Vol.2015-MPS-103 No.66
Vol.2015-BIO-42 No.66
2015/6/25

Prediction of gene structures from RNA-seq data using dual decomposition

Tatsumu Inatsuki[1] Kengo Sato[1,a)] Yasubumi Sakakibara[1,b)]

Abstract: Numerous computational algorithms for predicting protein-coding genes from genomic sequences have been developed, and hidden Markov models (HMMs) have frequently been used to model gene structures. For eukaryotes, more complex gene structures such as introns make gene prediction much harder due to isoforms of transcripts by alternative splicing machinery. We develop a novel gene prediction method for eukaryote genomes that extends the traditional HMM-based gene prediction model by incorporating comprehensive evidence of transcripts by using RNA sequencing (RNA-seq) technology. We formulate gene prediction as an integer programming problem, and solve it by the dual decomposition technique. To confirm the utility of the proposed algorithm, computational experiments on benchmark datasets were conducted. The results show that our algorithm efficiently and effectively employs RNA-seq data in gene structure prediction.

1. Introduction

Numerous computational algorithms for predicting protein-coding genes from genomic sequences have been developed, and hidden Markov models (HMMs) have frequently been used to model gene structures. Protein-coding genes can successfully be predicted for prokaryotes since no intronic sequences are included. In contrast, for eukaryotes, more complex gene structures (such as introns) make gene prediction much harder due to isoforms of transcripts by alternative splicing machinery. Because of this, adequate accuracy of gene prediction has not yet been realized for higher organisms, such as human, that have complex gene structures.

In this study, we develop a novel gene prediction method for eukaryote genomes. The method extends the traditional HMM-based gene prediction model by incorporating comprehensive evidence of transcripts by RNA-seq technology. We formulate gene prediction as an integer programming problem whose objective function is the sum of the HMM-based score for gene structures and the number of RNA-seq reads that support the gene structure. The algorithm calculates an optimal gene structure that maximizes the objective function subject to several constraints that should be satisfied by the predicted gene structure and the observed RNA-seq reads. In contrast to traditional HMM-based gene prediction algorithms, dynamic programming techniques cannot be applied to this optimization problem because of the additional constraints imposed for RNA-seq reads. We therefore use dual decomposition, which iterates over the following steps. (1) The constraints on RNA-seq reads are relaxed by Lagrangian relaxation. As a result, the original optimization problem is decomposed into two independent sub-problems: HMM-based gene structure prediction and supported read-maximization. These sub-problems can efficiently be solved by the Viterbi algorithm and coefficient comparison. (2) The consistency of RNA-seq constraints is maintained by imposing score penalties on inconsistent predictions. To confirm the proposed algorithm, we conducted computational experiments on benchmark datasets. The results show that our algorithm with RNA-seq data predicts gene structures significantly more accurately than do other methods.

[1] Department of Biosciences and Informatics, Keio University, 3–14–1 Hiyoshi, Kohoku-ku, Yokohama 223–8522, Japan
[a)] satoken@bio.keio.ac.jp
[b)] yasu@bio.keio.ac.jp

2. Methods

2.1 Preliminaries

As several existing works do, we construct a gene model as an HMM in which emission symbols correspond to nucleotides and hidden states correspond to internal gene structures such as exons and introns. We employ m kinds of gene structures, denoted $G = \{g_1, g_2, \ldots, g_m\}$. Let G_{exon} and G_{intron} be the subsets of G that contain gene structures corresponding to exons and introns, respectively.

Let Σ be the set of four DNA bases (adenine (A), cytosine (C), guanine (G), and thymine (T)) and let Σ^* denote the set of all finite DNA sequences, which consist of bases in Σ. Given a DNA sequence $x = x_1, \ldots, x_n \in \Sigma^*$ consisting of n bases, let $\mathcal{Y}(x)$ be the space of all possible gene structures of x. An element $y \in \mathcal{Y}(x)$ is represented as an $n \times m$ binary-valued matrix, where $y_{il} = 1$ indicates that the base x_i is assigned to the gene structure $g_l \in G$ as shown in Fig. 1. We define the problem of gene structure prediction as follows: given a DNA sequence x, predict a gene structure $y \in \mathcal{Y}(x)$.

2.2 Scoring model

A scoring function f is a function that assigns real-valued

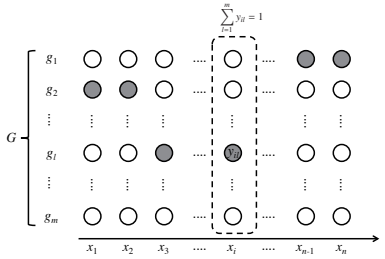

Fig. 1 The binary valued matrix $y \in \mathcal{Y}(x)$ that represents the assignments of gene structures for each nucleotide. Each gray circle represents $y_{il} = 1$, that is, x_i is assigned g_l.

scores to pairs consisting of a DNA sequence x and a gene structure $y \in \mathcal{Y}(x)$. Our aim is to find a gene structure $y \in \mathcal{Y}(x)$ that maximizes the scoring function $f(x, y)$ for a given DNA sequence x. The scoring function f consists of two parts: a gene-model-based scoring function f_{gene} and an evidence-based scoring function f_{evidence}.

The gene-model-based scoring function f_{gene} is defined as

$$f_{\text{gene}}(x, y) = \sum_{i=1}^{n} \sum_{l=1}^{m} \mu_{il} y_{il} + \sum_{i=2}^{n} \sum_{l=1}^{m} \sum_{k=1}^{m} \nu_{lk} y_{i-1,l} y_{ik}, \qquad (1)$$

where μ_{il} is a parameter of the preference of the gene structure g_l at x_i, and ν_{lk} is a parameter of the preference of transition of the gene structures from g_l to g_k. Since each base is assigned to exactly one gene structure as shown in Fig. 1, the following constraint must be satisfied:

$$\sum_{l=1}^{m} y_{il} = 1 \qquad (\text{for } 1 \le \forall i \le n). \qquad (2)$$

To calculate μ_{il}, we employ three types of 5-mers for local features around x_i: those centered at x_i (i.e., $x_{i-2} \dots x_{i+2}$), those centered at x_{i-2} (i.e., $x_{i-4} \dots x_i$), and those centered at x_{i-4} (i.e., $x_{i-6} \dots x_{i-2}$). We define the corresponding parameters of the local features on a gene structure g_l as $\mu^{(0)}_{x_{i-2} \dots x_{i+2}, l}$, $\mu^{(-2)}_{x_{i-4} \dots x_i, l}$, and $\mu^{(-4)}_{x_{i-6} \dots x_{i-2}, l}$, respectively, and use these to calculate μ_{il} as follows:

$$\mu_{il} = \mu^{(0)}_{x_{i-2} \dots x_{i+2}, l} + \mu^{(-2)}_{x_{i-4} \dots x_i, l} + \mu^{(-4)}_{x_{i-6} \dots x_{i-2}, l}. \qquad (3)$$

If transcriptional evidence such as RNA-seq reads is available, we can use them to improve the accuracy of predicting gene structures. To that end, we define an evidence-based scoring function f_{evidence} that evaluates the degree of fitness of a gene structure $y \in \mathcal{Y}(x)$ against observed transcripts. We consider two types of gene structure evidence from observed transcripts. The first type is split positions of mapped transcripts, as obtained by a spliced aligner such as TopHat [10] or STAR [5], which is represented by an $n \times n$ triangular matrix $S = (S_{ij})_{i<j}$, where $S_{ij} = 1$ if at least one transcript mapped onto x is split at x_i and x_j, and $S_{ij} = 0$ otherwise. The second type of evidence is mapped bases of transcripts, represented by an n-dimensional vector $R = (R_i)$, where $R_i = 1$ if at least one transcript mapped onto x covers x_i, and $R_i = 0$ otherwise. We define f_{evidence} as the weighted sum of the number of evidence items that support a gene structure $y \in \mathcal{Y}(x)$:

$$f_{\text{evidence}}(x, y, s, r) = \alpha \sum_{1 \le i < j \le n} \sigma_{ij} s_{ij} + \beta \sum_{1 \le i \le n} \rho_i r_i, \qquad (4)$$

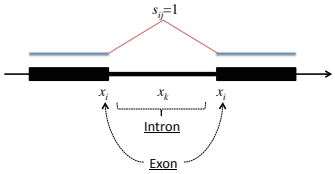

Fig. 2 The constraints (6)-(8) mean that if S_{ij} supports the gene structure y, i.e. $s_{ij} = 1$, then x_k for $i < k < j$ must be assigned to one of the intron states and both ends x_i and x_j must be assigned to one of the exon states.

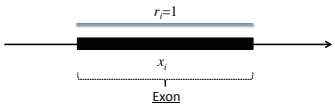

Fig. 3 The constraint (10) means that if R_i supports the gene structure y, i.e. $r_i = 1$, then x_i must be assigned to one of the exon states.

where s_{ij} is a binary-valued variable that indicates whether the split transcripts at x_i and x_j support the gene structure y. Similarly, r_i is a binary-valued variable that indicates whether the transcripts that cover x_i support the gene structure y. Here, $\alpha \ge 0$ and $\beta \ge 0$ mean the contribution ratio of the two types of evidence, and σ_{ij} and ρ_i specify the weight for each evidence. We used $\alpha = 30$, $\beta = 0.1$, $\sigma_{ij} = 1.0$, and $\rho_i = 1.0$ for all i, j in our experiments.

To make the variables s and r consistent with the gene structure y, the following constraints must be satisfied:

$$s_{ij} \le S_{ij} \qquad (\text{for } 1 \le \forall i < \forall j \le n) \qquad (5)$$

$$s_{ij} \le \sum_{l:\, g_l \in G_{\text{intron}}} y_{kl} \qquad (\text{for } 1 \le \forall i < \forall k < \forall j \le n) \qquad (6)$$

$$s_{ij} \le \sum_{l:\, g_l \in G_{\text{exon}}} y_{il} \qquad (\text{for } 1 \le \forall i < \forall j \le n) \qquad (7)$$

$$s_{ij} \le \sum_{l:\, g_l \in G_{\text{exon}}} y_{jl} \qquad (\text{for } 1 \le \forall i < \forall j \le n) \qquad (8)$$

$$r_i \le R_i \qquad (\text{for } 1 \le \forall i \le n) \qquad (9)$$

$$r_i \le \sum_{l:\, g_l \in G_{\text{exon}}} y_{il} \qquad (\text{for } 1 \le \forall i \le n) \qquad (10)$$

Here, the constraints (5) and (9) indicate that supporting transcripts can be enabled only for observed transcripts. In other words, not all the observed transcripts have to support the gene structure. The constraints (6)–(8) represent a condition of supporting transcripts where one of the intron states must be assigned into spliced regions from the $(i + 1)$th base to the $(j - 1)$th base, and that one of the exon states must be assigned to both ends, the ith and jth bases (Fig. 2). The constraint (10) assigns one of the exon states to the base covered by the supporting transcripts (Fig. 3).

2.3 Dual decomposition

Our aim is to find a gene structure $y \in \mathcal{Y}(x)$ that maximizes $f(x, y, s, r) = f_{\text{gene}}(x, y) + f_{\text{evidence}}(x, y, s, r)$ under the constraints (2) and (5)–(10). This is an integer programming problem for which no efficient algorithm is yet known. If we drop the constraints (6)–(8) and (10), maximization of f_{gene} and f_{evidence} can be solved separately and efficiently. This means that the constraints (6)–(8) and (10) make the problem of gene structure pre-

diction with transcriptional evidence extremely complex. To circumvent this difficulty, we deal with these constraints using Lagrangian relaxation [11], which is an efficient technique to solve large and complex problems as appeared in the field of bioinformatics [1], [2], [12], [13]. First, we define the Lagrangian dual by moving the constraints (6)–(8) and (10) to the objective function $f(x, y, s, r)$, as follows:

$$\begin{aligned} L(\lambda) = \max_{y,s,r} \Bigg\{ & f_{\text{gene}}(x, y) + f_{\text{evidence}}(x, y, s, r) \\ & + \sum_{i<j} \sum_{i<k<j} \lambda^{(1)}_{ijk} \left(\sum_{l:\, g_l \in G_{\text{intron}}} y_{kl} - s_{ij} \right) \\ & + \sum_{i<j} \lambda^{(2)}_{ij} \left(\sum_{l:\, g_l \in G_{\text{exon}}} y_{il} - s_{ij} \right) \\ & + \sum_{i<j} \lambda^{(3)}_{ij} \left(\sum_{l:\, g_l \in G_{\text{exon}}} y_{jl} - s_{ij} \right) \\ & + \sum_{i} \lambda^{(4)}_{i} \left(\sum_{l:\, g_l \in G_{\text{exon}}} y_{il} - r_i \right) \Bigg\}. \end{aligned} \tag{11}$$

Here, $\lambda^{(1)}_{ijk} \geq 0$ $(1 \leq i < k < j \leq n)$, $\lambda^{(2)}_{ij} \geq 0$ $(1 \leq i < j \leq n)$, $\lambda^{(3)}_{ij} \geq 0$ $(1 \leq i < j \leq n)$, and $\lambda^{(4)}_{i} \geq 0$ $(1 \leq i \leq n)$ are Lagrangian multipliers. We can then rewrite Eq. (11) as:

$$\begin{aligned} L(\lambda) = \max_{y} & \left\{ \sum_{i=1}^{n} \sum_{l=1}^{m} \mu'_{il} y_{il} + \sum_{i=2}^{n} \sum_{l=1}^{m} \sum_{k=1}^{m} \nu_{lk} y_{i-1l} y_{ik} \right\} \\ & + \max_{s} \sum_{1 \leq i < j \leq n} \sigma'_{ij} s_{ij} + \max_{r} \sum_{1 \leq i \leq n} \rho'_i r_i \end{aligned} \tag{12}$$

where

$$\mu'_{il} = \begin{cases} \mu_{il} + \sum_{j'<i<k'} \lambda^{(1)}_{j'ki'} & (\text{for } g_l \in G_{\text{intron}}) \\ \mu_{il} + \sum_{j'>i} \lambda^{(2)}_{ij'} + \sum_{j'<i} \lambda^{(3)}_{j'i} + \lambda^{(4)}_{i} & (\text{for } g_l \in G_{\text{exon}}) \\ \mu_{il} & (\text{otherwise}) \end{cases} \tag{13}$$

$$\sigma'_{ij} = \alpha \sigma_{ij} - \sum_{i<k<j} \lambda^{(1)}_{ijk} - \lambda^{(2)}_{ij} - \lambda^{(3)}_{ij} \tag{14}$$

$$\rho'_i = \beta \rho_i - \lambda^{(4)}_{i}. \tag{15}$$

This means that we can calculate each term of Eq. (12) independently and efficiently. The first term of Eq. (12) can be efficiently computed using dynamic programming techniques like the Viterbi algorithm for HMM using the following recursion:

$$V(i, l) = \begin{cases} \mu'_{il} & (\text{for } i = 1) \\ \mu'_{il} + \max_{1 \leq k \leq m} [V(i-1, k) + \nu_{kl}] & (\text{for } i > 1) \end{cases} \tag{16}$$

We can obtain the first term of Eq. (12) as $\max_{1 \leq k \leq m} V(n, k)$, and the optimal gene structure $\hat{y}$ can be recovered by traceback from the nth base. The second and last terms of Eq. (12) can be computed simply by finding positive coefficients for s_{ij} and r_i under the constraints (5) and (9).

Since the dual objective function $L(\lambda)$ gives an upper bound of the primal objective function $f(x, y, s, r)$, we aim to minimize Eq. (12) with respect to the multipliers to obtain a better upper bound. The Lagrangian function $L(\lambda)$ is convex, but not differentiable. Thus, to minimize the dual objective function (12), we can apply subgradient optimization in which the Lagrangian multipliers $\lambda^{(1)}$, $\lambda^{(2)}$, $\lambda^{(3)}$ and $\lambda^{(4)}$ are iteratively updated using their subgradients, $\sum_{l:\, g_l \in G_{\text{intron}}} y_{kl} - s_{ij}$, $\sum_{l:\, g_l \in G_{\text{exon}}} y_{il} - s_{ij}$, $\sum_{l:\, g_l \in G_{\text{exon}}} y_{jl} - s_{ij}$, and $\sum_{l:\, g_l \in G_{\text{exon}}} y_{il} - r_i$, respectively. As a result, we can obtain an algorithm similar to the gradient descent shown in Fig. 4, where $\eta_t > 0$ is a step size for each update. It is known that if $\lim_{t \to \infty} \eta_t = 0$ and $\sum_{t=1}^{\infty} \eta_t = \infty$, then the Lagrangian dual $L(\lambda)$ always converges to the optimal value. The update is iterated until a solution is found or the number of iterations reaches a sufficiently large predefined maximum number of iterations T (here, we use $T = 100$).

1: Set $\lambda^{(1)}_{ijk} = 0$, $\lambda^{(2)}_{ij} = 0$, $\lambda^{(3)}_{ij} = 0$ and $\lambda^{(4)}_{i} = 0$.
2: **for** $t = 1$ to T **do**
3: $\quad \hat{y} \leftarrow \arg\max_{y \in \mathcal{Y}(x)} \sum_{i=1}^{n} \sum_{l=1}^{m} \mu'_{il} y_{il} + \sum_{i=2}^{n} \sum_{l=1}^{m} \sum_{k=1}^{m} \nu_{lk} y_{i-1l} y_{ik}$
4: $\quad \hat{s} \leftarrow \arg\max_{s} \sum_{1 \leq i < j \leq n} \sigma'_{ij} s_{ij}$
5: $\quad \hat{r} \leftarrow \arg\max_{r} \sum_{1 \leq i \leq n} \rho'_i r_i$
6: $\quad$ **if** $\hat{y}, \hat{s}, \hat{r}$ satisfy the constraints (6)-(8) and (10) **then**
7: $\quad\quad$ **return** $\hat{y}, \hat{s}, \hat{r}$
8: $\quad$ **end if**
9: $\quad \lambda^{(1)}_{ijk} \leftarrow \lambda^{(1)}_{ij} - \eta_t \left(\sum_{l:\, g_l \in G_{\text{intron}}} y_{kl} - s_{ij} \right)$
10: $\quad \lambda^{(2)}_{ij} \leftarrow \lambda^{(2)}_{ij} - \eta_t \left(\sum_{l:\, g_l \in G_{\text{exon}}} y_{il} - s_{ij} \right)$
11: $\quad \lambda^{(3)}_{ij} \leftarrow \lambda^{(3)}_{ij} - \eta_t \left(\sum_{l:\, g_l \in G_{\text{exon}}} y_{jl} - s_{ij} \right)$
12: $\quad \lambda^{(4)}_{i} \leftarrow \lambda^{(4)}_{i} - \eta_t \left(\sum_{l:\, g_l \in G_{\text{exon}}} y_{il} - r_i \right)$
13: **end for**
14: **return** $\hat{y}, \hat{s}, \hat{r}$

Fig. 4 The algorithm for predicting RNA structural alignments using dual decomposition. T is the maximum number of iterations, set at 100.

The Lagrangian multipliers can be regarded as penalty scores against inconsistency between the gene-model-based scores and the evidence-based scores for the constraints (6)–(8) and (10).

2.4 Learning algorithm

To optimize the feature parameters μ and ν, we employ a max-margin framework called structured support vector machines [15]. Given a training dataset $\mathcal{D} = \{(x^{(k)}, y^{(k)}\}_{k=1}^{K}$, where $x^{(k)} \in \Sigma^*$ and $y^{(k)} \in \mathcal{Y}(x^{(k)})$ are respectively DNA sequences and their corresponding gene structures, we aim to find μ and ν that minimize the objective function

$$\begin{aligned} \mathcal{L}(\mu, \nu) = \sum_{(x,y) \in \mathcal{D}} \Big(& \max_{\hat{y}, \hat{s}, \hat{r}} [f(x, \hat{y}, \hat{s}, \hat{r}; \mu, \nu) + \Delta(y, \hat{y})] \\ & - \max_{s,r} f(x, y, s, r; \mu, \nu) + C(\|\mu\|_1 + \|\nu\|_1) \Big), \end{aligned} \tag{17}$$

where $\|.\|_1$ is the ℓ_1 norm and C is a weight for the ℓ_1 regularization term to avoid over-fitting to the training data. Here, $\Delta(y, \hat{y})$ is a loss function of $\hat{y}$ for y defined as

$$\begin{aligned} \Delta(y, \hat{y}) = & \delta^{\text{FN}} \sum_{i=1}^{n} \sum_{l=1}^{m} I(y_{il} = 1) I(\hat{y}_{il} = 0) \\ & + \delta^{\text{FP}} \sum_{i=1}^{n} \sum_{l=1}^{m} I(y_{il} = 0) I(\hat{y}_{il} = 1) \\ & + \delta^{\text{FN}} \sum_{i=2}^{n} \sum_{l=1}^{m} \sum_{k=1}^{m} I(y_{i-1l} y_{ik} = 1) I(\hat{y}_{i-1l} \hat{y}_{ik} = 0) \\ & + \delta^{\text{FP}} \sum_{i=2}^{n} \sum_{l=1}^{m} \sum_{k=1}^{m} I(y_{i-1l} y_{ik} = 0) I(\hat{y}_{i-1l} \hat{y}_{ik} = 1), \end{aligned} \tag{18}$$

Table 1 Summary of datasets.

	# regions	# of nt	# genes	# mapped reads
Training	2	3038447	511	561676
Test	31	21459613	1111	3144765

Table 2 The accuracy of our model compared with AUGUSTUS.

	PPV	SEN	F
Our model ($f_{\text{gene}} + f_{\text{evidence}}$)	15.7	31.1	20.9
Baseline (f_{gene})	14.1	7.1	9.4
AUGUSTUS	55.9	17.8	27.0

where $I(condition)$ is an indicator function that takes a value of 1 or 0 depending on whether *condition* is true or false. Here, δ^{FN} and δ^{FP} are hyperparameters to control the trade-off between sensitivity and specificity for learning the parameters. In this case, we can calculate the first term of Eq. (17) by replacing scores μ_{ij} and ν_{kl} in Eq. (12) as

$$\bar{\mu}_{ij} = \begin{cases} \mu_{ij} - \delta^{\text{FN}} & (\text{if } y_{ij} = 1) \\ \mu_{ij} + \delta^{\text{FP}} & (\text{if } y_{ij} = 0) \end{cases}$$

$$\bar{\nu}_{kl} = \begin{cases} \nu_{kl} - \delta^{\text{FN}} & (\text{if } y_{i-1l} y_{ik} = 1) \\ \nu_{kl} + \delta^{\text{FP}} & (\text{if } y_{i-1l} y_{ik} = 0) \end{cases}$$

We used $C = 1.0$, $\delta^{\text{FN}} = 1.0$, and $\delta^{\text{FP}} = 1.0$ in our experiments.

To minimize the objective function (17), we can apply stochastic subgradient descent or forward–backward splitting [7].

3. Results and discussion

To verify the model described in Sec. 2, we developed a preliminary implementation of our model. We conducted computational experiments using EGASP [8] as training and test datasets. Our model was trained using two regions from the EGASP training dataset, ENm004 and ENm006. The model was then evaluated on 31 regions from the EGASP test dataset. Gene annotations were obtained from GENECODE [9]. We used two replicates of RNA-seq data from liver hepatocellular carcinoma cell line HepG2 [4] as transcriptional evidence. We mapped RNA-seq reads into training and test regions using STAR [5]. Table 1 shows a summary of the datasets.

We evaluated the accuracy of predicting gene structures through nucleotide-level measurements as defined by [3]. The nucleotide-level accuracy is assessed by positive predictive value ($PPV = \frac{TP}{TP+FP}$) and sensitivity ($SEN = \frac{TP}{TP+FN}$), where TP is the number of nucleotides at which exon states are correctly predicted (true positives), FP is the number of nucleotides at which exon states are incorrectly predicted (false positives), and FN is the number of nucleotides in the true exon states that were not predicted (false negatives). We also used the F-value as a balanced measure between PPV and SEN, which is defined as their harmonic mean $F = \frac{2 \times PPV \times SEN}{PPV + SEN}$.

Table 2 shows the accuracy of our model (i.e. $f_{\text{gene}} + f_{\text{evidence}}$) in comparison to our model without the evidence-based scores (i.e. only f_{gene}) as a baseline method, and suggests that significant improvement, especially in sensitivity, can be observed by using the evidence-based scores. We also compared our method with AUGUSTUS [14] with default options, which was trained using the same regions as in our setting.

Although our model is improved by the use of evidence-based scoring, it is not yet sufficiently accurate compared with existing methods. This is because the preliminary implementation of our model cannot perform the learning algorithm using a large dataset, despite using dual decomposition for efficiency. To realize large-scale training, we need to implement an improved stochastic subgradient descent algorithm such as AdaGrad [6]. We also need to further optimize hyperparameters including α and β for the weights of transcriptional evidences, C for the ℓ_1 regularization, and δ^{FN} and δ^{FP} to control the trade-off between sensitivity and specificity.

4. Concluding Remarks

We develop a novel gene prediction method for eukaryote genomes that extends the traditional HMM-based gene prediction model by incorporating comprehensive evidence of transcripts using RNA sequencing (RNA-seq) technology. We formulated gene prediction as an integer programming problem, and solved it using the dual decomposition technique. To confirm the proposed algorithm, we conducted computational experiments on benchmark datasets. The results showed that our algorithm with RNA-seq data works efficiently and effectively.

References

[1] Andreotti, S., Klau, G. W. and Reinert, K.: Antilope–a Lagrangian relaxation approach to the de novo peptide sequencing problem, *IEEE/ACM Trans Comput Biol Bioinform*, Vol. 9, No. 2, pp. 385–394 (2011).

[2] Bauer, M., Klau, G. W. and Reinert, K.: Accurate multiple sequence-structure alignment of RNA sequences using combinatorial optimization, *BMC Bioinformatics*, Vol. 8, p. 271 (2007).

[3] Burset, M. and Guigo, R.: Evaluation of gene structure prediction programs, *Genomics*, Vol. 34, No. 3, pp. 353–367 (1996).

[4] Djebali, S. et al.: Landscape of transcription in human cells, *Nature*, Vol. 489, No. 7414, pp. 101–108 (2012).

[5] Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M. and Gingeras, T. R.: STAR: ultrafast universal RNA-seq aligner, *Bioinformatics*, Vol. 29, No. 1, pp. 15–21 (2013).

[6] Duchi, J., Hazan, E. and Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization, *The Journal of Machine Learning Research*, Vol. 12, pp. 2121–2159 (2011).

[7] Duchi, J. and Singer, Y.: Efficient online and batch learning using forward backward splitting, *Journal of Machine Learning Research*, Vol. 10, pp. 2899–2934 (2009).

[8] Guigo, R. et al.: EGASP: the human ENCODE Genome Annotation Assessment Project, *Genome Biol.*, Vol. 7 Suppl 1, pp. 1–31 (2006).

[9] Harrow, J. et al.: GENCODE: the reference human genome annotation for The ENCODE Project, *Genome Res.*, Vol. 22, No. 9, pp. 1760–1774 (2012).

[10] Kim, D., Pertea, G., Trapnell, C., Pimentel, H., Kelley, R. and Salzberg, S. L.: TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions, *Genome Biol.*, Vol. 14, No. 4, p. R36 (2013).

[11] Korte, B. and Vygen, J.: *Combinatorial Optimization: Theory and Algorithms*, Springer Verlag, Berlin, Germany (2008).

[12] Kumozaki, S., Sato, K. and Sakakibara, Y.: A machine learning based approach to de novo sequencing of glycans from tandem mass spectrometry spectrum, *IEEE/ACM Trans Comput Biol Bioinform* (in press).

[13] Sato, K., Kato, Y., Akutsu, T., Asai, K. and Sakakibara, Y.: DAFS: simultaneous aligning and folding of RNA sequences via dual decomposition, *Bioinformatics*, Vol. 28, No. 24, pp. 3218–3224 (2012).

[14] Stanke, M., Schoffmann, O., Morgenstern, B. and Waack, S.: Gene prediction in eukaryotes with a generalized hidden Markov model that uses hints from external sources, *BMC Bioinformatics*, Vol. 7, p. 62 (2006).

[15] Tsochantaridis, I., Joachims, T., Hofmann, T. and Altun, Y.: Large Margin Methods for Structured and Interdependent Output Variables, *J. Mach. Learn. Res.*, Vol. 6, pp. 1453–1484 (2005).