16:2878-2879.

Appendix 1

Stochastic models predictions: re-scanned NCBI Annotations

Predicted as "TRUE"

>DS571145|intron|56|join(<298420..299312,299368..299572>)
>DS571145|intron|55|join(<417175..417320,417375..417791>)
>DS571145|intron|52|join(<113382..113501,113553..113979>)
>DS571145|intron|complement|92|join(<9060..9961,10053..10191>)
>DS571145|intron|complement|49|join(<119871..119993,120042..121812>)
>DS571145|intron|52|join(<452230..452424,452476..452886>)
>DS571145|intron|complement|48|join(<424230..425735,425783..425913>)
>DS571145|intron|complement|45|join(<33005..34068,34113..34200>)
>DS571145|intron|52|join(<34475..34582,34634..34762>)
>DS571145|intron|56|join(<328491..328787,328843..329076>)
>DS571145|intron|64|join(<396167..396426,396490..397162>)
>DS571146|gene|581|(<142321..142902>)
>DS571146|intron|64|join(<143308..143411,143475..144029>)
>DS571146|intron|66|join(<143065..143242,143308..143411>)
>DS571146|intron|complement|46|join(<17024..17359,17405..18085>)
>DS571146|intron|complement|51|join(<28829..29233,29284..29446>)
>DS571146|intron|complement|150|join(<35836..36654,36804..36887>)
>DS571146|intron|89|join(<184663..185031,185120..185800>)
>DS571147|intron|54|join(<135571..135729,135783..136922>)
>DS571147|intron|complement|51|join(<3888..4034,4085..4725>)
>DS571147|intron|50|join(<109599..109952,110002..110172>)
>DS571147|intron|98|join(<58501..58795,58893..61465>)
>DS571147|intron|132|join(<23972..24101,24233..24471>)
>DS571147|intron|complement|54|join(<101384..102036,102090..102102>)
>DS571148|intron|79|join(<124642..124920,124999..125487>)
>DS571148|intron|51|join(<158072..158149,158200..160780>)
>DS571148|intron|78|join(<44809..45063,45141..46022>)
>DS571148|intron|57|join(<167783..167936,167993..169035>)
>DS571148|intron|70|join(<127775..128018,128088..128347>)
>DS571148|intron|complement|72|join(<66522..67075,67147..67420>)
>DS571149|intron|complement|49|join(<66782..66975,67024..68332>)
>DS571149|intron|44|join(<126841..127981,128025..128194>)
>DS571149|intron|complement|56|join(<139148..139265,139321..139707>)
>DS571149|intron|55|join(<167026..168457,168512..168621>)
>DS571150|intron|78|join(<6614..6621,6699..7418>)
>DS571150|intron|complement|180|join(<138695..139405,139585..141480>)
>DS571150|intron|48|join(<103952..104316,104364..106296>)
>DS571150|intron|69|join(<146442..148286,148355..148624>)
>DS571150|intron|50|join(<114592..115055,115105..116140>)
>DS571150|intron|77|join(<31388..31684,31761..41504>)
>DS571150|intron|62|join(<157490..158901,158963..159206>)
>DS571150|intron|66|join(<54194..54615,54681..56451>)
>DS571151|intron|63|join(<131076..131353,131416..132525>)
>DS571151|intron|49|join(<72717..72737,72786..73255>)
>DS571152|intron|54|join(<72228..72509,72563..75089>)
>DS571152|intron|complement|65|join(<27650..29097,29162..29282>)
>DS571152|intron|53|join(<34935..35395,35448..36030>)
>DS571152|intron|64|join(<57170..57371,57435..59170>)
>DS571154|intron|complement|137|join(<20607..20826,20963..21376>)
>DS571154|intron|85|join(<67213..67471,67556..68403>)
>DS571154|intron|complement|53|join(<83680..83709,83762..83855>)
>DS571154|intron|complement|113|join(<120899..121673,121786..122282>)

>DS571154|intron|complement|152|join(<137545..138678,138830..139747>)
>DS571155|intron|51|join(<130495..131841,131892..131960>)
>DS571155|intron|complement|54|join(<107255..107499,107553..107937>)
>DS571156|intron|45|join(<148209..148340,148385..148828>)
>DS571156|intron|complement|62|join(<10698..10919,10981..11586>)
>DS571156|intron|51|join(<31656..32119,32170..32581>)
>DS571156|intron|52|join(<50915..51016,51068..51634>)
>DS571156|intron|complement|45|join(<56724..57365,57410..58456>)
>DS571156|intron|67|join(<62872..63067,63134..64029>)
>DS571156|intron|complement|93|join(<94141..94677,94770..95137>)
>DS571156|intron|91|join(<96160..98452,98543..99093>)
>DS571156|intron|60|join(<101729..101908,101968..102933>)
>DS571156|intron|59|join(<103518..103841,103900..105090>)
>DS571156|intron|complement|81|join(<105749..107199,107280..108999>)
>DS571157|intron|complement|59|join(<42313..42545,42604..45019>)
>DS571157|intron|complement|51|join(<116782..117077,117128..117313>)
>DS571157|intron|complement|137|join(<129454..131016,131153..131218>)
>DS571158|intron|complement|56|join(<32355..32481,32537..32611>)
>DS571158|intron|complement|53|join(<34016..34142,34195..34269>)
>DS571158|intron|complement|60|join(<61890..62252,62312..63091>)
>DS571159|intron|118|join(<117379..118180,118298..118675>)
>DS571159|intron|48|join(<45274..45848,45896..46191>)
>DS571159|intron|47|join(<69952..70182,70229..70960>)
>DS571160|intron|49|join(<60984..61040,61089..61148>)
>DS571160|intron|44|join(<44401..45666,45710..45748>)
>DS571161|intron|complement|46|join(<23993..23999,24045..24200>)
>DS571161|intron|complement|54|join(<27443..27636,27690..27804>)
>DS571161|intron|complement|57|join(<28171..28326,28383..28555>)
>DS571161|intron|113|join(<91283..91428,91541..92009>)
>DS571161|intron|100|join(<37669..38191,38291..43545>)
>DS571162|intron|51|join(<26927..27047,27098..28018>)
>DS571162|intron|52|join(<107135..107249,107301..107407>)
>DS571162|intron|56|join(<5523..5584,5640..13158>)
>DS571162|intron|complement|53|join(<29756..30371,30424..31043>)
>DS571162|intron|complement|53|join(<36479..36640,36693..36767>)
>DS571162|intron|53|join(<24916..25261,25314..25828>)
>DS571162|intron|86|join(<91089..91508,91594..93345>)
>DS571163|intron|80|join(<7352..7565,7645..8027>)
>DS571163|intron|51|join(<18189..18497,18548..19492>)
>DS571164|intron|57|join(<21501..21615,21672..21724>)
>DS571164|intron|60|join(<104421..104553,104613..104731>)
>DS571164|intron|complement|52|join(<8664..9064,9116..9161>)
>DS571164|intron|116|join(<103785..103787,103903..104367>)
>DS571165|intron|60|join(<112512..113827,113887..114020>)
>DS571165|intron|55|join(<2088..2387,2442..2851>)
>DS571165|intron|50|join(<57098..57364,57414..58018>)
>DS571165|intron|87|join(<28883..29552,29639..29763>)
>DS571165|intron|complement|49|join(<80963..81177,81226..81290>)
>DS571165|intron|complement|88|join(<29838..30024,30112..31046>)
>DS571166|intron|46|join(<102469..102759,102805..103057>)
>DS571166|intron|complement|58|join(<3149..3365,3423..3466>)
>DS571166|intron|52|join(<44374..44896,44948..45144>)
>DS571167|intron|57|join(<19683..19694,19751..21559>)
>DS571167|intron|61|join(<44140..44664,44725..44835>)
>DS571167|intron|complement|55|join(<92306..92705,92760..92842>)
>DS571168|intron|complement|49|join(<95143..98593,98642..101253>)
>DS571169|intron|complement|55|join(<13535..13830,13885..14626>)
>DS571169|intron|45|join(<64376..64402,64447..67617>)
>DS571169|intron|complement|58|join(<83615..83774,83832..83878>)
>DS571170|intron|complement|51|join(<46488..46807,46858..48112>)
>DS571170|intron|complement|48|join(<43056..43798,43846..44131>)
>DS571171|intron|142|join(<41231..41250,41392..41623>)
>DS571171|intron|48|join(<4159..4301,4349..6305>)

>DS571172|intron|complement|53|join(<58634..59310,59363..59576>)
>DS571173|intron|71|join(<102741..103203,103274..103301>)
>DS571173|intron|complement|64|join(<9361..9916,9980..9993>)
>DS571173|intron|complement|54|join(<44201..44590,44644..45060>)
>DS571173|intron|76|join(<12692..13922,13998..14248>)
>DS571173|intron|complement|49|join(<70813..71162,71211..71448>)
>DS571173|intron|complement|80|join(<28091..28405,28485..28508>)
>DS571173|intron|complement|60|join(<83318..84086,84146..84266>)
>DS571173|intron|complement|57|join(<92365..92538,92595..93422>)
>DS571174|intron|55|join(<31999..32095,32150..32685>)
>DS571174|intron|complement|45|join(<8923..9195,9240..9620>)
>DS571174|intron|112|join(<93648..93950,94062..95042>)
>DS571174|intron|complement|57|join(<23383..23710,23767..24290>)
>DS571174|intron|complement|73|join(<25437..26376,26449..27119>)
>DS571175|intron|54|join(<70540..70770,70824..71240>)
>DS571175|intron|45|join(<7222..7353,7398..10721>)
>DS571175|intron|51|join(<50033..50454,50505..51068>)
>DS571177|intron|38|join(<36134..36265,36303..36487>)
>DS571177|intron|75|join(<64760..64864,64939..65062>)
>DS571178|intron|58|join(<27388..27573,27631..27786>)
>DS571178|intron|complement|54|join(<35627..35853,35907..36072>)
>DS571178|intron|complement|50|join(<38945..39022,39072..39193>)
>DS571179|intron|complement|50|join(<74168..74227,74277..74963>)
>DS571179|intron|47|join(<21735..21772,21819..22902>)
>DS571179|intron|complement|62|join(<88357..88511,88573..89281>)
>DS571179|intron|complement|51|join(<57141..59203,59254..59449>)
>DS571179|intron|56|join(<68350..68632,68688..68971>)
>DS571180|intron|50|join(<49841..50433,50483..50665>)
>DS571182|intron|complement|67|join(<25797..28005,28072..28214>)
>DS571182|intron|53|join(<70417..70740,70793..71392>)
>DS571183|intron|61|join(<19532..19542,19603..20203>)
>DS571183|intron|complement|56|join(<24419..24799,24855..24980>)
>DS571183|gene|641|(<70666..71307>)
>DS571184|intron|complement|54|join(<5288..6081,6135..6323>)
>DS571184|intron|106|join(<17113..17270,17376..19023>)
>DS571185|intron|complement|60|join(<72234..72262,72322..73114>)
>DS571186|intron|51|join(<2846..2935,2986..4387>)
>DS571186|intron|complement|171|join(<3632..4473,4644..5289>)
>DS571186|intron|58|join(<25909..26088,26146..27282>)
>DS571186|intron|55|join(<33505..34252,34307..34599>)
>DS571186|intron|57|join(<44673..45389,45446..45526>)
>DS571186|intron|complement|47|join(<51956..52202,52249..52493>)
>DS571187|gene|499|(<53572..54071>)
>DS571187|intron|51|join(<33211..33392,33443..39094>)
>DS571187|intron|132|join(<53572..53701,53833..54071>)
>DS571187|intron|complement|51|join(<71754..72256,72307..72551>)
>DS571187|intron|complement|45|join(<76415..77549,77594..77712>)
>DS571187|gene|662|(<86552..87214>)
>DS571187|intron|52|join(<86552..87044,87096..87214>)
>DS571188|intron|complement|49|join(<42238..43581,43630..44310>)
>DS571189|intron|complement|66|join(<25461..25545,25611..25701>)
>DS571189|intron|complement|64|join(<36927..42232,42296..42752>)
>DS571190|intron|58|join(<81372..81467,81525..81591>)
>DS571190|intron|51|join(<2110..2420,2471..2768>)
>DS571190|intron|complement|57|join(<73401..74534,74591..74925>)
>DS571190|intron|complement|55|join(<36347..36602,36657..36787>)
>DS571191|intron|complement|54|join(<16017..16233,16287..16897>)
>DS571191|intron|complement|91|join(<16287..16897,16988..17218>)
>DS571191|intron|complement|56|join(<33906..34142,34198..35778>)
>DS571191|intron|46|join(<19455..19736,19782..20546>)
>DS571191|intron|49|join(<38838..39131,39180..39413>)
>DS571191|intron|complement|93|join(<40494..40849,40942..41326>)
>DS571192|intron|137|join(<59714..59925,60062..60493>)

>DS571193|intron|complement|48|join(<23066..23312,23360..23740>)
>DS571194|intron|80|join(<7557..7811,7891..9003>)
>DS571194|intron|complement|61|join(<58797..59294,59355..59492>)
>DS571195|intron|complement|59|join(<9888..10025,10084..10602>)
>DS571195|intron|complement|87|join(<51005..51087,51174..51275>)
>DS571196|intron|53|join(<42819..43154,43207..43341>)
>DS571196|intron|complement|51|join(<48482..48843,48894..49239>)
>DS571197|gene|523|(<42..565>)
>DS571197|intron|59|join(<25319..25568,25627..26042>)
>DS571197|intron|complement|55|join(<44699..44775,44830..44966>)
>DS571198|intron|complement|42|join(<3379..4822,4864..5738>)
>DS571198|intron|complement|50|join(<8726..8821,8871..9227>)
>DS571198|intron|48|join(<27444..27846,27894..28267>)
>DS571198|intron|complement|50|join(<49067..51250,51300..52235>)
>DS571198|intron|complement|52|join(<58997..60897,60949..63589>)
>DS571199|intron|88|join(<20911..21263,21351..21774>)
>DS571200|intron|complement|52|join(<51551..51870,51922..52012>)
>DS571200|intron|complement|61|join(<63596..63880,63941..64774>)
>DS571202|intron|51|join(<14081..14321,14372..14656>)
>DS571202|intron|47|join(<13889..14034,14081..14321>)
>DS571202|intron|56|join(<20856..20908,20964..23079>)
>DS571202|intron|64|join(<62922..63610,63674..63743>)
>DS571203|intron|complement|61|join(<4839..5520,5581..5739>)
>DS571204|intron|complement|48|join(<7208..7510,7558..7875>)
>DS571204|intron|complement|57|join(<36361..37095,37152..37271>)
>DS571204|intron|57|join(<9920..10062,10119..10800>)
>DS571204|intron|48|join(<21301..21374,21422..22397>)
>DS571204|intron|66|join(<27754..27975,28041..28181>)
>DS571205|intron|complement|54|join(<33410..33569,33623..33730>)
>DS571205|intron|complement|49|join(<44240..44524,44573..45214>)
>DS571206|intron|59|join(<9877..10080,10139..10333>)
>DS571206|intron|53|join(<10588..10781,10834..10924>)
>DS571206|intron|complement|68|join(<20151..20229,20297..21084>)
>DS571207|intron|48|join(<56533..56870,56918..57140>)
>DS571208|intron|complement|53|join(<5151..5401,5454..6537>)
>DS571208|intron|49|join(<42085..42480,42529..42822>)
>DS571209|intron|complement|61|join(<22615..22851,22912..23406>)
>DS571209|intron|complement|60|join(<46417..46762,46822..47875>)
>DS571210|intron|complement|57|join(<38341..38456,38513..38549>)
>DS571210|intron|complement|111|join(<58738..59129,59240..59486>)
>DS571211|intron|complement|79|join(<6773..9805,9884..9955>)
>DS571211|intron|complement|60|join(<40355..40690,40750..41152>)
>DS571213|intron|46|join(<13876..13899,13945..14736>)
>DS571213|intron|complement|58|join(<13443..13529,13587..13594>)
>DS571213|intron|complement|60|join(<20116..20813,20873..21500>)
>DS571214|intron|72|join(<41254..41864,41936..42065>)
>DS571215|intron|complement|45|join(<12766..12903,12948..13001>)
>DS571215|intron|complement|50|join(<21792..21900,21950..22015>)
>DS571215|gene|613|(<25525..26138>)
>DS571215|intron|64|join(<26312..26336,26400..26614>)
>DS571216|intron|complement|62|join(<27013..30414,30476..30679>)
>DS571216|intron|complement|53|join(<38639..38754,38807..39036>)
>DS571218|intron|complement|86|join(<8755..9300,9386..9523>)
>DS571218|intron|complement|61|join(<9386..9523,9584..9703>)
>DS571218|intron|64|join(<11290..11546,11610..15321>)
>DS571218|intron|80|join(<16777..17133,17213..17245>)
>DS571219|intron|complement|206|join(<3918..4291,4497..4629>)
>DS571220|intron|325|join(<48971..49205,49530..49975>)
>DS571220|intron|complement|74|join(<50741..51257,51331..51431>)
>DS571222|intron|complement|54|join(<19637..19959,20013..21132>)
>DS571222|gene|626|(<23796..24422>)
>DS571222|intron|118|join(<23796..24041,24159..24422>)
>DS571222|intron|complement|83|join(<44757..45269,45352..46851>)

>DS571223|gene|602|(<14217..14819>)
>DS571223|intron|complement|53|join(<50289..50432,50485..50591>)
>DS571224|intron|58|join(<47723..47863,47921..49045>)
>DS571226|intron|94|join(<27366..27466,27560..27857>)
>DS571226|intron|54|join(<15076..15123,15177..15704>)
>DS571226|intron|complement|80|join(<41449..41472,41552..42051>)
>DS571227|intron|64|join(<12791..12797,12861..13004>)
>DS571227|intron|complement|57|join(<16220..16515,16572..16803>)
>DS571227|intron|83|join(<7644..7813,7896..8628>)
>DS571227|intron|complement|145|join(<51853..52587,52732..53112>)
>DS571228|intron|141|join(<41227..42082,42223..43088>)
>DS571228|intron|complement|47|join(<45819..46135,46182..46407>)
>DS571228|intron|57|join(<46586..46802,46859..48498>)
>DS571230|intron|complement|43|join(<9493..9666,9709..10995>)
>DS571230|intron|complement|55|join(<29566..29779,29834..29964>)
>DS571230|intron|50|join(<41706..43144,43194..43416>)
>DS571231|intron|85|join(<17033..17397,17482..18895>)
>DS571232|intron|complement|119|join(<39636..40261,40380..40858>)
>DS571233|intron|59|join(<4417..4554,4613..5164>)
>DS571233|intron|complement|47|join(<9372..9717,9764..10362>)
>DS571233|intron|complement|112|join(<37472..37720,37832..38042>)
>DS571233|intron|complement|47|join(<37832..38042,38089..38180>)
>DS571235|intron|complement|54|join(<51221..51294,51348..51495>)
>DS571236|intron|53|join(<21062..21368,21421..21497>)
>DS571237|intron|55|join(<11141..12046,12101..12190>)
>DS571237|intron|69|join(<21538..21657,21726..24365>)
>DS571238|intron|52|join(<16825..16956,17008..17405>)
>DS571238|intron|66|join(<20058..21450,21516..21691>)
>DS571238|intron|150|join(<11195..11786,11936..12708>)
>DS571240|intron|60|join(<19667..19808,19868..20283>)
>DS571241|intron|57|join(<31711..32364,32421..33653>)
>DS571242|intron|139|join(<45500..45664,45803..46029>)
>DS571242|intron|55|join(<16839..17030,17085..17465>)
>DS571244|intron|353|join(<16188..16232,16585..17709>)
>DS571244|intron|60|join(<38040..38442,38502..38761>)
>DS571244|intron|117|join(<45045..45364,45481..45820>)
>DS571245|intron|136|join(<5628..5851,5987..6740>)
>DS571245|intron|50|join(<10773..10910,10960..11163>)
>DS571246|gene|574|(<9656..10230>)
>DS571246|gene|690|(<23865..24555>)
>DS571246|intron|66|join(<41615..41920,41986..42240>)
>DS571248|intron|complement|161|join(<3593..3955,4116..9887>)
>DS571248|intron|complement|57|join(<11356..13811,13868..14243>)
>DS571249|intron|complement|58|join(<28029..35030,35088..35201>)
>DS571253|intron|complement|70|join(<7570..10571,10641..11111>)
>DS571253|intron|62|join(<40284..40553,40615..40890>)
>DS571254|intron|complement|59|join(<2022..2265,2324..5772>)
>DS571254|intron|complement|59|join(<2324..5772,5831..6145>)
>DS571254|intron|complement|55|join(<13265..13318,13373..13675>)
>DS571256|intron|60|join(<35798..35950,36010..36042>)
>DS571256|intron|54|join(<37556..38002,38056..42681>)
>DS571256|intron|complement|56|join(<43661..43916,43972..44102>)
>DS571256|intron|60|join(<16876..16972,17032..17144>)
>DS571256|intron|complement|55|join(<17334..17434,17489..18254>)
>DS571257|intron|complement|65|join(<14630..14863,14928..15002>)
>DS571257|intron|73|join(<41592..42460,42533..42869>)
>DS571259|intron|complement|61|join(<3249..4742,4803..5018>)
>DS571260|intron|complement|57|join(<15804..16033,16090..16217>)
>DS571260|intron|complement|59|join(<40542..41165,41224..41280>)
>DS571260|intron|complement|54|join(<41438..42285,42339..42507>)
>DS571260|intron|complement|50|join(<42339..42507,42557..42712>)
>DS571261|intron|53|join(<19700..19771,19824..22416>)
>DS571262|intron|140|join(<18686..19105,19245..21278>)

>DS571263|intron|70|join(<16754..17098,17168..19996>)
>DS571263|intron|complement|46|join(<20879..21419,21465..21488>)
>DS571264|intron|55|join(<17140..17565,17620..17700>)
>DS571264|intron|complement|72|join(<37006..37247,37319..37343>)
>DS571265|intron|65|join(<19557..20139,20204..20482>)
>DS571265|intron|99|join(<19313..19458,19557..20139>)
>DS571269|intron|complement|46|join(<27999..31516,31562..34184>)
>DS571270|intron|194|join(<12763..12945,13139..13264>)
>DS571270|intron|complement|47|join(<1643..1922,1969..2237>)
>DS571271|intron|complement|56|join(<7932..13150,13206..13671>)
>DS571272|intron|complement|49|join(<30832..31812,31861..32043>)
>DS571272|intron|56|join(<39637..42495,42551..42847>)
>DS571273|intron|70|join(<16701..16809,16879..17045>)
>DS571273|intron|complement|54|join(<20016..20162,20216..20236>)
>DS571275|intron|53|join(<18026..18300,18353..18566>)
>DS571276|intron|complement|76|join(<30970..32394,32470..32688>)
>DS571279|gene|588|(<754..1342>)
>DS571278|intron|46|join(<33891..33946,33992..34146>)
>DS571278|gene|345|(<31678..32023>)
>DS571279|intron|50|join(<754..802,852..1342>)
>DS571279|intron|55|join(<33364..33621,33676..34596>)
>DS571279|intron|57|join(<35382..35652,35709..37876>)
>DS571279|intron|59|join(<42100..42369,42428..43903>)
>DS571281|intron|51|join(<5907..6111,6162..6568>)
>DS571282|intron|complement|49|join(<12157..12681,12730..12876>)
>DS571283|gene|692|(<1047..1739>)
>DS571284|intron|complement|287|join(<7171..7673,7960..8137>)
>DS571284|intron|complement|48|join(<14708..14888,14936..15613>)
>DS571286|intron|235|join(<32437..32625,32860..33390>)
>DS571287|intron|176|join(<12742..12924,13100..13243>)
>DS571287|intron|complement|59|join(<27782..27965,28024..28198>)
>DS571287|gene|642|(<35984..36626>)
>DS571287|intron|206|join(<35984..36295,36501..36626>)
>DS571287|intron|91|join(<37065..37290,37381..37973>)
>DS571291|intron|60|join(<24949..25193,25253..25679>)
>DS571292|intron|34|join(<24324..25000,25034..25614>)
>DS571294|intron|complement|62|join(<20182..20981,21043..21292>)
>DS571294|intron|53|join(<12093..12713,12766..12896>)
>DS571294|intron|complement|53|join(<37380..37502,37555..37917>)
>DS571295|intron|71|join(<14856..15008,15079..18660>)
>DS571295|intron|complement|55|join(<28270..28578,28633..29445>)
>DS571297|intron|complement|68|join(<6864..7181,7249..7347>)
>DS571297|intron|59|join(<16853..17293,17352..17804>)
>DS571298|intron|complement|91|join(<2493..2947,3038..3095>)
>DS571298|intron|complement|35|join(<17331..17722,17757..17880>)
>DS571298|intron|complement|46|join(<17757..17880,17926..18234>)
>DS571301|intron|complement|65|join(<25038..25113,25178..25443>)
>DS571301|intron|complement|60|join(<2794..3165,3225..3566>)
>DS571302|intron|60|join(<22492..22596,22656..22917>)
>DS571303|intron|66|join(<13980..13982,14048..14261>)
>DS571303|intron|60|join(<31783..32337,32397..32555>)
>DS571304|intron|82|join(<24980..25408,25490..25590>)
>DS571311|intron|complement|83|join(<8168..8557,8640..8942>)
>DS571311|intron|complement|46|join(<31452..33277,33323..33395>)
>DS571312|intron|complement|58|join(<7505..7668,7726..7858>)
>DS571312|intron|complement|66|join(<12536..12910,12976..13080>)
>DS571313|intron|complement|63|join(<7017..7622,7685..7840>)
>DS571314|intron|complement|66|join(<29439..30270,30336..30604>)
>DS571315|intron|49|join(<12917..13110,13159..14079>)
>DS571315|intron|complement|136|join(<4144..5570,5706..5769>)
>DS571315|intron|complement|48|join(<10428..11214,11262..11573>)
>DS571315|intron|complement|148|join(<32139..33186,33334..33506>)
>DS571323|intron|complement|137|join(<13832..14315,14452..14795>)

>DS571323|intron|104|join(<27652..27748,27852..28434>)
>DS571324|intron|68|join(<16748..16846,16914..17231>)
>DS571326|intron|142|join(<14720..15228,15370..15909>)
>DS571330|intron|complement|70|join(<29914..30002,30072..30249>)
>DS571331|intron|complement|58|join(<1467..2357,2415..2760>)
>DS571331|intron|65|join(<10480..10902,10967..13486>)
>DS571331|intron|72|join(<19565..20041,20113..20265>)
>DS571332|intron|complement|54|join(<29194..30855,30909..31178>)
>DS571335|intron|52|join(<13730..13957,14009..14857>)
>DS571336|intron|complement|81|join(<6806..8421,8502..8826>)
>DS571337|intron|46|join(<14928..15035,15081..15361>)
>DS571337|intron|64|join(<20211..20363,20427..20735>)
>DS571337|intron|complement|58|join(<26243..26575,26633..26918>)
>DS571337|intron|complement|290|join(<1358..3631,3921..5330>)
>DS571338|intron|complement|149|join(<5062..5531,5680..5752>)
>DS571342|intron|62|join(<14438..15013,15075..15947>)
>DS571344|intron|55|join(<12291..12528,12583..12713>)
>DS571347|intron|complement|51|join(<10291..10331,10382..10664>)
>DS571349|intron|complement|62|join(<23544..24350,24412..24579>)
>DS571351|intron|complement|63|join(<795..986,1049..1300>)
>DS571356|intron|complement|353|join(<6097..7603,7956..8509>)
>DS571357|intron|complement|49|join(<5207..5326,5375..5683>)
>DS571358|intron|47|join(<16282..18859,18906..19390>)
>DS571362|intron|191|join(<15274..15760,15951..16486>)
>DS571365|intron|complement|59|join(<8516..8664,8723..8894>)
>DS571365|intron|complement|67|join(<10255..10365,10432..11787>)
>DS571367|intron|70|join(<5750..6047,6117..6247>)
>DS571368|gene|465|(<18831..19296>)
>DS571372|intron|complement|81|join(<11496..12002,12083..12769>)
>DS571373|intron|48|join(<12739..12864,12912..13289>)
>DS571374|intron|63|join(<17732..18030,18093..18243>)
>DS571376|intron|complement|54|join(<809..1954,2008..2136>)
>DS571380|intron|complement|63|join(<21607..21921,21984..22148>)
>DS571381|intron|54|join(<3445..3567,3621..4435>)
>DS571382|intron|complement|92|join(<6891..9616,9708..9864>)
>DS571385|intron|67|join(<16347..16362,16429..21041>)
>DS571386|intron|70|join(<10921..11046,11116..11316>)
>DS571386|intron|66|join(<16335..16679,16745..16996>)
>DS571386|intron|complement|65|join(<8981..9235,9300..9674>)
>DS571390|intron|46|join(<22040..22147,22193..22567>)
>DS571390|intron|50|join(<16546..16828,16878..16966>)
>DS571394|intron|complement|57|join(<10457..10632,10689..10878>)
>DS571397|intron|203|join(<17498..17857,18060..18703>)
>DS571400|intron|54|join(<7325..7513,7567..8007>)
>DS571400|intron|51|join(<7227..7274,7325..7513>)
>DS571400|intron|complement|60|join(<783..2426,2486..2794>)
>DS571400|intron|51|join(<17643..18155,18206..18802>)
>DS571401|intron|complement|63|join(<15558..16364,16427..16627>)
>DS571403|intron|complement|647|join(<8295..8698,9345..9366>)
>DS571406|intron|complement|55|join(<10937..11329,11384..11578>)
>DS571408|intron|complement|49|join(<12104..12550,12599..13441>)
>DS571411|intron|complement|51|join(<9987..10492,10543..10972>)
>DS571412|intron|53|join(<5671..6349,6402..7276>)
>DS571413|intron|49|join(<15259..15615,15664..16449>)
>DS571414|intron|complement|59|join(<10581..10813,10872..11498>)
>DS571425|intron|54|join(<5455..5671,5725..6284>)
>DS571427|intron|56|join(<3809..4389,4445..4520>)
>DS571439|intron|complement|59|join(<9502..11486,11545..11953>)
>DS571443|intron|complement|175|join(<1351..2048,2223..3300>)
>DS571450|intron|complement|46|join(<8466..10118,10164..10430>)
>DS571451|intron|complement|55|join(<13346..14832,14887..14998>)
>DS571452|intron|complement|56|join(<5406..6485,6541..6621>)
>DS571452|intron|complement|64|join(<11437..11682,11746..12261>)

>DS571459|intron|complement|60|join(<15990..16352,16412..16608>)
>DS571481|gene|385|(<14787..15172>)
>DS571487|intron|complement|69|join(<4881..5285,5354..8350>)
>DS571500|intron|complement|46|join(<7514..9993,10039..10438>)
>DS571507|intron|complement|39|join(<26..31,70..621>)
>DS571507|intron|47|join(<5069..5702,5749..5921>)
>DS571507|intron|complement|44|join(<6902..7065,7109..7457>)
>DS571518|intron|complement|91|join(<4517..4963,5054..5191>)
>DS571524|intron|51|join(<6934..8535,8586..9395>)
>DS571524|intron|122|join(<6789..6812,6934..8535>)
>DS571528|intron|46|join(<2611..2785,2831..2891>)
>DS571534|intron|complement|61|join(<5765..6059,6120..6142>)
>DS571534|intron|complement|78|join(<6280..7491,7569..7914>)
>DS571536|intron|325|join(<2629..4336,4661..4836>)
>DS571538|intron|94|join(<4594..4801,4895..5487>)
>DS571547|intron|complement|56|join(<4135..4234,4290..6436>)
>DS571553|intron|36|join(<1893..2600,2636..3385>)
>DS571561|intron|53|join(<4398..4564,4617..5020>)
>DS571563|intron|complement|89|join(<3114..3850,3939..5223>)
>DS571588|intron|63|join(<1632..1735,1798..2425>)
>DS571588|intron|60|join(<1156..1572,1632..1735>)
>DS571607|intron|47|join(<6034..6257,6304..6375>)
>DS571607|intron|56|join(<5336..5978,6034..6257>)
>DS571614|intron|48|join(<2092..2409,2457..2759>)
>DS571614|intron|68|join(<1926..2024,2092..2409>)
>DS571618|intron|complement|90|join(<133..151,241..563>)
>DS571624|intron|complement|45|join(<2392..2933,2978..3089>)
>DS571635|intron|complement|60|join(<112..129,189..1919>)
>DS571643|intron|complement|63|join(<3284..3646,3709..4023>)
>DS571646|intron|complement|58|join(<3717..4859,4917..5072>)
>DS571647|intron|116|join(<685..1762,1878..2216>)
>DS571653|intron|55|join(<901..978,1033..1128>)
>DS571675|intron|complement|68|join(<824..962,1030..1402>)
>DS571679|intron|complement|117|join(<2189..2329,2446..2637>)
>DS571684|intron|45|join(<2733..2889,2934..4615>)
>DS571685|intron|complement|55|join(<2332..2485,2540..3804>)
>DS571748|intron|complement|59|join(<552..692,751..1968>)
>DS571790|intron|complement|53|join(<746..1696,1749..1979>)
>DS571810|intron|complement|146|join(<553..781,927..1105>)
>DS571818|intron|41|join(<2394..2624,2665..2982>)
>DS571845|intron|45|join(<340..525,570..1322>)
>DS571865|intron|complement|47|join(<81..128,175..759>)
>DS571869|intron|24|join(<673..821,845..1271>)
>DS571872|intron|complement|90|join(<333..701,791..1048>)
>DS571879|intron|complement|92|join(<98..183,275..2180>)
>DS571885|intron|complement|62|join(<1610..1653,1715..2132>)
>DS572244|intron|51|join(<305..587,638..678>)
>DS572314|intron|complement|92|join(<670..991,1083..1339>)
>DS572478|intron|complement|42|join(<226..1039,1081..1124>)
>DS572647|intron|complement|73|join(<140..329,402..1012>)

Predicted as "FALSE"

>DS571145|intron|59|join(<404713..404807,404866..405451>)
>DS571145|intron|49|join(<437123..437989,438038..439043>)
>DS571145|intron|49|join(<297868..298371,298420..299312>)
>DS571145|intron|48|join(<404591..404665,404713..404807>)
>DS571145|intron|53|join(<418403..418503,418556..419315>)
>DS571145|intron|complement|82|join(<444035..444329,444411..444538>)
>DS571145|intron|complement|47|join(<168514..168867,168914..169203>)
>DS571145|intron|63|join(<313614..313913,313976..313981>)

>DS571145|intron|complement|58|join(<16316..23458,23516..23806>)
>DS571145|intron|complement|53|join(<127021..128134,128187..128409>)
>DS571145|intron|complement|50|join(<203940..205625,205675..205830>)
>DS571145|intron|complement|54|join(<323515..323895,323949..324657>)
>DS571145|intron|complement|62|join(<143062..143082,143144..143815>)
>DS571145|intron|62|join(<226227..227483,227545..227865>)
>DS571145|intron|complement|59|join(<515960..517034,517093..517256>)
>DS571145|intron|complement|54|join(<341635..342263,342317..342519>)
>DS571145|intron|42|join(<82012..84036,84078..84506>)
>DS571145|intron|69|join(<252114..252296,252365..252565>)
>DS571145|intron|139|join(<279142..279257,279396..279997>)
>DS571145|intron|52|join(<98509..98901,98953..99117>)
>DS571145|intron|52|join(<107563..107719,107771..107898>)
>DS571145|intron|84|join(<382939..382949,383033..383513>)
>DS571145|intron|67|join(<388933..389183,389250..390753>)
>DS571146|intron|44|join(<7163..7260,7304..7503>)
>DS571146|intron|81|join(<7003..7082,7163..7260>)
>DS571146|intron|60|join(<139397..139492,139552..139608>)
>DS571146|intron|45|join(<188412..188685,188730..189079>)
>DS571146|intron|complement|97|join(<145729..152430,152527..153192>)
>DS571146|intron|complement|61|join(<197842..198852,198913..199560>)
>DS571146|intron|complement|193|join(<199716..201722,201915..202217>)
>DS571146|intron|complement|59|join(<28184..28770,28829..29233>)
>DS571146|intron|complement|64|join(<34287..34793,34857..35660>)
>DS571146|intron|48|join(<202850..202996,203044..203577>)
>DS571146|gene|366|(<38488..38854>)
>DS571146|intron|complement|45|join(<62860..63028,63073..63290>)
>DS571146|intron|complement|52|join(<66745..67038,67090..67905>)
>DS571146|intron|66|join(<69221..69584,69650..70248>)
>DS571146|intron|complement|56|join(<91379..91842,91898..92206>)
>DS571146|intron|complement|47|join(<124373..125007,125054..125555>)
>DS571146|intron|complement|46|join(<125713..125953,125999..126408>)
>DS571146|intron|complement|46|join(<132805..136881,136927..137820>)
>DS571147|gene|494|(<117445..117939>)
>DS571147|intron|62|join(<135372..135509,135571..135729>)
>DS571147|intron|complement|201|join(<4085..4725,4926..4950>)
>DS571147|intron|73|join(<48846..49685,49758..49838>)
>DS571147|intron|52|join(<6313..7584,7636..7710>)
>DS571147|intron|85|join(<14010..14370,14455..15842>)
>DS571147|intron|complement|54|join(<63752..63933,63987..64218>)
>DS571147|intron|complement|144|join(<128234..128912,129056..129303>)
>DS571147|intron|complement|60|join(<17448..17610,17670..18106>)
>DS571147|intron|complement|61|join(<165901..166699,166760..166884>)
>DS571147|intron|48|join(<24599..24643,24691..24849>)
>DS571148|intron|47|join(<40751..40882,40929..41962>)
>DS571148|intron|58|join(<124483..124584,124642..124920>)
>DS571148|intron|58|join(<2317..2701,2759..4809>)
>DS571148|intron|complement|45|join(<170978..171678,171723..171876>)
>DS571148|intron|50|join(<25443..26005,26055..26571>)
>DS571148|intron|complement|108|join(<56267..56834,56942..57054>)
>DS571148|intron|48|join(<60186..60688,60736..61705>)
>DS571148|intron|complement|59|join(<30481..31963,32022..32122>)
>DS571148|intron|complement|72|join(<105005..105138,105210..106437>)
>DS571148|intron|complement|74|join(<121860..122331,122405..122445>)
>DS571149|intron|complement|363|join(<65028..65378,65741..65809>)
>DS571149|intron|71|join(<69945..69975,70046..71967>)
>DS571149|intron|complement|112|join(<79756..84372,84484..85509>)
>DS571149|intron|complement|53|join(<93866..94428,94481..94780>)
>DS571149|intron|complement|76|join(<107789..107973,108049..109189>)
>DS571149|intron|complement|59|join(<134431..134898,134957..135469>)
>DS571149|intron|49|join(<161996..162146,162195..162778>)
>DS571149|intron|complement|53|join(<182236..182679,182732..182852>)
>DS571150|gene|403|(<128582..128985>)

>DS571150|intron|53|join(<119082..119219,119272..119469>)
>DS571150|intron|52|join(<99925..100380,100432..101229>)
>DS571150|intron|48|join(<20232..20634,20682..21163>)
>DS571150|intron|65|join(<151445..151551,151616..152516>)
>DS571150|intron|complement|72|join(<61673..62668,62740..63159>)
>DS571151|intron|53|join(<130951..131023,131076..131353>)
>DS571151|intron|85|join(<2859..5632,5717..5774>)
>DS571151|intron|complement|77|join(<153567..153587,153664..153805>)
>DS571152|intron|complement|62|join(<3628..3968,4030..4678>)
>DS571152|intron|complement|58|join(<82683..82865,82923..83777>)
>DS571152|intron|75|join(<127577..127799,127874..129087>)
>DS571152|intron|complement|59|join(<51868..52819,52878..53071>)
>DS571152|intron|68|join(<53174..53273,53341..53573>)
>DS571152|intron|complement|97|join(<61213..62716,62813..63516>)
>DS571153|intron|54|join(<51987..52292,52346..52432>)
>DS571153|intron|complement|51|join(<86825..87372,87423..87825>)
>DS571153|intron|52|join(<95352..95456,95508..96503>)
>DS571153|intron|complement|55|join(<129032..130301,130356..130597>)
>DS571154|intron|53|join(<35589..36190,36243..36505>)
>DS571154|intron|complement|70|join(<740..7803,7873..8410>)
>DS571154|intron|complement|48|join(<79242..79509,79557..79738>)
>DS571154|intron|182|join(<83895..84088,84270..86760>)
>DS571155|intron|64|join(<108940..109031,109095..109151>)
>DS571155|intron|41|join(<153308..153318,153359..153498>)
>DS571155|intron|110|join(<54788..54799,54909..58592>)
>DS571155|intron|complement|137|join(<71513..76455,76592..79967>)
>DS571155|intron|complement|56|join(<152126..152406,152462..152879>)
>DS571156|intron|43|join(<21062..21353,21396..21715>)
>DS571156|intron|complement|196|join(<3521..4495,4691..4713>)
>DS571156|intron|complement|51|join(<49026..50527,50578..50791>)
>DS571156|intron|complement|58|join(<58593..58714,58772..58931>)
>DS571156|intron|complement|53|join(<61222..62488,62541..62761>)
>DS571156|intron|complement|52|join(<89590..89785,89837..90641>)
>DS571156|intron|complement|56|join(<89837..90641,90697..90826>)
>DS571156|intron|complement|77|join(<94770..95137,95214..95527>)
>DS571156|intron|complement|54|join(<99329..99664,99718..99818>)
>DS571156|intron|61|join(<116873..116927,116988..119800>)
>DS571157|intron|complement|74|join(<127559..127610,127684..128789>)
>DS571158|intron|168|join(<103651..103901,104069..105203>)
>DS571158|intron|70|join(<103558..103581,103651..103901>)
>DS571158|intron|complement|81|join(<123468..124598,124679..124756>)
>DS571158|intron|complement|74|join(<34478..34517,34591..34751>)
>DS571158|intron|47|join(<38366..38713,38760..40854>)
>DS571158|intron|complement|45|join(<94141..94743,94788..95027>)
>DS571159|intron|83|join(<45896..46191,46274..46584>)
>DS571159|intron|59|join(<116542..117320,117379..118180>)
>DS571159|intron|50|join(<72202..72501,72551..72676>)
>DS571159|intron|complement|51|join(<129633..129706,129757..130018>)
>DS571159|intron|89|join(<30627..31766,31855..32004>)
>DS571160|intron|complement|60|join(<22718..23239,23299..24111>)
>DS571160|intron|complement|51|join(<63726..63738,63789..67315>)
>DS571160|intron|48|join(<124630..125061,125109..127643>)
>DS571160|intron|47|join(<136384..136561,136608..137221>)
>DS571160|intron|61|join(<137349..137657,137718..138295>)
>DS571161|intron|55|join(<45229..45234,45289..45345>)
>DS571161|intron|complement|52|join(<45935..51006,51058..51805>)
>DS571161|intron|complement|68|join(<15847..16815,16883..17182>)
>DS571161|intron|complement|70|join(<61730..61867,61937..62068>)
>DS571161|intron|complement|57|join(<24045..24200,24257..24429>)
>DS571161|intron|102|join(<63904..63912,64014..65540>)
>DS571161|intron|complement|53|join(<72355..73398,73451..73807>)
>DS571161|intron|complement|46|join(<28119..28125,28171..28326>)
>DS571161|intron|63|join(<93401..94018,94081..94242>)

>DS571161|intron|91|join(<99589..100107,100198..100494>)
>DS571161|intron|complement|58|join(<100577..100651,100709..100797>)
>DS571161|intron|complement|63|join(<112905..113809,113872..114004>)
>DS571161|intron|complement|113|join(<114136..114396,114509..114900>)
>DS571162|intron|85|join(<118521..118724,118809..119413>)
>DS571162|intron|complement|51|join(<63991..64848,64899..65363>)
>DS571162|intron|63|join(<111436..111751,111814..115520>)
>DS571162|intron|complement|58|join(<117184..117751,117809..118212>)
>DS571162|intron|complement|87|join(<85944..87221,87308..87433>)
>DS571162|intron|complement|61|join(<106083..106144,106205..106862>)
>DS571163|intron|47|join(<42013..42148,42195..42961>)
>DS571163|intron|54|join(<58020..58163,58217..59002>)
>DS571163|intron|complement|42|join(<112957..114130,114172..114235>)
>DS571163|intron|147|join(<118847..118936,119083..120414>)
>DS571164|intron|49|join(<69591..69750,69799..70017>)
>DS571164|gene|678|(<118760..119438>)
>DS571164|gene|702|(<127254..127956>)
>DS571164|intron|53|join(<69534..69538,69591..69750>)
>DS571164|intron|57|join(<125015..125076,125133..125380>)
>DS571164|intron|59|join(<127254..127290,127349..127956>)
>DS571164|intron|complement|52|join(<39987..41662,41714..41825>)
>DS571164|intron|62|join(<68080..69191,69253..69361>)
>DS571165|intron|92|join(<7831..7965,8057..9136>)
>DS571165|intron|complement|88|join(<60761..64878,64966..66865>)
>DS571165|intron|57|join(<70726..71113,71170..72062>)
>DS571165|intron|56|join(<31282..31398,31454..31768>)
>DS571165|intron|complement|50|join(<102498..102870,102920..103328>)
>DS571165|intron|complement|54|join(<32484..33321,33375..34042>)
>DS571165|intron|54|join(<106806..107059,107113..107542>)
>DS571165|intron|49|join(<56616..56658,56707..56996>)
>DS571166|intron|60|join(<67533..67715,67775..68286>)
>DS571166|intron|55|join(<92854..93061,93116..93531>)
>DS571166|intron|56|join(<30373..32585,32641..32902>)
>DS571166|intron|complement|64|join(<39420..39498,39562..41351>)
>DS571166|intron|74|join(<52396..52534,52608..53401>)
>DS571167|intron|complement|115|join(<17037..17752,17867..18698>)
>DS571167|intron|complement|76|join(<27575..27678,27754..27985>)
>DS571167|intron|complement|51|join(<27754..27985,28036..28194>)
>DS571167|intron|complement|54|join(<38282..39721,39775..40197>)
>DS571167|intron|complement|55|join(<44957..46655,46710..47272>)
>DS571167|intron|complement|63|join(<47476..47660,47723..48163>)
>DS571167|intron|61|join(<53646..54170,54231..54341>)
>DS571167|intron|63|join(<54532..54965,55028..55790>)
>DS571167|intron|61|join(<107218..109005,109066..110199>)
>DS571168|intron|complement|81|join(<29548..39226,39307..39497>)
>DS571168|intron|complement|48|join(<73103..76078,76126..76776>)
>DS571168|intron|complement|62|join(<88290..91610,91672..91935>)
>DS571169|intron|52|join(<94179..94550,94602..94712>)
>DS571169|intron|58|join(<93819..94121,94179..94550>)
>DS571169|intron|complement|58|join(<83330..83557,83615..83774>)
>DS571170|intron|53|join(<44457..44675,44728..45144>)
>DS571170|intron|45|join(<85365..85523,85568..85864>)
>DS571170|intron|50|join(<98469..98810,98860..99364>)
>DS571170|intron|complement|47|join(<46858..48112,48159..48272>)
>DS571170|intron|60|join(<101687..101764,101824..105426>)
>DS571170|intron|complement|60|join(<69872..71005,71065..71238>)
>DS571170|intron|complement|63|join(<79911..80008,80071..80288>)
>DS571170|intron|complement|47|join(<80071..80288,80335..80503>)
>DS571171|intron|59|join(<47669..51409,51468..51584>)
>DS571172|intron|126|join(<46499..46658,46784..47106>)
>DS571172|intron|complement|64|join(<52533..53397,53461..53578>)
>DS571172|intron|complement|128|join(<18135..19702,19830..19978>)
>DS571172|intron|56|join(<27920..31960,32016..32216>)

>DS571173|intron|complement|175|join(<60610..61165,61340..61403>)
>DS571173|intron|298|join(<20341..20360,20658..20925>)
>DS571173|intron|complement|56|join(<81087..81266,81322..82230>)
>DS571173|intron|complement|51|join(<82343..83267,83318..84086>)
>DS571173|intron|51|join(<85064..85719,85770..86310>)
>DS571174|intron|complement|49|join(<33602..34047,34096..35338>)
>DS571174|intron|53|join(<48110..48333,48386..48926>)
>DS571175|intron|complement|57|join(<4815..5943,6000..6142>)
>DS571175|intron|complement|45|join(<71295..72332,72377..72544>)
>DS571175|intron|complement|48|join(<75059..75210,75258..75527>)
>DS571175|intron|complement|98|join(<24886..25406,25504..25891>)
>DS571175|intron|complement|67|join(<62031..62154,62221..62491>)
>DS571175|intron|complement|161|join(<67060..67404,67565..68098>)
>DS571176|intron|complement|58|join(<40035..40433,40491..40586>)
>DS571176|intron|complement|55|join(<53268..53337,53392..53492>)
>DS571176|intron|complement|55|join(<56554..57261,57316..57901>)
>DS571176|intron|complement|58|join(<57316..57901,57959..58098>)
>DS571176|intron|72|join(<59751..67706,67778..70780>)
>DS571176|intron|54|join(<78778..79311,79365..80048>)
>DS571177|intron|56|join(<36303..36487,36543..37920>)
>DS571177|intron|75|join(<58330..58457,58532..60095>)
>DS571177|intron|54|join(<66403..67003,67057..67250>)
>DS571177|intron|complement|59|join(<84980..85099,85158..86532>)
>DS571178|intron|complement|53|join(<8511..8735,8788..9430>)
>DS571178|intron|complement|55|join(<8788..9430,9485..9651>)
>DS571178|intron|complement|45|join(<39072..39193,39238..39367>)
>DS571178|intron|58|join(<43206..43962,44020..44324>)
>DS571178|intron|complement|46|join(<55906..56019,56065..56583>)
>DS571178|intron|47|join(<63056..63311,63358..64556>)
>DS571178|intron|complement|203|join(<87733..88353,88556..88912>)
>DS571179|intron|69|join(<79023..79565,79634..79834>)
>DS571180|intron|56|join(<14487..15061,15117..15282>)
>DS571180|intron|complement|56|join(<20232..20442,20498..21194>)
>DS571180|intron|complement|57|join(<23587..23699,23756..25175>)
>DS571181|intron|complement|97|join(<37468..37788,37885..37989>)
>DS571181|intron|complement|315|join(<57483..58474,58789..60412>)
>DS571181|intron|65|join(<75422..75803,75868..76049>)
>DS571182|intron|57|join(<73475..73634,73691..74796>)
>DS571182|intron|45|join(<73377..73430,73475..73634>)
>DS571182|intron|74|join(<74988..74990,75064..75289>)
>DS571182|intron|complement|47|join(<24061..24294,24341..24700>)
>DS571182|intron|112|join(<76438..77420,77532..82929>)
>DS571182|intron|complement|59|join(<44446..45066,45125..45241>)
>DS571182|intron|complement|75|join(<45125..45241,45316..45451>)
>DS571182|intron|complement|54|join(<58563..60098,60152..60317>)
>DS571183|intron|43|join(<10654..17365,17408..18618>)
>DS571183|intron|complement|63|join(<21542..22328,22391..22791>)
>DS571183|intron|85|join(<28694..31315,31400..31436>)
>DS571183|intron|complement|51|join(<31523..31859,31910..32046>)
>DS571183|intron|complement|199|join(<62972..63568,63767..64123>)
>DS571183|intron|136|join(<70666..70729,70865..71307>)
>DS571184|intron|complement|57|join(<86188..86669,86726..86728>)
>DS571184|intron|72|join(<13637..13905,13977..15189>)
>DS571184|intron|complement|47|join(<16220..16423,16470..16565>)
>DS571184|intron|complement|107|join(<48607..48893,49000..49666>)
>DS571185|intron|complement|38|join(<39693..40427,40465..40844>)
>DS571185|intron|complement|51|join(<40465..40844,40895..41090>)
>DS571186|intron|60|join(<1555..1657,1717..2669>)
>DS571186|intron|complement|57|join(<46633..48158,48215..48635>)
>DS571186|intron|complement|45|join(<50890..51065,51110..51887>)
>DS571187|gene|730|(<46375..47105>)
>DS571187|intron|complement|75|join(<72772..72795,72870..73664>)
>DS571187|intron|complement|51|join(<78367..78912,78963..79233>)

>DS571188|intron|62|join(<24725..24925,24987..32120>)
>DS571188|intron|complement|58|join(<21692..21915,21973..22387>)
>DS571188|intron|complement|104|join(<57545..58035,58139..58268>)
>DS571189|intron|46|join(<25818..25903,25949..26056>)
>DS571189|intron|75|join(<52991..53082,53157..53721>)
>DS571190|intron|56|join(<40619..40794,40850..41523>)
>DS571190|intron|60|join(<7573..7715,7775..8063>)
>DS571190|intron|complement|51|join(<77269..77955,78006..78299>)
>DS571191|intron|50|join(<22693..22793,22843..23133>)
>DS571191|intron|59|join(<22442..22634,22693..22793>)
>DS571191|intron|48|join(<39657..39941,39989..40416>)
>DS571191|intron|complement|50|join(<43142..43630,43680..43775>)
>DS571191|intron|complement|83|join(<67589..70618,70701..70943>)
>DS571191|intron|complement|55|join(<73102..74140,74195..74274>)
>DS571192|intron|complement|84|join(<7187..8293,8377..8406>)
>DS571192|intron|complement|48|join(<8556..10487,10535..10618>)
>DS571192|intron|complement|53|join(<59142..59454,59507..59574>)
>DS571192|intron|complement|54|join(<64729..64854,64908..65168>)
>DS571193|intron|complement|56|join(<67380..67585,67641..69084>)
>DS571195|intron|complement|50|join(<56298..57691,57741..57822>)
>DS571196|intron|53|join(<33020..33030,33083..36647>)
>DS571196|intron|complement|52|join(<62539..62769,62821..63293>)
>DS571196|intron|complement|44|join(<73086..73233,73277..73899>)
>DS571197|intron|54|join(<42..59,113..565>)
>DS571197|intron|47|join(<9594..9707,9754..10341>)
>DS571197|intron|complement|142|join(<44830..44966,45108..45244>)
>DS571198|intron|44|join(<9377..9424,9468..9818>)
>DS571198|intron|complement|138|join(<3071..3241,3379..4822>)
>DS571198|intron|complement|59|join(<10166..11131,11190..12029>)
>DS571198|intron|complement|65|join(<23066..23379,23444..24158>)
>DS571199|intron|184|join(<42583..42726,42910..43422>)
>DS571199|intron|53|join(<45939..48020,48073..48267>)
>DS571199|intron|complement|68|join(<29159..29376,29444..29536>)
>DS571199|intron|53|join(<62334..64721,64774..64929>)
>DS571200|intron|188|join(<30960..31109,31297..31601>)
>DS571200|intron|complement|51|join(<46287..46404,46455..46612>)
>DS571200|intron|62|join(<54483..54537,54599..55992>)
>DS571200|intron|50|join(<61739..62023,62073..62753>)
>DS571201|intron|52|join(<69687..69953,70005..70550>)
>DS571201|intron|60|join(<18913..19048,19108..19213>)
>DS571201|intron|complement|63|join(<23852..24900,24963..25602>)
>DS571201|intron|224|join(<62511..62748,62972..63342>)
>DS571202|intron|complement|54|join(<30193..30860,30914..31205>)
>DS571202|intron|complement|47|join(<34622..35423,35470..35576>)
>DS571202|intron|55|join(<58816..58963,59018..61143>)
>DS571203|intron|complement|48|join(<5581..5739,5787..5917>)
>DS571203|intron|complement|80|join(<9308..11681,11761..11927>)
>DS571203|intron|complement|65|join(<30326..30450,30515..30869>)
>DS571203|intron|complement|54|join(<30515..30869,30923..31348>)
>DS571203|intron|235|join(<56000..56969,57204..59221>)
>DS571204|intron|62|join(<31043..32074,32136..32419>)
>DS571204|intron|75|join(<30821..30968,31043..32074>)
>DS571204|intron|complement|51|join(<46047..46316,46367..46451>)
>DS571204|intron|complement|58|join(<47097..47426,47484..47525>)
>DS571204|intron|complement|57|join(<47623..48292,48349..48590>)
>DS571204|intron|complement|61|join(<48722..49088,49149..49378>)
>DS571205|intron|50|join(<17592..17921,17971..18468>)
>DS571205|intron|complement|113|join(<48731..54120,54233..54731>)
>DS571205|intron|complement|74|join(<56800..57051,57125..57757>)
>DS571206|intron|64|join(<32781..32983,33047..33095>)
>DS571206|intron|complement|62|join(<33117..34018,34080..34288>)
>DS571207|intron|complement|54|join(<12577..13240,13294..14183>)
>DS571207|intron|complement|51|join(<28959..31886,31937..32401>)

>DS571207|gene|505|(<36418..36923>)
>DS571207|intron|61|join(<46287..46752,46813..46887>)
>DS571209|intron|complement|254|join(<19852..21768,22022..22102>)
>DS571210|intron|50|join(<54729..54744,54794..55890>)
>DS571210|intron|complement|45|join(<58355..58693,58738..59129>)
>DS571212|intron|complement|173|join(<56560..58255,58428..58615>)
>DS571213|intron|complement|57|join(<15062..16355,16412..16629>)
>DS571213|intron|complement|68|join(<16412..16629,16697..16789>)
>DS571213|intron|complement|59|join(<13177..13384,13443..13529>)
>DS571213|intron|complement|88|join(<34294..34713,34801..34872>)
>DS571214|intron|complement|65|join(<124..153,218..454>)
>DS571214|intron|89|join(<497..943,1032..1286>)
>DS571214|intron|complement|62|join(<25723..26234,26296..26353>)
>DS571214|intron|complement|93|join(<26296..26353,26446..26502>)
>DS571216|intron|50|join(<17288..17500,17550..18395>)
>DS571216|intron|63|join(<16962..17225,17288..17500>)
>DS571216|intron|complement|63|join(<38807..39036,39099..39325>)
>DS571217|intron|55|join(<1138..1836,1891..2095>)
>DS571217|intron|complement|62|join(<24287..24582,24644..24952>)
>DS571217|intron|140|join(<40941..41630,41770..42357>)
>DS571217|intron|complement|52|join(<53091..53320,53372..53909>)
>DS571217|intron|complement|59|join(<54011..54067,54126..54578>)
>DS571218|intron|54|join(<26910..27386,27440..28042>)
>DS571218|intron|47|join(<56292..56447,56494..56889>)
>DS571218|intron|111|join(<60986..61408,61519..61881>)
>DS571219|intron|complement|58|join(<22708..22889,22947..24582>)
>DS571219|intron|complement|45|join(<25432..25950,25995..26096>)
>DS571219|intron|complement|116|join(<26301..26765,26881..26889>)
>DS571219|intron|complement|67|join(<41659..42106,42173..42371>)
>DS571219|intron|63|join(<60074..60450,60513..61065>)
>DS571220|intron|complement|48|join(<17309..19054,19102..19338>)
>DS571220|intron|complement|66|join(<28901..28975,29041..29250>)
>DS571220|intron|48|join(<35266..35635,35683..37973>)
>DS571220|intron|54|join(<38331..38417,38471..39943>)
>DS571221|intron|58|join(<54866..55009,55067..55450>)
>DS571221|intron|complement|60|join(<16652..17575,17635..17934>)
>DS571221|intron|191|join(<33679..34415,34606..35359>)
>DS571223|intron|59|join(<6108..6215,6274..6433>)
>DS571223|intron|67|join(<14217..14534,14601..14819>)
>DS571223|intron|51|join(<16128..16586,16637..16945>)
>DS571223|intron|61|join(<30348..30431,30492..31016>)
>DS571223|intron|49|join(<36672..37148,37197..37361>)
>DS571224|gene|635|(<16761..17396>)
>DS571224|intron|70|join(<16761..17086,17156..17396>)
>DS571224|intron|53|join(<17445..17492,17545..18123>)
>DS571224|intron|complement|51|join(<53694..53920,53971..54428>)
>DS571224|intron|74|join(<19720..19863,19937..20417>)
>DS571226|intron|89|join(<27088..27277,27366..27466>)
>DS571226|intron|complement|57|join(<29892..31307,31364..31430>)
>DS571226|intron|complement|89|join(<41552..42051,42140..42668>)
>DS571227|intron|43|join(<26012..26190,26233..27331>)
>DS571227|intron|171|join(<30556..31323,31494..31607>)
>DS571227|intron|complement|58|join(<53249..53343,53401..53494>)
>DS571229|intron|complement|49|join(<33321..33846,33895..34019>)
>DS571230|intron|complement|47|join(<29370..29519,29566..29779>)
>DS571230|intron|complement|72|join(<50187..50193,50265..50324>)
>DS571231|intron|complement|85|join(<9787..10581,10666..11331>)
>DS571232|intron|56|join(<33108..33500,33556..34605>)
>DS571232|intron|64|join(<48896..49231,49295..50590>)
>DS571233|intron|65|join(<4233..4352,4417..4554>)
>DS571233|intron|57|join(<12197..12199,12256..12810>)
>DS571234|intron|45|join(<30058..30733,30778..30944>)
>DS571235|intron|complement|54|join(<32713..32895,32949..35357>)

>DS571235|intron|complement|61|join(<36840..39304,39365..39458>)
>DS571235|intron|65|join(<45784..45941,46006..47494>)
>DS571236|intron|complement|53|join(<21636..22687,22740..23004>)
>DS571238|intron|57|join(<16765..16768,16825..16956>)
>DS571239|intron|complement|51|join(<48074..49017,49068..49311>)
>DS571240|intron|complement|49|join(<20325..21422,21471..21554>)
>DS571240|intron|52|join(<26800..27157,27209..34224>)
>DS571241|intron|49|join(<36066..36206,36255..37832>)
>DS571242|intron|64|join(<12059..13020,13084..13201>)
>DS571242|intron|complement|70|join(<36553..36656,36726..38319>)
>DS571242|intron|79|join(<38587..38861,38940..40473>)
>DS571243|intron|49|join(<50152..50292,50341..50610>)
>DS571243|intron|86|join(<7312..7383,7469..9221>)
>DS571243|intron|complement|52|join(<9595..9915,9967..10028>)
>DS571243|intron|52|join(<6152..6294,6346..7152>)
>DS571243|intron|complement|63|join(<14057..14167,14230..15621>)
>DS571243|intron|60|join(<34477..36309,36369..36508>)
>DS571245|intron|49|join(<10960..11163,11212..11289>)
>DS571246|intron|63|join(<9656..9909,9972..10230>)
>DS571246|intron|complement|56|join(<16637..16786,16842..17250>)
>DS571247|intron|complement|54|join(<25156..25279,25333..25809>)
>DS571247|gene|639|(<32618..33257>)
>DS571248|intron|complement|58|join(<16008..16093,16151..16399>)
>DS571248|intron|complement|57|join(<17066..18641,18698..18864>)
>DS571248|intron|complement|97|join(<30975..31146,31243..31490>)
>DS571250|intron|61|join(<29420..29585,29646..29840>)
>DS571250|intron|complement|234|join(<26763..27123,27357..28060>)
>DS571250|intron|165|join(<42406..42894,43059..43463>)
>DS571251|intron|54|join(<34092..34313,34367..40978>)
>DS571252|intron|complement|125|join(<19884..22259,22384..22407>)
>DS571253|intron|58|join(<20585..20693,20751..21049>)
>DS571254|intron|complement|75|join(<18186..18748,18823..18944>)
>DS571254|intron|complement|57|join(<22718..22822,22879..23289>)
>DS571255|intron|58|join(<27687..27808,27866..28637>)
>DS571255|intron|143|join(<1179..1311,1454..2391>)
>DS571256|intron|complement|71|join(<21739..22241,22312..22408>)
>DS571256|intron|complement|49|join(<35003..35310,35359..35398>)
>DS571257|intron|53|join(<26875..27006,27059..27140>)
>DS571257|intron|complement|92|join(<5222..5394,5486..5606>)
>DS571257|intron|62|join(<28283..29877,29939..30152>)
>DS571258|intron|complement|57|join(<33754..33864,33921..35387>)
>DS571259|intron|complement|102|join(<75..105,207..460>)
>DS571259|intron|complement|58|join(<1306..2154,2212..2508>)
>DS571259|intron|51|join(<5142..5535,5586..5647>)
>DS571260|intron|complement|57|join(<14814..16033,16090..16261>)
>DS571260|intron|complement|62|join(<16090..16217,16279..16364>)
>DS571260|intron|complement|67|join(<18090..18216,18283..18383>)
>DS571260|intron|complement|46|join(<27519..27701,27747..27799>)
>DS571260|intron|complement|77|join(<28787..29161,29238..29498>)
>DS571261|intron|63|join(<32143..33248,33311..33419>)
>DS571261|intron|complement|76|join(<43833..44318,44394..44396>)
>DS571262|intron|71|join(<11899..13671,13742..13897>)
>DS571263|intron|complement|55|join(<9643..9738,9793..9928>)
>DS571263|intron|complement|47|join(<9793..9928,9975..10195>)
>DS571263|intron|complement|69|join(<25175..25192,25261..25581>)
>DS571266|intron|complement|61|join(<15590..15684,15745..16933>)
>DS571266|intron|complement|45|join(<26811..27292,27337..27373>)
>DS571266|intron|68|join(<29951..30658,30726..30806>)
>DS571267|intron|63|join(<17476..18133,18196..19172>)
>DS571268|intron|complement|54|join(<34727..34882,34936..35194>)
>DS571268|intron|60|join(<43040..43046,43106..44109>)
>DS571272|intron|48|join(<16459..16581,16629..16943>)
>DS571272|intron|57|join(<19892..21320,21377..21489>)

>DS571272|intron|59|join(<14517..14630,14689..15168>)
>DS571272|intron|48|join(<15317..15574,15622..16215>)
>DS571272|intron|complement|57|join(<32134..32258,32315..33407>)
>DS571273|intron|complement|273|join(<33361..33730,34003..34049>)
>DS571274|intron|complement|53|join(<15439..15583,15636..15885>)
>DS571275|gene|540|(<18026..18566>)
>DS571275|intron|51|join(<12671..12721,12772..13050>)
>DS571276|intron|115|join(<35160..35243,35358..36209>)
>DS571276|intron|241|join(<34707..34919,35160..35243>)
>DS571276|intron|50|join(<3822..4025,4075..4605>)
>DS571277|intron|54|join(<26232..26365,26419..26592>)
>DS571278|intron|75|join(<18269..18786,18861..19320>)
>DS571279|intron|48|join(<31266..31278,31326..31651>)
>DS571280|intron|73|join(<24183..24506,24579..24776>)
>DS571280|intron|48|join(<28077..28505,28553..29539>)
>DS571280|intron|complement|172|join(<32129..32277,32449..32764>)
>DS571283|intron|45|join(<3062..3145,3190..3356>)
>DS571284|intron|57|join(<25479..25501,25558..26536>)
>DS571286|intron|complement|49|join(<31392..31528,31577..31850>)
>DS571286|intron|complement|52|join(<31577..31850,31902..31967>)
>DS571287|intron|154|join(<13100..13243,13397..14059>)
>DS571287|intron|complement|50|join(<28024..28198,28248..28413>)
>DS571287|intron|complement|188|join(<37870..37944,38132..39943>)
>DS571291|intron|73|join(<12181..12613,12686..12747>)
>DS571292|intron|250|join(<23800..24074,24324..25000>)
>DS571292|intron|51|join(<7863..8150,8201..9901>)
>DS571292|intron|63|join(<15214..15606,15669..16091>)
>DS571292|intron|complement|146|join(<17277..17446,17592..17835>)
>DS571293|intron|complement|85|join(<18284..18451,18536..18844>)
>DS571293|intron|complement|57|join(<18536..18844,18901..19404>)
>DS571293|intron|80|join(<32383..32549,32629..32842>)
>DS571294|intron|complement|53|join(<13585..13807,13860..14291>)
>DS571296|intron|58|join(<15847..15926,15984..16838>)
>DS571296|intron|52|join(<17472..17534,17586..18536>)
>DS571297|intron|71|join(<9820..10450,10521..10870>)
>DS571297|intron|complement|197|join(<23789..24781,24978..25250>)
>DS571298|intron|complement|49|join(<22067..22348,22397..22502>)
>DS571298|gene|633|(<34032..34665>)
>DS571298|intron|47|join(<34032..34244,34291..34665>)
>DS571299|intron|complement|59|join(<36302..36399,36458..36505>)
>DS571300|intron|complement|53|join(<7208..7502,7555..7815>)
>DS571302|intron|complement|52|join(<33254..33841,33893..34063>)
>DS571304|intron|119|join(<25490..25590,25709..27398>)
>DS571305|intron|54|join(<12214..12273,12327..14174>)
>DS571305|intron|179|join(<14469..14826,15005..15207>)
>DS571306|intron|complement|70|join(<5863..6030,6100..6312>)
>DS571307|gene|509|(<11294..11803>)
>DS571307|intron|48|join(<9252..9441,9489..9604>)
>DS571307|intron|complement|50|join(<7030..8367,8417..9097>)
>DS571308|intron|87|join(<6436..6636,6723..13895>)
>DS571308|intron|56|join(<6237..6380,6436..6636>)
>DS571309|intron|complement|51|join(<7504..8788,8839..8984>)
>DS571309|intron|63|join(<20733..21152,21215..22051>)
>DS571309|intron|53|join(<27939..28064,28117..29904>)
>DS571311|intron|68|join(<13474..13809,13877..14122>)
>DS571312|intron|41|join(<4830..4893,4934..5232>)
>DS571313|intron|complement|155|join(<12524..13397,13552..13685>)
>DS571315|intron|complement|79|join(<8752..10349,10428..11214>)
>DS571316|intron|complement|94|join(<1431..1759,1853..2312>)
>DS571316|intron|complement|50|join(<5076..5288,5338..5592>)
>DS571316|intron|55|join(<11893..12021,12076..12393>)
>DS571317|intron|49|join(<5681..5745,5794..6008>)
>DS571317|intron|complement|55|join(<2073..3964,4019..4322>)

>DS571317|intron|complement|61|join(<4019..4322,4383..4448>)
>DS571317|intron|163|join(<6237..6540,6703..7400>)
>DS571317|intron|56|join(<20850..20852,20908..21374>)
>DS571317|intron|67|join(<23754..23800,23867..24425>)
>DS571318|intron|72|join(<624..2015,2087..2275>)
>DS571318|intron|42|join(<9143..10157,10199..10725>)
>DS571318|intron|complement|62|join(<16010..16456,16518..16604>)
>DS571318|intron|complement|63|join(<16518..16604,16667..16852>)
>DS571319|intron|complement|80|join(<14686..14852,14932..15853>)
>DS571320|intron|complement|73|join(<13228..13807,13880..14685>)
>DS571320|intron|52|join(<27032..27293,27345..27445>)
>DS571321|intron|79|join(<6485..6612,6691..6852>)
>DS571321|intron|complement|65|join(<22974..23007,23072..23997>)
>DS571322|intron|complement|210|join(<10892..11443,11653..11769>)
>DS571322|intron|57|join(<15994..16797,16854..17774>)
>DS571324|intron|45|join(<18056..18229,18274..18663>)
>DS571326|intron|complement|65|join(<17349..18248,18313..18345>)
>DS571327|intron|complement|50|join(<11151..12106,12156..12287>)
>DS571327|intron|complement|44|join(<12428..13056,13100..13306>)
>DS571327|intron|81|join(<13768..13801,13882..15473>)
>DS571327|intron|116|join(<15544..15650,15766..18178>)
>DS571327|intron|85|join(<25115..25218,25303..25687>)
>DS571329|intron|60|join(<8757..8917,8977..9505>)
>DS571329|intron|57|join(<8377..8700,8757..8917>)
>DS571330|intron|complement|60|join(<16275..19568,19628..20056>)
>DS571330|intron|complement|55|join(<29731..29859,29914..30002>)
>DS571331|intron|complement|45|join(<2415..2760,2805..3019>)
>DS571332|intron|48|join(<8195..8209,8257..9093>)
>DS571332|intron|69|join(<14597..14695,14764..15258>)
>DS571332|intron|45|join(<15960..16061,16106..16954>)
>DS571332|intron|complement|56|join(<25671..25694,25750..26244>)
>DS571335|intron|complement|53|join(<24739..26040,26093..26125>)
>DS571335|intron|140|join(<27721..28099,28239..28936>)
>DS571337|intron|68|join(<20427..20735,20803..20906>)
>DS571337|intron|complement|78|join(<26633..26918,26996..27099>)
>DS571338|intron|complement|226|join(<3039..3511,3737..3929>)
>DS571339|intron|60|join(<5777..8840,8900..8967>)
>DS571340|intron|complement|24|join(<24755..25181,25205..25353>)
>DS571345|intron|63|join(<21485..22697,22760..23064>)
>DS571346|intron|50|join(<11400..11408,11458..11553>)
>DS571347|intron|complement|57|join(<5032..5753,5810..6128>)
>DS571347|intron|complement|90|join(<12004..12372,12462..12719>)
>DS571349|intron|complement|59|join(<8271..8863,8922..9119>)
>DS571351|intron|complement|60|join(<16984..17071,17131..18123>)
>DS571352|intron|50|join(<13383..13623,13673..14484>)
>DS571355|intron|complement|51|join(<6552..7032,7083..7711>)
>DS571356|intron|48|join(<16866..16971,17019..17242>)
>DS571356|intron|complement|124|join(<9489..10650,10774..12197>)
>DS571358|intron|71|join(<14112..14237,14308..15333>)
>DS571360|intron|complement|46|join(<24964..25197,25243..25317>)
>DS571362|intron|54|join(<16630..16768,16822..18506>)
>DS571368|intron|47|join(<13523..13745,13792..14114>)
>DS571373|gene|550|(<12739..13289>)
>DS571373|intron|70|join(<15416..19331,19401..19639>)
>DS571373|intron|complement|57|join(<20764..20980,21037..22952>)
>DS571375|intron|complement|140|join(<8741..10533,10673..11060>)
>DS571376|intron|complement|120|join(<21514..21748,21868..23252>)
>DS571377|intron|52|join(<12881..13610,13662..13801>)
>DS571378|intron|84|join(<3815..4045,4129..4248>)
>DS571382|intron|complement|190|join(<3466..4337,4527..5391>)
>DS571385|intron|47|join(<23092..23218,23265..24124>)
>DS571388|intron|complement|57|join(<958..1936,1993..2015>)
>DS571390|intron|complement|54|join(<16030..16246,16300..16427>)

>DS571391|intron|143|join(<11782..12184,12327..12634>)
>DS571392|intron|64|join(<2958..3120,3184..3917>)
>DS571395|intron|247|join(<7650..7812,8059..8237>)
>DS571397|intron|complement|109|join(<9868..9951,10060..10272>)
>DS571400|intron|complement|46|join(<16316..16444,16490..16954>)
>DS571400|intron|complement|65|join(<3047..4216,4281..4412>)
>DS571405|intron|63|join(<2335..2495,2558..3196>)
>DS571405|intron|complement|67|join(<5408..5443,5510..5887>)
>DS571408|intron|complement|42|join(<877..1223,1265..1943>)
>DS571410|intron|60|join(<12494..13174,13234..13404>)
>DS571414|intron|complement|50|join(<11690..15672,15722..15809>)
>DS571415|intron|complement|73|join(<14793..15694,15767..15848>)
>DS571416|intron|complement|49|join(<5573..5959,6008..6049>)
>DS571416|intron|complement|50|join(<7177..8084,8134..8347>)
>DS571416|intron|complement|62|join(<15861..16649,16711..16896>)
>DS571417|intron|56|join(<15535..15720,15776..16171>)
>DS571421|intron|complement|57|join(<6372..6742,6799..7420>)
>DS571421|intron|120|join(<19636..19889,20009..20039>)
>DS571422|intron|69|join(<6886..7182,7251..7556>)
>DS571423|intron|complement|245|join(<7961..8117,8362..9413>)
>DS571425|intron|complement|74|join(<200..716,790..890>)
>DS571425|intron|complement|50|join(<16891..17907,17957..18058>)
>DS571429|intron|59|join(<11281..11589,11648..12631>)
>DS571431|intron|54|join(<372..496,550..1162>)
>DS571431|intron|complement|58|join(<5054..6189,6247..6589>)
>DS571431|intron|complement|54|join(<10483..10941,10995..11246>)
>DS571431|intron|complement|56|join(<15047..15193,15249..15434>)
>DS571436|intron|complement|70|join(<10748..10876,10946..11054>)
>DS571439|intron|complement|49|join(<13100..13490,13539..13630>)
>DS571442|intron|54|join(<6268..6358,6412..7796>)
>DS571444|intron|53|join(<2556..2808,2861..3771>)
>DS571444|intron|212|join(<11713..12355,12567..13273>)
>DS571455|intron|complement|141|join(<6692..9104,9245..10053>)
>DS571459|intron|complement|55|join(<9510..9894,9949..10010>)
>DS571461|intron|complement|141|join(<5951..8363,8504..9408>)
>DS571463|intron|47|join(<12597..12692,12739..12942>)
>DS571466|intron|complement|52|join(<12229..12525,12577..13006>)
>DS571466|gene|549|(<15833..16382>)
>DS571466|intron|152|join(<15833..16201,16353..16382>)
>DS571468|intron|complement|50|join(<12587..12989,13039..13835>)
>DS571471|intron|complement|59|join(<1121..1998,2057..2084>)
>DS571481|intron|complement|107|join(<9160..9667,9774..9817>)
>DS571482|intron|complement|55|join(<934..1177,1232..1359>)
>DS571482|intron|complement|54|join(<8563..9064,9118..9230>)
>DS571487|intron|54|join(<9711..9716,9770..10432>)
>DS571488|intron|59|join(<5816..5962,6021..6389>)
>DS571490|intron|complement|71|join(<10321..10414,10485..11398>)
>DS571493|intron|50|join(<3578..3648,3698..4517>)
>DS571495|intron|53|join(<3444..3603,3656..5835>)
>DS571498|intron|complement|57|join(<1736..2540,2597..3006>)
>DS571506|intron|73|join(<33..127,200..362>)
>DS571506|intron|complement|93|join(<4115..4834,4927..5151>)
>DS571507|intron|65|join(<2448..2486,2551..3969>)
>DS571508|intron|148|join(<4870..4926,5074..5217>)
>DS571513|intron|64|join(<9403..9694,9758..9924>)
>DS571514|intron|complement|50|join(<1470..1872,1922..2718>)
>DS571522|intron|107|join(<5447..6010,6117..6143>)
>DS571528|intron|67|join(<1980..2544,2611..2785>)
>DS571528|intron|complement|42|join(<6040..6612,6654..6721>)
>DS571528|intron|complement|64|join(<7666..7954,8018..8276>)
>DS571534|intron|complement|71|join(<25..50,121..352>)
>DS571538|intron|64|join(<5625..6165,6229..7223>)
>DS571539|intron|complement|55|join(<789..917,972..1331>)

```
>DS571546|intron|complement|58|join(<6040..6662,6720..7044>)
>DS571557|intron|296|join(<9016..9264,9560..9802>)
>DS571561|intron|70|join(<3991..4328,4398..4564>)
>DS571561|intron|83|join(<2561..2651,2734..3845>)
>DS571572|intron|complement|57|join(<5812..5847,5904..6404>)
>DS571576|intron|complement|89|join(<54..73,162..897>)
>DS571587|intron|68|join(<4131..4229,4297..4614>)
>DS571587|intron|186|join(<6175..6372,6558..6824>)
>DS571588|intron|complement|50|join(<6643..6931,6981..6993>)
>DS571588|intron|complement|61|join(<7972..8346,8407..8808>)
>DS571591|intron|67|join(<2856..2996,3063..3503>)
>DS571594|intron|56|join(<3806..4186,4242..4493>)
>DS571600|intron|120|join(<8591..8946,9066..9090>)
>DS571602|intron|173|join(<4677..5302,5475..6063>)
>DS571618|intron|complement|63|join(<6157..6714,6777..6962>)
>DS571641|intron|50|join(<27..128,178..618>)
>DS571641|intron|complement|229|join(<5498..6073,6302..7153>)
>DS571645|intron|158|join(<403..413,571..1033>)
>DS571711|intron|complement|71|join(<1964..2584,2655..2886>)
>DS571718|intron|complement|56|join(<3283..3891,3947..3980>)
>DS571748|intron|complement|57|join(<423..495,552..692>)
>DS571762|intron|273|join(<72..998,1271..1429>)
>DS571764|intron|29|join(<2345..2878,2907..3374>)
>DS571771|intron|83|join(<3118..4092,4175..4219>)
>DS571778|intron|213|join(<1817..1902,2115..2583>)
>DS571799|intron|69|join(<1808..1922,1991..3390>)
>DS571800|intron|complement|42|join(<1719..2390,2432..2896>)
>DS571865|intron|complement|174|join(<882..1738,1912..2467>)
>DS571906|intron|73|join(<77..204,277..613>)
>DS571959|intron|105|join(<27..522,627..1162>)
>DS571970|intron|29|join(<298..314,343..1669>)
>DS572020|intron|complement|56|join(<32..90,146..1442>)
>DS572094|intron|complement|135|join(<1103..1580,1715..1746>)
>DS572240|intron|complement|89|join(<84..133,222..489>)
>DS572273|intron|complement|23|join(<25..33,56..439>)
>DS572327|intron|65|join(<157..1005,1070..1105>)
>DS572384|intron|81|join(<65..1068,1149..1188>)
>DsS572459|intron|129|join(<16..19,148..806>)
>DS572469|intron|62|join(<298..1031,1093..1138>)
>DS572471|intron|49|join(<17..559,608..781>)
>DS572532|intron|complement|47|join(<145..195,242..979>)
```

Appendix 2

WEKA Decision Tree j48 result:

Integration of models through j48 (WEKA):

WEKA's j48 gave following results and decision tree after 5 fold cross-validation.

```
Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     splice-site
Instances:    216
Attributes:   29
              donor_pssm_score
              acceptor_pssm_score
              donor_mkv1_score
              acceptor_mkv1_score
              donor_mkv2_score
              acceptor_mkv2_score
              donor_mkv3_score
              acceptor_mkv3_score
```

```
                dinuc_AA
                dinuc_AT
                dinuc_AG
                dinuc_AC
                dinuc_TA
                dinuc_TT
                dinuc_TG
                dinuc_TC
                dinuc_GA
                dinuc_GT
                dinuc_GG
                dinuc_GC
                dinuc_CA
                dinuc_CT
                dinuc_CG
                dinuc_CC
                mono_A
                mono_T
                mono_G
                mono_C
                pred
Test mode:    5-fold cross-validation
== Classifier model (full training set) ===

J48 pruned tree
------------------
Donor_pssm_score <= 3.00616
|   mono_G <= 5
|   |   acceptor_pssm_score <= 0.647167
|   |   |   dinuc_CA <= 3
|   |   |   |   dinuc_TG <= 0
|   |   |   |   |   mono_T <= 3: ACCEPTOR (3.0/1.0)
|   |   |   |   |   mono_T > 3
|   |   |   |   |   |   acceptor_pssm_score <= -0.925198
|   |   |   |   |   |   |   dinuc_GA <= 0
|   |   |   |   |   |   |   |   dinuc_GG <= 0
|   |   |   |   |   |   |   |   |   dinuc_GC <= 0
|   |   |   |   |   |   |   |   |   |   dinuc_TA <= 2: OTHER (4.0)
|   |   |   |   |   |   |   |   |   |   dinuc_TA > 2: DONOR (3.0)
|   |   |   |   |   |   |   |   |   dinuc_GC > 0: OTHER (7.0)
|   |   |   |   |   |   |   |   dinuc_GG > 0: DONOR (5.0/2.0)
|   |   |   |   |   |   |   dinuc_GA > 0: OTHER (17.0/1.0)
|   |   |   |   |   |   acceptor_pssm_score > -0.925198
|   |   |   |   |   |   |   dinuc_CG <= 0: DONOR (3.0/1.0)
|   |   |   |   |   |   |   dinuc_CG > 0: ACCEPTOR (4.0)
|   |   |   |   dinuc_TG > 0
|   |   |   |   |   dinuc_GG <= 0
|   |   |   |   |   |   dinuc_CG <= 1
|   |   |   |   |   |   |   dinuc_TA <= 3
|   |   |   |   |   |   |   |   dinuc_CT <= 4
|   |   |   |   |   |   |   |   |   mono_C <= 5
|   |   |   |   |   |   |   |   |   |   acceptor_pssm_score <= -6.598251: ACCEPTOR (2.0)
|   |   |   |   |   |   |   |   |   |   acceptor_pssm_score > -6.598251: DONOR (17.0/1.0)
|   |   |   |   |   |   |   |   |   mono_C > 5
|   |   |   |   |   |   |   |   |   |   acceptor_mkv2_score <= 197.25401
|   |   |   |   |   |   |   |   |   |   |   dinuc_CC <= 1: ACCEPTOR (6.0/1.0)
```

```
|   |   |   |   |   |   |   |   |   |   |   dinuc_CC > 1: OTHER (2.0)
|   |   |   |   |   |   |   |   |   |   acceptor_mkv2_score > 197.25401
|   |   |   |   |   |   |   |   |   |   |   dinuc_TA <= 1
|   |   |   |   |   |   |   |   |   |   |   |   donor_mkv1_score <= 43.596224: DONOR (3.0)
|   |   |   |   |   |   |   |   |   |   |   |   donor_mkv1_score > 43.596224: OTHER (2.0)
|   |   |   |   |   |   |   |   |   |   |   dinuc_TA > 1
|   |   |   |   |   |   |   |   |   |   |   |   mono_C <= 6: DONOR (4.0)
|   |   |   |   |   |   |   |   |   |   |   |   mono_C > 6: ACCEPTOR (2.0)
|   |   |   |   |   |   |   |   |   dinuc_CT > 4: OTHER (3.0/1.0)
|   |   |   |   |   |   |   |   dinuc_TA > 3: OTHER (3.0)
|   |   |   |   |   |   |   dinuc_CG > 1: ACCEPTOR (2.0)
|   |   |   |   |   |   dinuc_GG > 0
|   |   |   |   |   |   |   dinuc_TA <= 3
|   |   |   |   |   |   |   |   mono_T <= 5
|   |   |   |   |   |   |   |   |   acceptor_mkv1_score <= 18.801335: ACCEPTOR (3.0)
|   |   |   |   |   |   |   |   |   acceptor_mkv1_score > 18.801335: DONOR (4.0/1.0)
|   |   |   |   |   |   |   |   mono_T > 5
|   |   |   |   |   |   |   |   |   acceptor_mkv1_score <= 20.54916: OTHER (6.0)
|   |   |   |   |   |   |   |   |   acceptor_mkv1_score > 20.54916: ACCEPTOR (7.0/1.0)
|   |   |   |   |   |   |   dinuc_TA > 3: DONOR (2.0)
|   |   |   |   dinuc_CA > 3
|   |   |   |   |   dinuc_AG <= 0: OTHER (4.0)
|   |   |   |   |   dinuc_AG > 0
|   |   |   |   |   |   mono_G <= 1: OTHER (2.0)
|   |   |   |   |   |   mono_G > 1: ACCEPTOR (2.0)
|   |   |   acceptor_pssm_score > 0.647167
|   |   |   |   dinuc_GT <= 2
|   |   |   |   |   dinuc_TG <= 2
|   |   |   |   |   |   donor_pssm_score <= -0.578242
|   |   |   |   |   |   |   dinuc_GG <= 0
|   |   |   |   |   |   |   |   donor_pssm_score <= -0.953576
|   |   |   |   |   |   |   |   |   donor_mkv3_score <= 564.467837: DONOR (4.0/1.0)
|   |   |   |   |   |   |   |   |   donor_mkv3_score > 564.467837: ACCEPTOR (15.0)
|   |   |   |   |   |   |   |   donor_pssm_score > -0.953576: DONOR (2.0)
|   |   |   |   |   |   |   dinuc_GG > 0
|   |   |   |   |   |   |   |   dinuc_TA <= 1: ACCEPTOR (4.0)
|   |   |   |   |   |   |   |   dinuc_TA > 1: DONOR (4.0/1.0)
|   |   |   |   |   |   donor_pssm_score > -0.578242
|   |   |   |   |   |   |   mono_G <= 3
|   |   |   |   |   |   |   |   mono_A <= 6: OTHER (3.0)
|   |   |   |   |   |   |   |   mono_A > 6
|   |   |   |   |   |   |   |   |   dinuc_TG <= 0: ACCEPTOR (6.0)
|   |   |   |   |   |   |   |   |   dinuc_TG > 0: OTHER (3.0/1.0)
|   |   |   |   |   |   |   mono_G > 3: ACCEPTOR (6.0)
|   |   |   |   |   dinuc_TG > 2: OTHER (2.0)
|   |   |   |   dinuc_GT > 2
|   |   |   |   |   dinuc_TT <= 0: ACCEPTOR (2.0)
|   |   |   |   |   dinuc_TT > 0: OTHER (4.0)
|   |   mono_G > 5
|   |   |   dinuc_GT <= 1: DONOR (5.0/1.0)
|   |   |   dinuc_GT > 1: ACCEPTOR (3.0)
donor_pssm_score > 3.00616
|   dinuc_TT <= 1
|   |   donor_mkv3_score <= 640.075481: DONOR (7.0/1.0)
|   |   donor_mkv3_score > 640.075481: OTHER (4.0)
|   dinuc_TT > 1
```

```
|   |   dinuc_AT <= 1
|   |   |   donor_mkv2_score <= 195.824015: ACCEPTOR (3.0)
|   |   |   donor_mkv2_score > 195.824015: DONOR (4.0/1.0)
|   |   dinuc_AT > 1: DONOR (13.0)

Number of Leaves  :       46
Size of the tree :  91

Time taken to build model: 0.1 seconds

=== Predictions on test data ===
=== Summary ===
Correctly Classified Instances          85               39.3519 %
Incorrectly Classified Instances       131               60.6481 %
Kappa statistic                          0.0903
Mean absolute error                      0.4118
Root mean squared error                  0.6012
Relative absolute error                 92.642  %
Root relative squared error            127.5251 %
Total Number of Instances              216
```

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.39	0.33	0.37	0.39	0.38	0.52	DONOR
	0.42	0.26	0.44	0.42	0.43	0.57	ACCEPTOR
	0.38	0.31	0.38	0.38	0.38	0.54	OTHER
Weighted Avg.	0.39	0.3	0.4	0.39	0.39	0.541	

=== Confusion Matrix ===

a b c <-- classified as
28 20 24 | a = DONOR
21 30 21 | b = ACCEPTOR
27 18 27 | c = OTHER

Appendix 3
Atribute Relation File Format (".arff" files :- recommended WEKA input file format)

```
@relation splice-site
@attribute donor_pssm_score real
@attribute acceptor_pssm_score real
@attribute donor_mkv1_score real
@attribute acceptor_mkv1_score real
@attribute donor_mkv2_score real
@attribute acceptor_mkv2_score real
@attribute donor_mkv3_score real
@attribute acceptor_mkv3_score real
@attribute dinuc_AA real
@attribute dinuc_AT real
@attribute dinuc_AG real
@attribute dinuc_AC real
@attribute dinuc_TA real
@attribute dinuc_TT real
@attribute dinuc_TG real
@attribute dinuc_TC real
@attribute dinuc_GA real
@attribute dinuc_GT real
@attribute dinuc_GG real
@attribute dinuc_GC real
@attribute dinuc_CA real
@attribute dinuc_CT real
@attribute dinuc_CG real
@attribute dinuc_CC real
@attribute mono_A real
@attribute mono_T real
@attribute mono_G real
@attribute mono_C real
@attribute pred {DONOR, ACCEPTOR, OTHER}
@data
-0.721530572862108,-
4.90289121227968,35.66285620503,25.8419429452137,195.824015238913,193.778732508708,
409.776920231214,406.221705073887,3,2,1,1,2,4,0,3,0,1,1,0,2,1,0,0,6,12,1,2,DONOR

6.84036862006373,-
2.07012914827308,39.209610050365,18.262280724324,202.92905853085,203.943532552791,6
57.322974764783,659.4870744948,2,2,2,1,1,3,1,1,0,2,0,0,2,2,0,2,6,5,3,7,DONOR

.....
```