

Adaptive multi-agent architecture for functional sequence motifs recognition

Jia Zeng^{1*}, Reda Alhajj¹ and Douglas Demetrick²

¹Department of Computer Science, University of Calgary, Calgary, AB, Canada,

²Departments of Pathology & Laboratory Medicine, Oncology, Biochemistry & Molecular Biology and Medical Genetics, University of Calgary, Calgary, AB, Canada

Associate Editor: Prof. Limsoon Wong

ABSTRACT

Motivation: Accurate genome annotation or protein function prediction requires precise recognition of functional sequence motifs. Many computational motif prediction models have been proposed. Due to the complexity of the biological data, it may be desirable to apply an integrated approach that uses multiple models for analysis.

Results: In this paper, we propose a novel multi-agent architecture for the general purpose of functional sequence motif recognition. The approach takes advantage of the synergy provided by multiple agents through the employment of different agents equipped with distinctive problem solving skills and promotes the collaborations among them through decision maker (DM) agents that work as classifier ensembles. A genetic algorithm based fusion strategy is applied which offers evolutionary property to the DM agents. The consistency and robustness of the system are maintained by an evolvable agent that mediates the team of the ensemble agents. The combined effort of a recommendation system (Seer) and the self-learning mediator agent yields a successful identification of the most efficient agent deployment scheme at an early stage of the experimentation process, which has the potential of greatly reducing the computational cost of the system. Two concrete systems are constructed which aim at predicting two important sequence motifs – the translational initiation sites (TISs) and the core promoters. With the incorporation of three distinctive problem solver agents, the TIS predictor consistently outperforms most of the state-of-the-art approaches under investigation. Integrating three existing promoter predictors, our system is able to yield consistently good performance.

Availability: The program (MotifMAS) and the data sets are available upon request.

Contact: jzeng,alhajj,demetric@ucalgary.ca

1 INTRODUCTION

Thanks to large scale genomic sequencing efforts, the precise nucleotide sequence comprising the chromosomes of many organisms has been ascertained. Through improved protein sequencing methods, many functional protein sequences can be obtained, and compared to the genomic template. However, knowing the basic constituents of the object under investigation, be it a genome or a protein, is not sufficient in understanding its

biological significance. Inferring the precise locations and splicing patterns of genes in DNA is a challenging but important task, which involves the recognition of functional sequence motifs such as the *promoter*, the *terminator*, the *translation initiation site* (TIS), the *stop codon*, the *splice-junction site*, etc. Unfortunately, the problem of genome or protein annotation is plagued with ambiguity. Take gene annotation as a basic example. Most eukaryotic genes follow a pattern that begins with a start codon (ATG) and ends with a stop codon (one of TAG, TAA or TGA in most cases), and having embedded within them zero or more introns separated by donor sites (typically GT) at their 5' ends and acceptor site (usually AG) at their 3' ends. Within one DNA strand, it is highly likely that multiple occurrences of ATG, GT or AG exist. However, not all of them denote functional motifs.

We parallel this problem of accurately identifying the actual functional motifs from an ambiguous candidate set to the pattern recognition problem that has been extensively studied in the realms of computer science. As a matter of fact, up till today, several computational approaches have been proposed which intend to provide accurate gene annotation to genomic sequences by identifying the aforementioned functional sequence motifs. These approaches extract multiple features from the sequence data based upon the biological evidence that indicates their potential relevance and utilize theories from statistics and machine learning to process and analyze the acquired numerical or symbolic features with the purpose of recognizing the desired pattern, i.e., the functional motifs. A variety of statistical approaches have been explored. Examples include hidden Markov model (e.g., GeneMark.hmm by Lukashin and Borodovsky (1998), HMMgene by Krogh (2000), VEIL by Henderson *et al.* (1997), and UNVEIL by Majoros *et al.* (2003)), generalized hidden Markov model (e.g., TIGRscan by Majoros *et al.* (2004)), and linear discriminant analysis (LDA) (e.g., FGENEH by Solovyev *et al.* (1995)). Many machine learning approaches including neural network (e.g., GRAIL by Uberbacher and Mural (1991)), decision tree (e.g., GlimmerM by Pertea and Salzberg (2002)), support vector machine (e.g., Zien *et al.* (2000)), are extensively used in biomedical applications. Some of the computational approaches use a single feature to analyze the sequence data (e.g., Pedersen and Nielsen (1997)) whereas the most successful models apply multiple features to investigate the target and utilize some ensemble approach to yield the final prediction. For instance, GRAIL employs a neural network to integrate the

*to whom correspondence should be addressed

scores of seven content sensors, in order to predict the likelihood of a putative exon. For the purpose of identifying TIS in genomic, cDNA and mRNA sequences, Salamov *et al.* (1998) have used a *linear discriminant function* (LDF) to ensemble six different features, Hatzigeorgiou (2002) applied a sum rule to combine the output of two neural networks, each examining a distinctive feature of the sequence and Saeys *et al.* (2007) used the sum of the predictions offered by three primitive analysis approaches.

Evidently for any functional sequence motif, many different computational models have been proposed to solve the problem of identifying it. Some excel in examining one particular set of aspects of the data. Some may be good at others. Due to the complexity of the biological data under investigation and the state of our incomplete understanding of them, it may be desirable to apply an integrated approach that uses more than one feature/model to analyze the data. We believe that by incorporating multiple models, a more robust solution will emerge. However, most of the existing approaches either use a static ensemble strategy such as the plain sum rule or majority voting, or a black box method such as neural network or LDF, which may seem confounding to the end user. In particular, when linear discrimination function is applied to ensemble multiple features to arrive at a prediction, it suffers a major limitation in that LDF assumes all the features are independent from each other (Fisher (1936)), which is not applicable to the biological problems under investigation. In this paper, we propose a generalized multi-agent architecture called MotifMAS for the general purpose of identifying functional sequence motifs from genomic, cDNA, mRNA sequences and polypeptides. MotifMAS employs different agents equipped with distinctive problem solving skills and promotes collaborations among them through agents that work as classifier ensembles. A multi-agent system tends to provide a more diversified solution set compared to its centralized counterpart. It also does not rely solely on one model, thus avoiding a single point of failure. The consistency and robustness of the system are maintained by an evolvable agent that mediates the team of the ensemble agents. With the incorporation of genetic algorithm based ensemble strategy in decision making process and the mediator agent with the capacity of learning from experience, our system has presented a strong adaptive property. A statistical-based recommendation system is also applied which intends to identify an important heuristic — the smallest set of necessary agents at a very early stage of experimentation, with the purpose of greatly reducing unnecessary computational costs of the model.

Besides our proposed MotifMAS approach, the idea of MAS has been widely applied to solve a spectrum of bioinformatics problems. For instance, Lam *et al.* (2006) have proposed an MAS to assist gene expression analysis. Orro *et al.* (2005) devised an architecture called PACMAS for the purpose of predicting protein secondary structure. Vignal and Lisacek (1997) explored the application of MAS for exon prediction in human sequences.

2 APPROACH

According to the framework shown in Fig. 1, the MotifMAS architecture consists of five stages: solution generation, decision making, negotiation, execution and feedback. Four classes of agents participate in this entire process — *problem solver agents* (PS), *decision maker agents* (DM), *mediator agent* and *actuator*

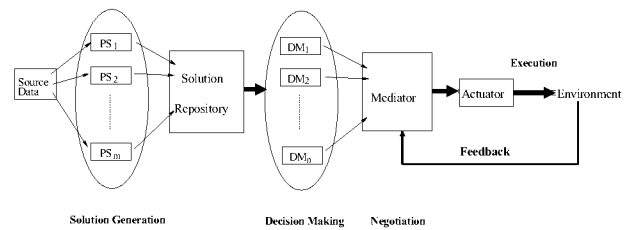


Fig. 1. General Architecture of MotifMAS

agent. Problem solver agents are the fundamental agents in the system, which originate a pool of solution candidates. Each of them acts as an independent domain expert which is specialized in solving the problem from a distinct biological perspective and outputs a solution candidate to the solution repository. During the stage of solution generation, the PS agents work independently instead of collaboratively. In order to take advantage of the synergy provided by multiple domain experts, the decision maker agents are introduced. Each of the DM agents employs a unique strategy to apply the outputs of PS agents. For instance, it can solely rely on one single PS agent's recommendation, or combine the input from multiple PS agents, or even consider some additional input from other sources. In the end, each DM agent will inform the mediator agent of its proposed decision. Since the DM agents may propose conflicting decisions, and there exists some variation among the data (i.e., the putative motifs), the mediator agent works as a negotiator that tries to identify the best DM agent to employ for a particular case. After one single final decision is yielded, the actuator agent is going to label the datum accordingly. During the training phase, the effect of this decision is back-propagated to the mediator agent which can then evolve over time.

2.1 Problem solver agent

Each problem solver agent is equipped with a unique set of domain expertise that helps it to provide a solution candidate to the particular biological problem under investigation. For instance, if the goal is to accurately locate the translational initiation sites within genomic, cDNA or mRNA sequences, a PS agent can be focused on investigating the favorability of the context of a putative start codon; or it can be trained to examine a putative open reading frame (ORF) in the light of protein secondary structure; it also may utilize the codon usage statistics that are observed in most of the genes of the target organism. In order to provide a general schema that is applicable for solving any functional sequence motif recognition problem, we ask each PS agent to yield a *problem solver message* (PSM) that conforms to a standard syntax: (AGENT-NAME: a , $a \in \{\text{the set of PS agents}\}$, CLASS: x , $x \in \{\text{True, False}\}$).

2.2 Genetic algorithm based decision maker agent

To offer a separation of PS agents and the subsequent layers of agents in the framework, a layer of decision maker agents is introduced. Some DM agents may solely rely on one PS agent's prediction, thus they output the CLASS prediction offered by the participating PS agent. Since each PS agent is designed to investigate the problem from a local view that matches its own

expertise, it may be beneficial to apply some decision maker agents that integrate several PS agents for the purpose of obtaining broader perspective of the problem. For these DM agents, we propose a scheme that applies the weighted sum rule as the ensemble strategy where the optimal weight assignment is acquired by using genetic algorithm. A quick review to the classical ensemble strategy is in order. Suppose there are n classifiers: C_1, C_2, \dots, C_n . Given a query datum q , each of them predicts q 's class membership by outputting a real value. Then the weighted sum of the classifier ensemble C_{Ens} can be computed using Eq. 1.

$$C_{Ens}(q) = \sum_{i=1}^n w_i \times C_i(q) \quad (1)$$

where w_i refers to the weight given to $C_i(q)$.

In our system, the PS agents are essentially classifiers and the DM agents that involve using more than one problem solver agent can be considered as classifier ensembles. To obtain the optimal weight vector and provide adaptiveness to the DM agents, we employ *genetic algorithm* (GA) to evolve the fittest weight assignment scheme. It is also believed that a classifier's prediction capacity on different classes may be unequal. Therefore, more weight can be given to class which the classifier has more confidence predicting.

A description of using GA in the problem solver agent ensemble process is in order. During the training phase, to arrive at the optimal weight assignment for decision maker agent DM: PS_1, PS_2, \dots, PS_n , a training data set has to be provided where for each datum D_i , the following vector is available: $(C_{PS_1}(D_i), C_{PS_2}(D_i), \dots, C_{PS_n}(D_i), C(D_i))$, where $C_{PS_j}(D_i)$ is PS_j 's class prediction on D_i and $C(D_i)$ is D_i 's actual class, in all cases, class predictions are represented by real numbers (e.g., 1 for *True* and -1 for *False*). First of all, some essential parameters for the genetic algorithm are initialized. A chromosome is comprised of $2n$ genes where for $i \in [1, n]$, the $(2i-1)$ -th gene corresponds to the weight assigned to a positive prediction by PS_i agent and $2i$ -th gene corresponds to the weight associated with a negative prediction by PS_i agent. Secondly, a random population is generated. A fitness function is applied to each chromosome in the population in order to estimate its fitness level. For any datum D_i , depending on the classes predicted by the participating PS agents, half of the genes contained in the chromosome will be used for calculating the weighted sum of the ensemble. When the numeric result has the same sign as the real number representing the actual class, the fitness of the chromosome is incremented. This procedure iterates through all of the data in the training set and final fitness value of the chromosomes can be determined. Subsequently, a reproduction operation should be conducted in order to generate a new mating generation and the strategy of weighted roulette wheel (as indicated in Goldberg (1989)) is used for imposing selection bias towards fitter individuals. Crossover operator is then applied on the pairs of selected parents in order to generate a new next generation. A mutation operation conforming to a given probability (e.g., 0.001) will be conducted to provide further diversification on the individuals of the new generation. In turn, their fitness values will be calculated and the best fitness will be recorded. Such a procedure is repeated until a termination condition is met (e.g., when the iteration counter reaches a pre-defined upper bound).

After the application of genetic algorithm during the training phase, the fittest chromosome can be obtained. Then for any participating PS agent, depending on its prediction (whether it is positive or negative), its corresponding weight would be used in calculating the overall sum. This total is compared to a threshold value to determine the query datum's class membership.

2.3 Mediator agent

To offer a good generality, our system employs an exhaustive strategy of constructing DM agents, i.e., given a problem solver agent set P , the DM agent set is the power set of P . Given multiple decision candidates offered by the DM agents, which may contain conflicting predictions, we propose two strategies to implement the mediator agent whose goal is to mediate the decisions of DM agents. A *naive mediator agent* relies on the evaluation using a validation (training) set and selects the DM agent with the best performance to predict any future testing data. The second solution is called the *incremental learner mediator agent*. During the training phase, the agent carries out three processes: database initialization, database update and optimal policy generation. In the first stage, all of the involved problem solver agents (say n of them) are considered and a key space that consists of n dimensions is constructed, where the i -th dimension represents a possible class prediction from the i -th PS agent. A record that fits into such a n -dimensional space is called a *problem solver message tuple* (PSM tuple). For example, in a system where three PS agents are used, the key space will contain three dimensions and there would exist 8 unique PSM tuples ranging from $(True, True, True)$ to $(False, False, False)$. For each distinct tuple in the key space, every DM agent's performance will be tracked in terms of its profitability whose value is originally set to be zero.

The database is updated every time when a datum q enters the system. For q , all of the participating problem solver agents would offer their proposed solutions accordingly. Therefore a corresponding PSM tuple t_q can be formed, which is compared to the already existing tuples in order to locate the one that is the closest to itself. Subsequently, all of the DM agents would be tried out on predicting q 's class membership and their corresponding effects would be credited through their profitability trackers. An accurate prediction yields X points credit whereas an inaccurate one brings no gain. To further take the cost of the PS agents' into consideration, each agent is assigned a different penalty whenever it is applied, i.e., the more computational costly agent will be assigned a bigger penalty than a more efficient counterpart.

After the training data have been exhausted, the database reaches a static point. The next task is to identify the optimal policy for each PSM tuple in the database simply by selecting the DM agent with the biggest profitability score. It is worth mentioning that during a situation when more than one DM agent have the same maximal score, a tie-breaking scheme needs to be employed. Our solution is to favor the DM agent that uses the least costly set of agents. At the end of this process, we can obtain a set of DM agents that have been considered as optimal in at least some of the cases. It is possible that this new DM agent set only requires the existence of some of the PS and/or meta agents. If this is the case, the absent PS agent(s) would be considered unnecessary, therefore the corresponding dimension(s) should be eliminated from the database's key space. As a consequence, the database is refined and a set of optimal policies can be finalized.

During the testing phase, only the necessary set of agents would be utilized to analyze a query datum. A corresponding PSM tuple can be acquired, which is then used to locate the best DM agent to employ in the set of optimal policies yielded during training. Finally, the mediator agent outputs the recommended DM agent's prediction as the system's final decision on the datum under investigation.

2.4 Recommendation system

In reality, most of the complete genomic, cDNA, mRNA and amino acid sequences are long, and usually contain a substantial number of putative motifs. Most of the existing computational approaches treat each potential motif as a datum. Therefore, the actual data collection size is approximated to be the number of sequences multiplied by a factor of the average number of putative motifs included in one sequence. Consider a three-fold cross validation process. Two thirds of the original data collections need to be used for training, which depending on the particular training algorithm may require a great deal of computational resources.

Therefore, we recognize the potential benefits of locating the smallest set of agents that are necessary for good performance at the earliest stage possible. The identification of such an agent set can be achieved by the deployment of the incremental learner agent. In order to obtain such information as early as possible, we incorporate a recommendation system called *Seer*, which was originally proposed by Zeng and Alhadjj (2007). The principle which *Seer* is based upon is to sample the original collection so as to obtain the smallest sample collection that is still able to sufficiently represent the original collection and conduct the experiments using the sample set in order to obtain the optimal parameter configuration scheme, which in our case, corresponds to the heuristic about the smallest necessary agent set.

3 EVALUATION OF THE SYSTEM

In order to evaluate the effectiveness and applicability of the proposed architecture, we have chosen two representative functional sequence motifs — the translational initiation site (TIS) and the core promoter and implemented concrete systems using MotifMAS architecture. They are named as MAS-TIS and MAS-CP respectively.

3.1 Case study I: MAS-TIS

As the name indicates, a TIS refers to the location where translation begins. Due to the characteristics of translational machinery, once a start site is precisely identified, it is trivial to locate the first in-frame stop codon and thus derive the corresponding protein sequence using the universal genetic code. The significance of TIS prediction is therefore self-explanatory.

The problem solver agents are the only type of agents that are specific to the application domain. In our system, three PS agents are used, including *context-based problem solver agent* (CPS), *downstream-based problem solver agent* (DPS) and *codon usage bias problem solver agent* (CUBPS). We also incorporated a meta agent called *upstream meta agent* (UM). Details of the construction of these agents can be seen in our previous publications (Zeng and Alhadjj (2007, 2008); Zeng et al. (2008).)

We conducted testing using three benchmark data sets — *vertebrates*, *Arabidopsis thaliana*, and *TIS+50*. The sequences from the first two data collections were constructed by Pedersen and Nielsen (1997) and were extracted from GenBank, release 95. All of the sequences have undergone preprocessing so that possible introns were removed and only the sequences that contain at least 10bp upstream of the TIS and at least 150bp downstream of the TIS were selected. The vertebrates group consists of sequences from *Bos taurus* (cow), *Gallus gallus* (chicken), *Homo sapiens* (human), *Mus musculus* (mouse), *Oryctolagus cuniculus* (rabbit), *Rattus norvegicus* (rat), *Sus scrofa* (pig), and *Xenopus laevis* (African clawed frog). The second data set contains sequences from *Arabidopsis thaliana* (thale cress, a dicot plant), which presents large deviation from vertebrates. Nadershahi et al. (2004)'s *TIS+50* data set contains 50 human *expressed sequence tags* (EST) sequences with complete ORFs. Our choice of using these data sets is justified as follows. Firstly, all of these collections have been used to test the effectiveness of more than one existing algorithm, especially the vertebrates data set, which has been cited in most of the related work. Secondly, the vertebrates and *Arabidopsis* collections only include conceptually-spliced mRNAs whereas *TIS+50* contains EST sequences that may contain errors resulting in frame shifts, and represent different parts of their parent cDNA. The difference between the two types of sequences provides some diversity to the testing process. In Nadershahi et al.'s paper, the authors had used *TIS+50* as the single testing set for evaluating a variety of state-of-the-art TIS predictors. This led us to believe that the data set is meritable. Lastly all of these three data sets are readily accessible from the Internet.

Though the *TIS+50* data set poses challenges to TIS predictors because of the nature of EST data, it is nonetheless a very small data set. In order to provide a more comprehensive investigation on the effectiveness of our approaches, we have constructed two larger synthetic data sets based on the characteristics presented by *TIS+50* — *TIS-1000* and *TIS-1500*. We conduct statistical analysis on *TIS+50*, to estimate the mean and standard deviation of the length of 5' UTR and the length of ORF respectively. We also apply the statistics including the codon usage frequency in upstream and downstream TIS as well as the nucleotide frequency around real TISs proposed by Ma et al. (2006). To construct a synthetic sequence, we obtain the length of 5' UTR by selecting a random number that conforms to a normal distribution given the corresponding sample estimates of mean and standard deviation. Similarly we can obtain a random assignment of the length of ORF. We then generate the major part of 5' UTR (from 5' end to position -11) by randomly selecting codons while considering upstream codon usage statistics in the meantime. Secondly, we utilize the statistics of nucleotide frequency around real TISs to obtain a random sequence whose positions fall into the range [-10, +8]. Then we complete the ORF by selecting codons with a random scheme that uses the downstream codon usage information. Similarly, 3' UTR is constructed¹.

The evaluation criteria we employed include recall, precision and F-measure. Due to the fact that there are disproportionately more false data than true data in TIS collections, recall and precision is

¹ Since 3' UTR is not very important for TIS prediction, we assume that it contains 30 bases for every sequence.

a more suitable pair of measures than sensitivity and specificity (although recall is equal to sensitivity). Since it is prudent not to discuss recall and precision in isolation, F-measure which is the weighted harmonic mean of these two metrics is used as the definitive criterion in the ensuing comparative study.

3.2 Comparative study

Due to the complexity of the translation start site selection process, accurate recognition of TIS in genomic, cDNA or mRNA sequences remains a challenging undertaking. A number of approaches that predict translational initiation sites have been proposed in the literature. In this study, we consider the following eight methods: Pedersen and Nielsen (1997)'s NetStart system, Burge and Karlin (1997)'s GENSCAN system, Saeys *et al.* (2007)'s StartScan system, Li and Jiang (2005)'s TISHunter program, Salzberg (1997)'s positional conditional probability approach, Zien *et al.* (2000)'s engineered SVM method, Zeng *et al.* (2002)'s system, and Liu *et al.* (2004)'s method.

The first four systems exist in the form of online servers, therefore complete results using all of the five data collections can be obtained, with the exception of TISHunter on *vert.* where the results from their original paper are quoted. In particular, GENSCAN is considered as one of the most successful *ab initio* coding sequence recognition approaches. Most of the remaining existing approaches report the performance of their systems using *vert.* collection alone.

In this series of experimentation (MAS-TIS), one of our major objectives is to evaluate the performance of the incremental learner mediator agent. Unless specified otherwise, the incremental learner is used as the mediator agent by default. In order to identify the smallest set of necessary agents at an early stage, we have requested *Seer* to provide the heuristic information. After one iteration of running *Seer*, the desired result becomes readily available. With the exception of *TIS+50* which uses all of the agents, for all of the other four data collections, CPS and UM are considered as redundant agents by the recommendation system. Therefore, they are both removed from the system before conducting the complete experiments on the original data collection. Due to the nature of the agent, CPS requires extensive computational resources. Thus the elimination of this agent yields a substantial reduction in the execution cost of the entire system.

We have conducted three-fold cross validation and reported the mean and standard deviation of the results of our system for each evaluation criterion. To provide a comprehensive evaluation of the incremental learner and *Seer*, we presented the results of the MAS-TIS system using the incremental learner mediator agent with *Seer*, denoted by MAS-TIS (w *Seer*) and that without *Seer*, denoted by MAS-TIS (w/o *Seer*) (i.e., incorporate all of the available problem solver agents throughout the training phase). We also reported the prediction performance of MAS-TIS using the naive mediator agent, denoted by MAS-TIS (Naive). Though the canonical MAS-TIS system advocates the benefits of using multiple decision maker agents each of which ensembles a set of problem solver agents, we also believe that it is important to present the results using only one single decision maker agent that combines all of the available problem solver agents using genetic algorithm sum rule for the purpose of justifying such a claim. Therefore, we have also included the data corresponding to the single DM version of MAS-TIS, which is referred to as MAS-TIS (IDM) in Table

1. For NetStart, GENSCAN, StartScan and TISHunter², all the results are obtained using the provided pre-trained models to predict the entire collections. Salzberg and Zien *et al.*'s systems are both tested with six-fold cross validation. Zeng *et al.* and Liu *et al.* have also used three-fold cross validation. Table 1 summarizes the results. From the data we can observe that our MAS-TIS system has outperformed most of the methods under investigation. TISHunter achieved comparable results to MAS-TIS but did not perform very well on *TIS+50*. The effectiveness and robustness of the proposed approach have been well demonstrated. To facilitate a better understanding of the data presented in the table, a more in-depth comparative analysis is in order.

Like most other statistical based approaches, Salzberg's method suffers from high false positive rate. Our MAS-TIS approach integrates a variety of algorithms that diversify the solution set. Although Pedersen and Nielsen were the pioneers of applying machine learning algorithm to solve TIS prediction problems, by considering only the context information around the ATG codons, their strategy has oversimplified the underlying biological process. We believe that the region that is upstream to a putative TIS should be considered differently from the one that is downstream to an ATG. As well, according to Kozak (1989)'s ribosomal scanning model, the relative position of an ATG in an mRNA sequence also plays a critical role in determining the fitness of a TIS candidate. Using a similar strategy as Pedersen and Nielsen's method, Zien *et al.*'s SVM prediction model also presents similar limitations. In MAS-TIS, we have incorporated a more versatile set of biological aspects, each investigated by an independent problem solver agent. Li and Jiang's TISHunter system uses an edit SVM kernel that incorporates a new measure of sequence similarity. This system has yielded very good results on four out of five data sets we used, with the exception of *TIS+50*. On the said four collections, TISHunter achieved results that are significantly better than MAS-TIS on *vert.* and *Arab.*, and that are comparable on *TIS-1000* and *TIS-1500*. However, our model significantly outperformed TISHunter on *TIS+50*. On average, MAS-TIS (w *Seer*) yields a F score of 0.8237, comparable to that of TISHunter's, which is 0.8289. Driven by the goal of finding a reasonable explanation of TISHunter's less-than-ideal performance on *TIS+50*, we have performed careful examination on the sequences in *TIS+50* on which TISHunter fails to accurately predict. This reveals that all of these sequences have a short 5' UTR whose length is less than 10 nt. We then experimentally verified all of the other four collections and confirmed that none of them contains any sequence of such property. TISHunter's author has also indicated on his server page that a minimum of 10 nt-long 5' UTR is required for accurate prediction using TISHunter. After confirming our speculation, we conducted an evaluation of our model on these sequences with short 5' UTR and MAS-TIS is able to accurately predict TIS for all of them. From these experimental data, we speculate that TISHunter's model relies on some built-in procedures that examine the TIS's upstream neighborhood of 10 nt or more. Generally speaking, this is a fair assumption. In fact, our CPS agent also examines an upstream context of the putative TISs. However, by diversifying the problem solving techniques with the use of a variety of PS agents, our MAS-TIS protocol manages to avoid

² with the exception of *vert.* for TISHunter

a single point of failure in a scenario where sequences' 5' UTR are shorter than assumed, therefore, demonstrating the robustness of the approach. Additionally, the success of TISHunter highly depends on the existence of related proteins or cDNA sequences in the database. Another major limitation of the method is that the kernel, once determined by a training set, cannot be easily adapted. Therefore, TISHunter lacks the flexibility and adaptability offered by MAS-TIS.

Some other existing approaches share this perspective of combining multiple models to solve the problem. For instance, Saeys *et al.*'s StartScan system applies a plain sum rule to combine the output from three independent classifiers, Zeng *et al.* and Liu *et al.* both use some machine learning algorithm that takes all of the relevant features as the feature set. Regardless of the differences in the feature set, these approaches share one common problem, i.e., the decision fusion strategy is static. In MAS-TIS, during the decision making process, we incorporate a genetic-algorithm based decision maker scheme that takes advantage of the adaptive property presented by the evolutionary computing scheme; in the negotiation process, each participating DM agent is evaluated based upon its potential profitability and depending on the outputs of the problem solver agents, different DM agents will be considered to be optimal ensemble strategy. Such a configuration provides a good flexibility and adaptability to the system as a whole. It is also worth mentioning that despite GENSCAN's success in common gene prediction, it is not solely designed to predict TIS. Its corresponding TIS predictions are yielded by applying an indirect procedure using the output from the program. This may explain its relatively unsatisfactory performance on some of the testing sets reported in this paper.

As far as the single DM version of MAS-TIS is concerned, it managed to achieve promising results. However, compared to its canonical counterpart, the variant still presents three major disadvantages. Firstly, in terms of the prediction accuracy, measured by the F score, MAS-TIS (1DM) has an average of 0.7834, which is still much lower than 0.8237, obtained by the canonical MAS-TIS (w Seer). Secondly, in some cases, e.g., on *TIS-1000* data set, MAS-TIS with single DM agent has shown significant variability in the results of the three-fold cross validation. This can be an indication of non-robust performance of the single DM strategy, which further demonstrates the necessity of using more than one DM agent in order to avoid a single point of failure. Additionally, in Table 2, we list the weight assignments evolved by the genetic algorithm based single decision maker agent (DM: CPS-DPS-CUBPS) on each data collection. Since cross validation is used, we report the result of each fold independently. Three tuples are included, representing the pair of weights related to the CPS, DPS and CUBPS respectively. It can be noted that there does not exist a consistent pattern which can describe the relative importance of the constituent PS agents. We believe that this may also indicate the underlying limitation of solely relying on one single ensemble. With the full MAS-TIS protocol, judging the merit of a DM agent on a case-by-case basis (i.e., by estimating its profitability for a particular problem solver message tuple), the mediator may select a variety of DM agents that are used during the testing phase. The selected set of DM agents may include solo DM agents that use only one PS agent and/or ensemble DM agents that apply other combination strategy, which may very well offset any instability brought by a less-than-ideal ensemble scheme. Our experiments have shown that a consistent pattern of DM agent selection by the mediator agent has been reserved during cross

validation. Thirdly, from the computational cost stand point, the MAS protocol is more advantageous than the single DM protocol (e.g., DM:CPS-DPS-CUBPS) in that the latter scheme has to use all of the available PS agents whereas the former one can avoid doing so.

Except for Zeng *et al.*'s approach where a feature selection scheme is employed, no other existing approaches discussed in this paper have integrated any heuristic component that facilitates the experimentation process such as the selection of the relevant features or the configuration of system parameters, e.g., threshold values. The incorporation of Seer in our approach provides an automatic mechanism of parameter configuration, thus greatly simplifies the experimentation procedure. Coupled with the mediator agent, Seer can also identify the smallest set of agents that are necessary and sufficient for providing an effective solution to the problem under investigation. Table 1 illustrates the prediction performance of using MAS-TIS protocol coupled with the incremental learner scheme with and without the Seer agent as well as the MAS-TIS with the naive mediator agent. In Table 3, we have reported the training run time of the three aforementioned MAS-TIS systems, together with the smallest necessary agent set identified by the recommendation system. In the case of MAS-TIS (w Seer), only select agents which are recommended by Seer are used. While Seer is not incorporated with MAS-TIS, the full set of agents (PS agents and meta agents) will be used in the training phase, and only the necessary ones will be used throughout the testing phase. It is worth mentioning that when the naive mediator agent is used, the MAS-TIS exhaustively uses every agent available, therefore, its training time is exactly the same as MAS-TIS (w/o Seer). We can observe that Seer suggested the elimination of the CPS and UM agents in all of the cases with the exception of *TIS+50*. In terms of prediction accuracy, MAS-TIS (w Seer) consistently yields results that are slightly better than or equal to that of MAS-TIS (w/o Seer), which demonstrates the effectiveness of Seer. A detailed discussion regarding the aforementioned results is in order. In the cases when Seer does not suggest the elimination of any agent (e.g. on *TIS+50*), MAS-TIS (w Seer) and MAS-TIS (w/o Seer) both take exactly the same set of PS agents (the full PS agent set) during the training phase and their results should be identical. However, when Seer does recommend the removal of certain agents, for the initial PS agent set, MAS-TIS (w Seer) only uses the select agents whereas MAS-TIS (w/o Seer) continues to use the full PS agent set. But with the incorporation of a sophisticated and adaptive incremental learner as the mediator agent during the training phase, these two systems are expected to arrive at two sets of optimal policies that are similar to each other (if not identical). In other words, even without the help of Seer, MAS-TIS (w/o Seer) should identify the redundant PS (and/or meta) agents at the end of the training phase and eliminate the use of them in the final optimal policy. Since the results are reported on the testing sets using the optimal policies yielded during the training phase, MAS-TIS systems with and without Seer should arrive at very similar or even identical results. It is worth pointing out that in MAS-TIS (w/o Seer), a key space containing more dimensions is used compared to MAS-TIS (w Seer), assuming that the PS agent sets used by these two schemes are different. This may result in the MAS-TIS (w/o Seer) scheme to encounter a more divisive key space compared to its counterpart, thus creating the slight difference in accuracy. The success of incorporating Seer agent in our MAS-TIS protocol

further confirms our hypothesis that not every agent is beneficial to the entire prediction process and therefore including all agents will not necessarily lead to better results. Due to CPS's distinct nature of being computational intensive, its absence in the system when the original data collection is used leads to substantial reduction in the execution time of the system as a whole. Take an extreme case for example, the MAS-TIS system would need more than six hours to complete its training on the *vert.* collection if all of the problem solver agents are used whereas it completes the same task within 19 minutes when the Seer's recommendation is taken into account. This has demonstrated another great advantage of incorporating Seer into the full MAS-TIS protocol. Though it is worth noting that for very small data set such as *TIS+50*, the advantage of using Seer diminishes.

Compared with MAS-TIS using the naive mediator agent (yielding an average F score of 0.7967), the MAS-TIS systems that use a more sophisticated model (i.e., the incremental learner) demonstrated much better prediction performance (average F scores: 0.8237 and 0.8158 for MAS-TIS with and without Seer respectively). With the help of Seer, the incremental learner coupled MAS-TIS also spent much less time for training (as shown in Table 3). The advantage of using the incremental learner over the naive scheme for the mediator agent is therefore evident.

3.3 Case Study II: MAS-CP

Transcription start sites signify the beginning of the transcription process. In order for transcription to take place, an enzyme that synthesizes RNA, called RNA polymerase, must attach to the regulatory region near the gene in question. These DNA sequences that promote transcription are called the promoters. In particular, core promoter refers to the minimal portion of the promoter that is required to properly initiate transcription. Accurate location of the core promoters helps elucidating the transcription-initiation process as well as enhancing the quality of genome annotation. We have employed three state-of-the-art core promoter predictors: EP3 by Abeel *et al.* (2008a), ProSOM by Abeel *et al.* (2008b) and Eponine by Down and Hubbard (2002) to serve as the problem solver agents and utilized the proposed architecture to ensemble the predictors' input. To avoid having overlap in training and testing data, we used human genome 17 (<http://hgdownload.cse.ucsc.edu/downloads.html>) chromosome 20 for training and chromosome 21 for testing.

The PS agents' predictions are yielded by using the default parameters set by the authors of the software. To facilitate the training and testing of MAS-CP, we have used a bin-based validation protocol 1A which is proposed by Abeel *et al.* (2009). This protocol uses the CAGE data set (<http://fantom.gsc.riken.go.jp/>) as reference. The chromosome is divided into bins of 500 nucleotides (nt). When a bin overlaps with the center of a *transcription start region* (TSR), then it is labeled as a positive TSR. Each bin that is both labeled by a prediction and a TSR is considered a *true positive* (TP). A *true negative* (TN) is one that is not labeled as predicted nor labeled as TSR. A *false positive* (FP) is a bin that is labeled as predicted but not labeled as TSR. Finally, a *false negative* (FN) is a bin that is not labeled as predicted but is labeled as TSR. The precision, recall and the F score can therefore be computed.

To have a more thorough evaluation of the MAS-CP system, we have experimented with four different ways to assign the PS agent

set: {EP3-Eponine}, {EP3-ProSOM}, {ProSOM-Eponine} and {EP3-ProSOM-Eponine}. We have applied the genetic algorithm to obtain the optimal weight assignment scheme for the weighted sum rule ensemble in DM agents and explored the use of both incremental learner and naive mediator agents. Table 4 illustrates the results, since no cross-validation is performed, the concept of standard deviation is not applicable. Whenever a MAS protocol is used, we denote our method by the mediator agent scheme followed by the set of PS agents that are used. For instance, Naive (EP3-ProSOM) uses the naive mediator agent in a system where only EP3 and ProSOM agents are used as the problem solver agents. In other words, three possible decision maker agents are considered: EP3, ProSOM and EP3-ProSOM. We can observe that, ProSOM is significantly more effective than its counterparts. When it is combined with other predictors, the results of MAS-CP's prediction are comparable to that using ProSOM alone. In the case of the naive mediator agent, the ensemble system can actually outperform ProSOM, although the improvement is insignificant. The major advantage of using MAS-CP is demonstrated by only using EP3 and Eponine in our system. With two weaker predictors integrated, the MAS-CP is able to perform significantly better than any of the participating predictors. Compared to the baseline set by the naive mediator agent, the incremental learner agent yielded similar results. However, as a more sophisticated model, the incremental learner mediator agent did not outperform the naive agent in this experiment. This is not very surprising in that the advantage of using incremental learner agent over its simpler counterpart mainly lies on the potential reduction of running redundant agents (if there are any). Especially, for such an application where all of the PS agents are external, it may be harder for the incremental learner mediator to evaluate the penalty setting for each individual PS agent. Therefore, we conclude that the naive mediator agent is also meritable in certain circumstances. In a nutshell, we believe that MAS-CP produces consistently good performance and can outperform its constituent PS agents in most cases. Though it is true that the current version of MAS-CP system does not present significant advantage over its single constituent counterpart, we still believe that generally our approach is beneficial because of the following three reasons: 1) it is a robust system which guarantees good results overall since it is not biased toward any particular feature of the data being analyzed; 2) it is a fully automated system that runs a comprehensive analysis of the possible agent deployment schemes and selects the most promising one; 3) as a general paradigm which offers a high level of abstraction, our system only has one layer of agents that are problem specific, thereby it can be easily applied to a spectrum of related application domains.

4 CONCLUSIONS

In this paper, we have presented a general multi-agent architecture for predicting functional sequence motifs in genomic, cDNA, mRNA and amino acid sequences. The application of the multi-agent paradigm takes advantage of the synergy provided by using more than one agent and the utilization of genetic algorithm and incremental learner mediator agent has offered more adaptive elements to the system, which is able to learn over its own experience and evolve over time. The incorporation of a statistical-based recommendation system has helped the experimenters to

identify the most efficient agent deployment scheme with minimal computational cost. The acquisition of such heuristic information greatly facilitates the application of the approach thanks to the time reduction it may yield. A concrete system using the MotifMAS architecture is constructed, which aims at predicting translational initiation sites in nucleotide sequences. Experiments on three benchmark and two synthetic data sets have shown the effectiveness and applicability of the proposed method, which consistently outperforms most of the existing approaches under investigation.

Within the general scheme of MotifMAS, the problem solver agents are the only class of actors that are application specific. Therefore, the proposed architecture can be readily applied to solve many more bioinformatics problems of the same nature. A number of computational models have been proposed to identify other types of gene signals such as promoters, terminators, splice-junction sites, and regulatory motifs including transcription factor binding sites, DNA protein binding sites, or even protein sequence motifs that are useful for predicting the protein function. The methods that are capable of yielding a prediction on any target motif can be viewed as problem solver agents and be easily integrated into the MotifMAS architecture. Their predictions will be further analyzed and processed by the decision maker agents whose output are mediated by the mediator agent. The layout of the paradigm provides a high level of abstraction and wide applicability. Therefore, the proposed architecture can be used on predicting any of the aforementioned functional sequence motifs. To justify this, we have also applied the proposed architecture to predict promoters in the human genome. Using three existing state-of-the-art predictors, our system can outperform the constituent predictors in most of the cases.

The TIS predictor (MAS-TIS) system presented in this paper is a major extension to our previous work (Zeng and Alhajj (2007, 2008)). A number of new elements have been proposed, including the incorporation of a simplistic yet effective new problem solver agent (CUBPS), the use of genetic algorithm in the ensemble process during decision making, the integration of a novel and adaptive mediator agent and the application of a statistical-based recommendation system which offers important heuristics to facilitate the experimentation process. The introduction of two synthetic data sets is also a novel contribution of this paper.

However, the current version of the system only considers the symbolic output from the participating problem solver agents, i.e., the predicted class label. It is observed that many existing models provide some intermediate results that indicate the likelihood of a putative datum as the target motif by a continuous value. It may also be beneficial to take this numerical output into consideration. As well, in this current stage of development, a weighted sum rule is used as the ensemble strategy in the decision making process. More intricate techniques such as Dempster-Shafer theory can be explored. In our future work, all of these topics will be further investigated.

ACKNOWLEDGMENTS

We are very grateful to Dr. Thomas Abeel for his detailed explanation of his papers as well as his willingness to share the validation data set and protocol method with us. We would like to thank all of the anonymous reviewers for their valuable and

constructive comments and suggestions. We also wish to thank the Associate Editor, Dr. Limsoon Wong for his encouragement. Salary support for DJD was provided by Calgary Laboratory Services.

REFERENCES

- Abeel, T., Saecys, Y., Bonnet, E., Rouze, P., and de Peer, Y. V. (2008a). Generic eukaryotic core promoter prediction using structural features of DNA. *Genome Research*, **18**, 310–323.
- Abeel, T., Saecys, Y., Rouze, P., and de Peer, Y. V. (2008b). ProSOM: core promoter prediction based on unsupervised clustering of DNA physical profiles. *Bioinformatics*, **24 ISMB 2008**, i24–i31.
- Abeel, T., de Peer, Y. V., and Saecys, Y. (2009). Towards a gold standard for promoter prediction evaluation. *Bioinformatics*. In press.
- Burge, C. and Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, **268**(1), 78–94.
- Down, T. and Hubbard, T. (2002). Computational detection and location of transcription start sites in mammalian genomic DNA. *Genome Research*, **12**, 458–461.
- Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, **7**, 179–188.
- Goldburg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley.
- Hatzigeorgiou, A. (2002). Translation initiation start prediction in human cDNAs with high accuracy. *Bioinformatics*, **18**(2), 343–350.
- Henderson, J., Salzberg, S., and Fasman, K. (1997). Finding genes in human DNA with a hidden Markov model. *J Comput Biol*, **4**, 127–141.
- Kozak, M. (1989). The scanning model for translation: an update. *Journal of Cell Biology*, **108**(2), 229–241.
- Krogh, A. (2000). Using database matches with HMMGene for automated gene detection in drosophila. *Genome Research*, **10**, 523–528.
- Lam, H., Garcia, M., Juncja, B., Fahrenkrug, S., and Bolcy, D. (2006). A multi-agent approach to gene expression analysis. In *Proceedings of the 2nd International Workshop on Multi-Agent Systems for Medicine, Computational Biology, and Bioinformatics*, pages 60–73, Hakodate, Japan.
- Li, H. and Jiang, T. (2005). A class of edit kernels for SVMs to predict translation initiation sites in eukaryotic mRNAs. *Journal of Computational Biology*, **12**(6), 702–718.
- Liu, H., Han, H., Li, J., and Wong, L. (2004). Using amino acid patterns to accurately predict translation initiation sites. *In Silico Biology*, **4**(3), 255–269.
- Lukashin, A. and Borodovsky, M. (1998). GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Research*, **26**(4), 1107–1115.
- Ma, C., Zhou, D., and Zhou, Y. (2006). Feature mining and integration for improving the prediction accuracy of translation initiation sites in eukaryotic mRNAs. In *Proceedings of the Fifth International Conference on Grid and Cooperative Computing Workshop*, pages 349–356.
- Majoros, W., Pertea, M., Antonescu, C., and Salzberg, S. (2003). GlimmerM, Exonomy and Unveil: three ab initio eukaryotic gene finders. *Nucleic Acids Res*, **31**, 3601–3604.
- Majoros, W., Pertea, M., and Salzberg, S. (2004). TIGRscan and glimmerHMM: two open source ab initio eukaryotic gene finders. *Bioinformatics*, **20**, 2878–2879.
- Nadershahi, A., Fahrenkrug, S., and Ellis, L. (2004). Comparison of computational methods for identifying translation initiation sites in EST data. *BMC Bioinformatics*, **5**(14).
- Orto, A., Saba, M., Vargiu, E., and Mancosu, G. (2005). Using a personalized, adaptive and cooperative multiagent system to predict protein secondary structure. In *Proceedings of the First International Workshop on Multi-Agent Systems for Medicine, Computational Biology, and Bioinformatics*, pages 170–183, Utrecht, Netherlands.
- Pedersen, A. and Nielsen, H. (1997). Neural network prediction of translation initiation sites in eukaryotes: Perspectives for EST and genome analysis. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, pages 226–233.
- Pertea, M. and Salzberg, S. (2002). Computational gene finding in plants. *Plant Mol Biol*, **48**, 48–49.
- Saecys, Y., Abeel, T., Degroev, S., and de Peer, Y. (2007). Translation initiation site prediction on a genomic scale: beauty in simplicity. *Bioinformatics*, **23 ISMB/ECCB 2007**, i418–i423.
- Salamov, A., Nishikawa, T., and Swindells, M. (1998). Assessing protein coding region integrity in cDNA sequencing projects. *Bioinformatics*, **14**(5), 384–390.

- Salzberg, S. (1997). A method for identifying splice sites and translational initiation sites in eukaryotic mRNA. *Computer Applications in the Biosciences*, **13**, 365–376.
- Solovyev, V., Salamov, A., and Lawrence, C. (1995). Identification of human gene structure using linear discriminant functions and dynamic programming. In *Proceedings of the 3rd International Conference on Intelligent Systems for Molecular Biology*, pages 367–375.
- Uberbacher, E. and Mural, R. (1991). Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. *Proc. Natl. Acad. Sci. USA*, **88**, 11261–11265.
- Vignal, L. and Lisacek, F. (1997). A multi-agent system for exon prediction in human sequences. *Genome Informatics*, **8**, 156–165.
- Zeng, F., Yap, R., and Wong, L. (2002). Using feature generation and feature selection for accurate prediction of translation initiation sites. *Genome Informatics*, **13**, 192–200.
- Zeng, J. and Alhajj, R. (2007). Multi-agent system in translation initiation site prediction. In *2007 IEEE International Conference on Bioinformatics and Biomedicine*, pages 103–108, Silicon Valley, US.
- Zeng, J. and Alhajj, R. (2008). Predicting translation initiation sites using a multi-agent architecture empowered with reinforcement learning. In *2008 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 241–248.
- Zeng, J., Alhajj, R., and Demetrick, D. (2008). The effectiveness of applying codon usage bias for translational initiation sites prediction. In *2008 IEEE International Conference on Bioinformatics and Biomedicine*, pages 121–126.
- Zien, A., Ratsch, G., Mika, S., Scholkopf, B., Lemmen, C., Smola, A., Lengauer, T., and Muller, K. (2000). Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, **16**(9), 799–807.

Table 1. Comparative Study on TIS Prediction (best F score is highlighted in boldface).

Data Set	Method	Recall	Precision	F
vert.	MAS-TIS (w Scer)	0.9574 ± 0.0412	0.9023 ± 0.0056	0.9290 ± 0.0213
	MAS-TIS (w/o Scer)	0.9623 ± 0.0451	0.8770 ± 0.0164	0.9167 ± 0.0134
	MAS-TIS (Naive)	0.9300 ± 0.0225	0.9210 ± 0.0165	0.9254 ± 0.0190
	MAS-TIS (1DM)	0.9300 ± 0.0225	0.9210 ± 0.0165	0.9254 ± 0.0190
	NetStart	0.8223	0.6867	0.7484
	GENSCAN	0.0023	0.0076	0.0035
	StartScan	0.7984	0.3835	0.5181
	TISHunter	0.9982	0.9975	0.9979
	Salzberg	0.6694	0.6694	0.6694
	Zien <i>et al.</i>	0.5482	0.9768	0.7023
	Zeng <i>et al.</i>	0.8847	0.8867	0.8857
	Liu <i>et al.</i>	0.8027	0.8790	0.8391
Arab.	MAS-TIS (w Scer)	0.9600 ± 0.0526	0.9044 ± 0.0127	0.9314 ± 0.0191
	MAS-TIS (w/o Scer)	0.9140 ± 0.0620	0.8998 ± 0.0454	0.9040 ± 0.0143
	MAS-TIS (Naive)	0.9751 ± 0.0109	0.9080 ± 0.0185	0.9402 ± 0.0095
	MAS-TIS (1DM)	0.9751 ± 0.0109	0.9080 ± 0.0185	0.9402 ± 0.0095
	NetStart	0.9733	0.7486	0.8463
	GENSCAN	0.0057	0.0179	0.0086
	StartScan	0.2446	0.2644	0.2541
	TISHunter	0.9732	0.9732	0.9732
TIS+50	MAS-TIS (w Scer)	0.8407 ± 0.0243	0.7584 ± 0.0582	0.7974 ± 0.0216
	MAS-TIS (w/o Scer)	0.8407 ± 0.0243	0.7584 ± 0.0582	0.7974 ± 0.0216
	MAS-TIS (Naive)	0.6189 ± 0.0140	0.8889 ± .0157	0.7238 ± 0.0636
	MAS-TIS (1DM)	0.6189 ± 0.0140	0.8208 ± 0.0615	0.6991 ± 0.0635
	NetStart	0.8801	0.2500	0.3894
	GENSCAN	0.6399	0.8208	0.7192
	StartScan	0.8744	0.1737	0.2898
	TISHunter	0.7000	0.7000	0.7000
TIS-1000	MAS-TIS (w Scer)	0.7340 ± 0.0188	0.7340 ± 0.0188	0.7340 ± 0.0188
	MAS-TIS (w/o Scer)	0.7340 ± 0.0188	0.7340 ± 0.0188	0.7340 ± 0.0188
	MAS-TIS (Naive)	0.6099 ± 0.1849	0.7867 ± 0.0651	0.6601 ± 0.1139
	MAS-TIS (1DM)	0.6398 ± 0.1983	0.7343 ± 0.0983	0.6492 ± 0.1022
	NetStart	0.7339	0.2701	0.3949
	GENSCAN	0.4939	0.7660	0.6006
	StartScan	0.1839	0.1856	0.1847
	TISHunter	0.7360	0.7360	0.7360
TIS-1500	MAS-TIS (w Scer)	0.7267 ± 0.0139	0.7267 ± 0.0139	0.7267 ± 0.0139
	MAS-TIS (w/o Scer)	0.7267 ± 0.0139	0.7267 ± 0.0139	0.7267 ± 0.0139
	MAS-TIS (Naive)	0.7340 ± 0.0214	0.7340 ± 0.0214	0.7340 ± 0.0214
	MAS-TIS (1DM)	0.7000 ± 0.0483	0.7077 ± 0.0035	0.7029 ± 0.0244
	NetStart	0.7188	0.2728	0.3955
	GENSCAN	0.4801	0.7560	0.5873
	StartScan	0.1799	0.1813	0.1806
	TISHunter	0.7373	0.7373	0.7373

Table 2. Weight Assignment Yielded by Genetic Algorithm Sum Rule (Normalization of weights: for each set of weights, using the first one as the basic unit, scale all the other weights accordingly.)

Data Set	Fold #	Weights
<i>vert.</i>	1	(1, 2.50), (3.86, 3.86), (2.41, 2.45)
	2	(1, 10), (3.85, 1.57), (12.43, 2.29)
	3	(1, 3.40), (3.96, 3.28), (0.20, 3.20)
<i>Arab.</i>	1	(1, 0.32), (0.93, 1.34), (0.08, 1.07)
	2	(1, 0.91), (2.20, 2.05), (0.5, 1.89)
	3	(1, 0.14), (0.82, 0.86), (0.07, 0.30)
<i>TIS+50</i>	1	(1, 0.84), (1.21, 1.37), (0.97, 0.96)
	2	(1, 1.91), (1.42, 0.82), (0.53, 1.44)
	3	(1, 1.16), (1.14, 0.09), (1.45, 0.24)
<i>TIS-1000</i>	1	(1, 0.60), (0.33, 0.01), (1.08, 0.60)
	2	(1, 1.60), (0.22, 0.22), (0.38, 0.93)
	3	(1, 0.63), (0.13, 0.81), (1.81, 0.46)
<i>TIS-1500</i>	1	(1, 3.20), (8.70, 7.50), (7.20, 7.70)
	2	(1, 0.15), (1.15, 0.08), (1.11, 1.18)
	3	(1, 0.32), (0.88, 0.34), (0.90, 0.83)

Table 3. MAS-TIS training time

Data Set	MAS-TIS (w Seer)	MAS-TIS (w/o Seer) or MAS-TIS (Naive)	Seer Recommendation
<i>vert.</i>	18m41s	6h1m48s	CUBPS, DPS
<i>Arab.</i>	3m22s	6m30s	CUBPS, DPS
<i>TIS+50</i>	5m22s	2m58s	CUBPS, CPS, DPS, UM
<i>TIS-1000</i>	12m6s	2h23m11s	CUBPS, DPS
<i>TIS-1500</i>	19m2s	4h52m15s	CUBPS, DPS

Table 4. Comparative Study on Core Promoter Prediction

Method	Recall	Precision	F
EP3	0.1832	0.3490	0.2402
ProSOM	0.2418	0.3734	0.2935
Eponine	0.1484	0.5786	0.2362
Naive (EP3-ProSOM)	0.2418	0.3734	0.2935
Naive (EP3-Eponine)	0.2134	0.3530	0.2660
Naive (ProSOM-Eponine)	0.2473	0.3595	0.2930
Naive (EP3-ProSOM-Eponine)	0.2445	0.3688	0.2941
Incremental (EP3-ProSOM)	0.2418	0.3734	0.2935
Incremental (EP3-Eponine)	0.2134	0.3530	0.2660
Incremental (ProSOM-Eponine)	0.2473	0.3595	0.2930
Incremental (EP3-ProSOM-Eponine)	0.2473	0.3595	0.2930