Reinforcement Learning for Improving Gene Identification Accuracy by Combination of Gene-Finding Programs

使用增強學習法結合不同基因搜尋程式以提高基因識別之準確率

尹邦嚴	徐熊健	楊世任	張猷忠
國立暨南大學資管所	銘傳大學資工所	銘傳大學資工所	銘傳大學生物科技系
pyyin@ncnu.edu.tw	sjshyu@mcu.edu.tw	yang5@pchome.com.tw	d80106@mcu.edu.tw

Abstract

A great number of gene-finding programs have been developed for annotating newly sequenced DNA genomes. However, none of them have consistent performance over various species. Recently, a decision fusion concept that improves the prediction accuracy by combining the predictions obtained by multiple gene-finding programs has been raised. The existing combination methods are relatively ad-hoc or lack intensive experiments. In this paper, we propose a new combination method based on reinforcement learning which learns from history predictions obtained by existing gene-finding programs and derives the optimal policy for selecting the best prediction program at each nucleotide. The experimental results manifest that the proposed method can significantly improves the performance compared to the single best program.

Keywords: Bioinformatics; DNA sequence; Gene identification; Reinforcement learning.

摘要

為了註解新定序的基因體 DNA，已經有許多基因搜尋程式被發展出來，但是沒有一個程式能對各種不同物種表現一致的效能。近來，有人提出結合數個基因搜尋程式的預測以提高準確度的決策融合觀念，但是目前已提出的結合方法卻都相當的直觀或者缺乏大量的實驗。在這篇論文當中，我們利用增強學習法提出一個新的結合方法，它可以學習已存在的基因搜尋程式過去預測基因的歷史紀錄，進而推論出可以在每一個核苷酸位置挑選出最適合的基因搜尋程式的最佳策略。實驗結果顯示我們提出的方法與最佳的單一基因搜尋程式相比，可以顯著提高基因預測的效能。

關鍵詞：生物資訊、去氧核糖核酸序列、基因識別、增強學習法。

1. Introduction

There is an explosive growth in the amount of sequenced nucleotides of genomic DNA due to the newly developed biotechnology and various genome projects. Several million bases of genomic DNA are sequenced daily and made available to the public. It becomes crucial to analyze the data and characterize sequence content in a high-throughput computational way. To date, many gene-finding programs have been developed to annotate the newly sequenced genomes. This is an essential and important step in the works of genome annotation. Those programs are based on pattern recognition methods such as artificial neural networks [GRAIL (Xu *et al*., 1997), GeneParser (Snyder and Stormo, 1995)], discriminant analysis [GeneFinder (Solovyev *et al*., 1994), MZEF (Zhang, 1997)], and hidden Markov models [Genie (Kulp *et al*., 1996), GENSCAN (Burge and Karlin, 1997), HMMgene (Krogh, 1997)].

Although these gene-finding programs have reported high prediction accuracy in specific domains, we still lack a universal program that can report satisfactory accuracy in general cases. Researchers thus further verify the predictions of these programs by searching for similar homologues in the database, however, it has been already known that about 50% of newly discovered genes have no similar homologues in the protein sequence database (Uberbacher *et al*., 1996; Dunham *et al*., 1999). As such, improving the prediction accuracy of gene-finding programs is more important than the validation task thereafter. Moreover, most researchers strive to develop a new gene-finding program that can attain better prediction accuracy than the others, they ignore the fact that a gene-finding method may yield highly accurate predictions in a specific domain, but there is no single gene-finding approach which is the most appropriate gene predictor for all newly sequenced genomes. Even if a worse gene-finding program can correct part of the

predictions produced by a novel gene predictor. For instance, when we annotate a 28,984 bp-long contig of human DNA sequence into exon and intron regions, GENSCAN can correctly identify gene structures from 28,430 bp of them. Among the 554 bp remaining sequences that are not well annotated by GENSCAN, more gene structures can be recognized by GeneView (366 bp) and HMMGene (454 bp), respectively. This reveals a good compensation among those programs in gene prediction. Therefore, a careful combination of multiple gene-finding programs is very likely to overcome any individual program.

In this paper, a reinforcement learning model (RLM) for combining three gene-finding programs is presented. The predictions produced by GENSCAN, HMMgene, and GeneView on an annotated HMR195 genomic dataset (http://www.cs.ubc.ca/labs/beta/genefinding/) are used to train the RLM network. The results manifest that the RLM learns the optimal policy which can determine the best gene-finding program at a given nucleotide site to maximize the expected gene identification accuracy over the whole genome, thus the synergism among those programs is achieved.

2. Methods

2.1 Combining Gene Predictions

There already exist some methods which combine the predictions by several gene-finding programs. GeneNomi (Harris, 1997) combines several sources of genomic analysis tools, such as BLAST, GRAIL, GeneFinder, Genie, etc., to make better predictions, however, the details for combining those tools are not described. Murakami and Takagi (1998) proposed five combination methods, namely AND, OR, HIGHEST, RULE, and BOUNDARY methods, to integrate the predictions by FEXH, GeneParser3, GENSCAN, and GRAIL2. The AND method labels the exon candidates as the intersection of the predicted exon regions by the programs, while the OR method determines the exon regions as the union of the program-predicted exons. As for the HIGHEST method, the exon candidates are those regions which have the highest score among the programs. The RULE method determines the predictions in accordance with a priority order of the programs based on a previous empirical study. Finally, the BOUNDARY method determines which are the best coding-/noncoding boundaries based on the score and boundary type given by the programs. The five combination methods are simple and ad-hoc, and cannot accommodate the correlations between programs and adjacent nucleotides. For instance, given the predictions (exon or non-exon) by three programs on two adjacent nucleotides there are in total 64 possible combinations, however, most of the combinations are not differentiated by those methods. Rogic *et al*. (2002) proposed three methods for combining predictions by GENSCAN and HMMgene. They focused on improving exon level accuracy by union or intersection of predicted exon regions considering probabilistic scores and reading frame consistency. The accuracy improvement on a newly assembled dataset is 7.9% over the single best program. Nevertheless, these methods are also rule-based and are not able to model complex correlations among programs and adjacent nucleotides. Pavlovic *et al*. (2002) provided a full Bayesian framework and adopted the hidden input/output Markov models for combining gene-predictions produced by a set of program experts. The prior observations on the predictions by the programs can be used to train the Bayesian network which models the correlations between programs and adjacent nucleotides. The authors claimed that the probabilistic model can significantly improve the prediction accuracy over a single best program, however, only one annotated drosophila sequence was managed to testify their method.

2.2 Reinforcement Learning Model

The reinforcement learning model (RLM) is broadly used in the machine learning community and has exhibited many successful applications [Kaelbling and Moore (1996), Mitchell (1997), Peng and Bhanu (1998)]. The RLM addresses the issue of how a simple agent can learn a task through many trial-and-error interactions with its environment. The agent senses the current state of its environment then makes a decision of choosing an action to perform. The state of the environment is therefore, activated by the agent's action, changed to another state, and the agent will receive a scalar reward regarding the desirability of the state transition. The process is repeated until the agent has learned an optimal policy that maximizes the accumulative reward received over time.

The RLM can model the task for learning to combine predictions of gene-finding programs as depicted in Figure 1. A set of annotated nucleotide sequences (the environment) is used to train the RLM network. The agent serving as the combiner which can observe the recent history predictions (the state) by the programs and makes a decision regarding which current prediction (the action) of those experts to perform at the present nucleotide site. Then, the

environment returns a binary indicator (the reward) to the agent about the precision of the performed prediction, i.e., whether the performed prediction matches the annotation. The environmental state is updated by adding the current predictions to the recent history predictions. Therefore, the agent learns the optimal policy which maximizes the expected sum of precisions attained at each nucleotide site.

Three gene-finding programs, namely GENSCAN, HMMGene, and GeneView, are combined by the RLM where the former two programs had been reported to have high prediction accuracy and the latter was found to be able to complement the other two programs in our early experiments. Let us denote the three programs by e_1, e_2, and e_3, and their prediction decisions at base t by $g_{e_1}^t$, $g_{e_2}^t$, and $g_{e_3}^t$, respectively. The principal elements of the RLM for combining predictions of gene-finding programs are characterized in the following.

(1) Environmental state, $s \in S$, where S denotes the set of the environmental states. The state should involve the factors that would affect the learning task. We propose two presentation schemes, namely RLM-1 and RLM-2, for describing the environmental states. The RLM-1 intends to model the correlation of gene-finding programs by looking into the adopted expert program at base $t-1$ and the predictions made by all the programs at bases t. Additionally, the probability scores produced by GENSCAN and HMMGene are taken into account to determine the confidence level since they have already been found to be useful in previous studies (Murakami and Takagi, 1998; Rogic *et al.*, 2002). In practice, the probability scores can be quantified into a fixed number of discrete levels. As such, the environmental state can be described by $s = [\, e^{t-1}, g_{e_1}^t, g_{e_2}^t, g_{e_3}^t, p_{GENSCAN}^t, p_{HMMGene}^t \,]$, where e^{t-1} is the expert program adopted at base $t-1$ and $p_{GENSCAN}^t$ and $p_{HMMGene}^t$ are the confidence levels of the probability scores produced by GENSCAN and HMMGene at base t. In addition to the factors adopted by RLM-1, the RLM-2 further considers the dependence between the predictions at the adjacent nucleotides and adds into the state descriptor the predictions made by all the programs at base $t-1$. Thus, the environmental state employed by RLM-2 is described by $s = [\, e^{t-1}, g_{e_1}^{t-1}, g_{e_2}^{t-1}, g_{e_3}^{t-1}, g_{e_1}^t, g_{e_2}^t, g_{e_3}^t, p_{GENSCAN}^t, p_{HMMGene}^t \,]$. The comparative performance of the two state presentation schemes will be given in the result section.

(2) Agent action, $a \in A$, where A denotes the set of actions to be performed by the agent. The agent action reflects the decision regarding which expert program and the corresponding prediction is adopted at base t. Therefore, the action is characterized by $a = [\, e^t, g_{e^t}^t \,]$.

(3) Scalar reward, $r \in R$, where R denotes the set of scalar rewards which represent the desirability about the currently adopted prediction. We use a binary reward, i.e. $R = \{0, 1\}$ and let $r = 1$ if the adopted prediction matches the annotation, and $r = 0$ otherwise.

(4) A state transition function, $\delta : S \times A \to S$. The state transition function determines the next state which is triggered by a performed action at the preceding state. By the afore-mentioned definitions, we get

$$\delta_{RLM-1}\left(\left(e^{t-1}, g_{e_1}^t, g_{e_2}^t, g_{e_3}^t, p_{GENSCAN}^t, p_{HMMGene}^t\right), \left(e^t, g_{e^t}^t\right)\right) = \left(e^t, g_{e_1}^{t+1}, g_{e_2}^{t+1}, g_{e_3}^{t+1}, p_{GENSCAN}^{t+1}, p_{HMMGene}^{t+1}\right) \tag{1}$$

and

$$\delta_{RLM-2}((e^{t-1}, g_{e_1}^{t-1}, g_{e_2}^{t-1}, g_{e_3}^{t-1}, g_{e_1}^t, g_{e_2}^t, g_{e_3}^t, p_{GENSCAN}^t, p_{HMMGene}^t), (e^t, g_{e^t}^t)) = \left(e^t, g_{e_1}^t, g_{e_2}^t, g_{e_3}^t, g_{e_1}^{t+1}, g_{e_2}^{t+1}, g_{e_3}^{t+1}, p_{GENSCAN}^{t+1}, p_{HMMGene}^{t+1}\right) \tag{2}$$

In essence, the agent makes a decision at current site and moves forward to collect new evidence, thus the agent learns from collective experience.

The optimal policy, $\pi^* : S \to A$, learned by the agent maximizes the prediction accuracy over all of the training sequences, that is,

$$\pi^* = \arg\max_{\pi} \{r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots\} = \arg\max_{\pi} \sum_{t=0}^{\infty} \gamma^t r_t \tag{3}$$

where r_t is the accuracy reward received at base t using the policy π to select experts and predictions, $\gamma \in [0,1]$ indicates the discounting factor that determines the relative weight of r_t. Let $Q(s,a)$ be the maximum overall prediction accuracy which can be received by performing action a in state s and

then proceeding optimally using π^*. The optimal policy can be learned by an iterative *Q*-learning algorithm (Kaelbling and Moore, 1996) through the following recursive definition of the *Q* function,

$$Q(s, a) = r + \gamma \max_{a'} Q(\delta(s, a), a'). \quad (4)$$

The implementation of *Q*-learning algorithm for combining gene-finding programs is described as follows. First, the agent initializes a table of the estimate of the *Q* function for each possible state-action pair. Then, the annotations about a training set of genomic DNA sequences and the predictions by the gene-finding programs on the same training set are fed into the agent at the sequential base order for training the *Q* function. The agent senses the current state and performs an action according to the *action selection rule* as will be described further. The state thus is transited to a new one and a reward regarding to the prediction accuracy of the current base is determined. The corresponding *Q* table entry is accordingly updated by the new state and the received reward using Equation (3). The algorithm is iterated until the agent experiences all training sequences and an approximate *Q* function is obtained. Then the agent is able to use the approximate *Q* function to predict the genes of newly sequenced genomes.

The action selection rule is a search heuristic which controls the relative importance between *exploration* and *exploitation* of the policy space. Exploration search focuses on the selection of the actions that have not been performed yet in the hope that new and better policy can be found. On the other hand, exploitation search utilizes the previous experience (the *Q* function estimate) about the desirability of choosing the actions and favors the one that will yield higher reward. Based on these guidelines, we propose a thresholded maximum selection rule which performs with a probability threshold the action leading to the maximal *Q* function estimate, otherwise a random action is drawn. The thresholded maximum selection rule oscillates between the exploitation for the best-so-far action and the exploration of new actions. It has been shown in our previous study (Yin, 2002) that the thresholded maximum selection rule overcomes many others involving linear, quadratic, exponential weighting rules, etc.

The proposed RLM for the combination of gene-finding programs can capture the correlation among programs and the dependence between adjacent nucleotides. The *Q*-learning algorithm is simple and can be easily implemented but it is still powerful in learning the optimal policy which gives the highest prediction accuracy based on combination of multiple sources.

3. Results

To quantify the prediction performance, two commonly used measures, namely the *sensitivity* and *specificity*, are adopted and defined as

$$Sn = \frac{TP}{TP + FN} \quad (5)$$

and

$$Sp = \frac{TP}{TP + FP}, \quad (6)$$

where TP (true positive) is the number of nucleotides that are correctly predicted as exons, FN (false negative) is the number of nucleotides that are predicted as introns while their groundtruth annotations are exons, and FP (false positive) is the number of nucleotides that are labeled as exons even though they are actually part of introns. In addition to measuring the prediction accuracy at nucleotide base level, sensitivity and specificity can also be calculated at exon levels and will be referred to as *ESn* and *ESp*, respectively. However, each of sensitivity and specificity cannot be used alone since perfect sensitivity of 1 can be obtained if all the nucleotides are predicted as coding region, and perfect specificity can be obtained if all the nucleotides are predicted as noncoding region. A unified measure named *correlation coefficient* (*CC*) has been proposed and intensively used for evaluating gene-finding programs (Burset and Guigo, 1996; Rogic *et al*., 2001). It is defined as

$$CC = \frac{TP \cdot TN - FN \cdot FP}{\sqrt{(TP + FN) \cdot (TN + FP) \cdot (TP + FP) \cdot (TN + FN)}}. \quad (7)$$

But it is undefined if no nucleotides are predicted as coding region. Thus, Burset and Guigo (1996) introduced the *approximate correlation* (*AC*) as

$$AC = 2(ACP - 0.5), \quad (8)$$

where

$$ACP = \frac{1}{4}\left(\frac{TP}{TP+FN} + \frac{TP}{TP+FP} + \frac{TN}{TN+FP} + \frac{TN}{TN+FN}\right), \quad (9)$$

which is defined under any circumstances.

The testing human genomic sequences for performance evaluation are those in the HMR195 genomic dataset which contains 195 sequences of human, mouse, and rat (http://www.cs.ubc.ca/labs/beta/genefinding/). The mean length of the sequences is 7096 bp, and the number of exons per gene is 4.86. Sixty randomly selected sequences from this set is used as training sequences and the rest are for testing. Table 1 shows the average performances obtained at both the base level and the exon level by the three gene-finding programs, the AND and OR combination methods, and the proposed RLM algorithms. It is observed that GENSCAN has the best performance among the three gene-finding programs as we expected. The AND method is strictly conservative and has a very low sensitivity value while the OR method is plagued with over-predictions about the exons and fails to improve the specificity. Both the two combination rules do not deliver better predictions as compared to those by GENSCAN. On the other hand, the RLM-1 method which models the correlations between expert programs does improve the prediction accuracy at the base level compared to GENSCAN. The RLM-2 method accommodating the correlations between programs as well as the dependence between adjacent nucleotides seems to has no advantage over RLM-1. We believe that it is because we let both RLM-1 and RLM-2 be trained for the same number of nucleotides, the performance of RLM-2 should be superior if we trained RLM-2 with more sequences since RLM-2 has a more complex probabilistic network. We notice that neither RLM-1 nor RLM-2 exhibits significant improvement on the exon level performance since they do not utilize exon-level-information such as exon boundary type and reading frame consistency, which will be considered in our future research.

4. Conclusions

In this paper, we have proposed a reinforcement learning approach for improving the gene identification accuracy by combination of gene-finding programs. The correlation of the programs and the dependence between adjacent nucleotides are suitably modeled and the proposed method can learn from history predictions of several gene-finding programs and derives the optimal policy that determines the best program to make annotation of a given nucleotide. The HMR195 dataset has been used to testify our method. The experimental results reveal that our combination method outperforms the single best gene-finding program and some of the existing combination methods. This will be very useful in the recent works of genome annotation, because it can improve the accuracy of prediction of gene structure in substance.

References

[1] C. Burge and S. Karlin, “Prediction of complete gene structures in human genomic DNA,” *Journal of Molecular Biology*, vol. 268, pp. 78-94, 1997.

[2] M. Burset and R. Guigo, “Evaluation of gene structure prediction programs,” *Genomics*, vol. 34, pp. 353-367, 1996.

[3] I. Dunham, N. Shimizu, B. Roe, and S. Chissoe, “The DNA sequence of human chromosome 22,” *Nature*, vol. 402, pp. 489-495, 1999.

[4] N. L. Harris, “Genotator: a workbench for sequence annotation,” *Genome Research*, vol. 7, pp. 754-762, 1997.

[5] L. P. Kaelbling and A. W. Moore, “Reinforcement learning: a survey,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 237-285, 1996.

[6] A. Krogh, “Two methods for improving performance of an HMM and their application for gene finding,” in *Proc. Fifth International Conference on Intelligent System for Molecular Biology*, AAAI Press, Menlo Park, CA, pp. 179-186, 1997.

[7] D. Kulp, D. Haussler, M. Reese, and F. Eeckman, “Integrating database homology in a probabilistic gene structure model,” in *Proc. Pacific Symposium on Biocomputing*, Hawaii, World Scientific, 1997.

[8] T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

[9] K. Murakami and T. Takagi, “Gene recognition by combination of several gene-finding programs,” *Bioinformatics*, vol. 14, no. 8, pp. 665-675, 1998.

[10] V. Pavlovic, A. Garg, and S. Kasif, “A Bayesian framwork for combining gene predictions,” *Bioinformatics*, vol. 18, no. 1, pp. 19-27, 2002.

[11] J. Peng and B. Bhanu, “Closed-loop object recognition using reinforcement learning,” *IEEE Trans. Pattern Analysis*

and Machine Intelligence, vol. 20, no. 2, pp. 139-154, 1998.

[12] S. Rogic, A. K. Mackworth, and B. F. Ouellette, "Evaluation of gene-finding programs on mammaliam sequences," *Genome Research*, vol. 11, pp. 817-832, 2001

[13] S. Rogic, B. F. Ouellette, and A. K. Mackworth, "Improving gene recognition accuracy by combining predictions from two gene-finding programs," *Bioinformatics*, vol. 18, no. 8, pp. 1034-1045, 2002.

[14] E. Snyder and G. Stormo, "Identification of protein coding regions in genomic DNA," *Journal of Molecular Biology*, vol. 248, pp. 1-18, 1995.

[15] V. V. Solovyev, A. A. Salamov, and C. B. Lawrence, "Predicting external exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames," *Nucleic Acids Research*, vol. 22, pp. 5156-5163, 1994.

[16] E. Uberbacher, Y. Zu, and R. Mural, "Discovering and understanding genes in human DNA sequence using GRAIL," *Methods Enzymolecular*, vol. 266, pp. 259-281, 1996.

[17] Y. Xu and E. C. Uberbacher, "Reference-based gene model prediction on DNA contigs", *Journal of Computational Biology*, vol. 4, pp. 325-338, 1997.

[18] P. Y. Yin, "Maximum entropy-based optimal threshold selection using deterministic reinforcement learning with controlled randomization," *Signal Processing*, vol. 82, pp. 993-1006, 2002.

[19] M. Zhang, "Identification of protein coding regions in the human genome based on quadratic discriminant analysis," *Proc. National Academic Science USA*, vol. 94, pp. 565-568, 1997.

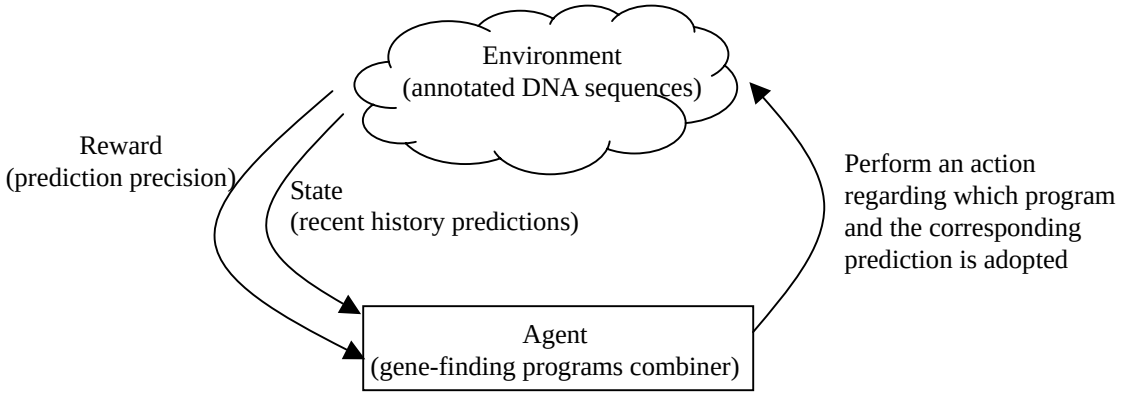

Figure 1 The RLM for combining predictions of gene-finding programs.

Table 1 The average performances obtained using various methods.

	GENSCAN	GenView	HMMgene	AND	OR	RLM-1	RLM-2
Sn	95.0	75.3	91.4	72.0	97.6	95.3	95.2
Sp	91.0	84.1	93.8	98.3	81.0	91.6	91.7
CC	91.7	76.2	91.3	81.9	86.8	92.2	92.2
AC	91.7	76.3	91.3	82.6	87.1	92.2	92.2
ESn	77.6	27.3	75.8	37.2	63.3	78.0	70.4
ESp	75.8	37.2	80.4	59.3	55.4	75.5	65.7