The application of alternative splicing graphs in quantitative analysis of alternative splicing form from EST database

Hsun-Chang Chang, Po-Shun Yu and Tze-Wei Huang
Department of Computer Science and Information Management,
Providence University, Shalu,
Taichung County, Taiwan 443, ROC
E-mail: hcchang@cs.pu.edu.tw E-mail: peteryu@cs.pu.edu.tw
E-mail: g9134012@cs.pu.edu.tw

Fang-Rong Hsu
Department of Information Engineerging and Computer Science,
Feng Chia University,
Taichung, Taiwan 40724, ROC
E-mail: frhsu@fcu.edu.tw

Yaw-Ling Lin*
Department of Computer Science and Information Management,
Providence University, Shalu,
Taichung County, Taiwan 443, ROC
E-mail: yllin@pu.edu.tw
*Corresponding author

Abstract: Alternative splicing is an important mechanism for generating protein diversity from a single gene. However, standard molecular biology techniques have only identified several hundred alternative splicing variants and created a bottleneck in terms of experimental validation. In this paper, we propose models of weighted alternative splicing graphs and ways of generating all alternative splicing forms from an alternative splicing graph. By formulating linear programming models, we can deduce the quantitative distributions of various alternative splicing forms. Linear time algorithms that produce all alternative splicing variants with their corresponding probabilities are proposed. Finally, by aligning sequences of EST databases to the genomic data, locations of exons as well as the alternative splicing sites can be identified, and several sets of putative alternative splicing forms is inferred using these methods.

Keywords: splicing graph; alternative splicing; expressed sequence tag (EST); weighted alternative splicing graph; algorithm; linear programming; alternative splicing form.

Reference to this paper should be made as follows: Chang, H-C., Yu, P-S., Huang, T-W., Hsu, F-R. and Lin, Y-L. (2005) 'The application of alternative splicing graphs in quantitative analysis of alternative splicing form from EST database', *Int. J. Computer Applications in Technology*, Vol. 22, No. 1, pp.14–22.

Biographical notes: Hsun-Chang Chang is a graduate student of Department of Computer Science and Information Management at Providence University. His research areas include bioinformatics, graph algorithm, parallel and distributed computing. He is interested in applying algorithm and parallel computing environment to solve bioinformatics problems.

Po-Shun Yu is a graduate student of Providence University in Taichung, Taiwan. He has studied Computer Science when he was a college student. He was a member of PU's Algorithm Lab from its inception until 2002. At present he is an engineer with a high tech company.

Tze-Wei Huang was a graduate student of Department of Computer Science and Information Management at Providence University. At present, he is a PhD student of Department of Computer Information Science at National Chiao Tung University and doing genetic network studies.

Fang Rong Hsu received a PhD in Computer Science from the National Chiao-Tung University, Hsinchu, Taiwan, 1992. He is a Professor in the Department of Information Engineering and Computer Science at Feng-Chia University, Taiwan, where he also directs the Bioinformatics Research Center. He was the Chairperson and a Professor of the Department of Bioinformatics, Taichung Healthcare and Management University from 2003 to 2004. He was the Chairperson and a Professor of the Department of Information Technology, Taichung Healthcare and Management University from 2002 to 2003. He was an Associate Professor and a Professor of Providence University from 1994 to 2002. His current research interests include bioinformatics, grid computing, data mining and graph algorithm.

Dr. Lin obtained his PhD degree from the Department of Computer Science, State University of New York at Stony Brook in 1993. He is currently a full Professor of the Department of Computer Science and Information Engineering, Providence University, Taichung, Taiwan. He was the chairman of the department from 1994 to 1997, and he is currently the Dean of the College of Computing and Informatics, Providence University. His current research interests include bioinformatics, computational geometry, graph algorithms, data compression, and algorithm design and analysis in general.

1 INTRODUCTION

RNA splicing is an essential and precisely regulated post-transcriptional process that occurs prior to mRNA translation. A gene is first transcribed into a pre-messenger RNA (pre-mRNA), which is a copy of the genomic DNA containing intronic regions destined to be removed during pre-mRNA processing (RNA splicing), as well as exonic sequences that are retained within the mature mRNA. Alternative splicing of eukaryotic pre-mRNAs is a mechanism for generating potentially many transcript isoforms from a single gene. It serves versatile regulatory functions in controlling major developmental decisions and fine-tuning of gene function (Lopez, 1998). The majority of alternative splicing events give rise to different protein products. At least 35% of human genes undergo alternative splicing (Kan et al., 2001, 2002). The physiological functions of these splice variants may be similar, opposite, or unrelated. In addition, they may differ in other properties, such as stability, tissue and cellular localisation, temporal expression pattern, and responses to agonists or antagonists. The presence or level of specific splice variants may cause and/or indicate pathological or normal conditions. A classic example is the Prostate Specific Anitgene, the most important marker available today for diagnosing and monitoring patients with prostate cancer (Heuze et al., 1999).

However, alternative splicing on biological functions is hard to distinguish and identify. The abundant prevalence of *expressed sequence tags* (ESTs) (Paquola et al., 2003) serves as a good target in helping the researchers understanding and exploring gene expression information within tissues. dbEST is a division of GenBank that contains sequence data and other information on 'single-pass' cDNA sequences, or Expressed Sequence Tags, from a number of organisms. The abundance of the expressed sequence tag collection in dbEST certainly provides an extensive resource for detecting alternative splicing form on a genomic scale.

Expressed sequence tags (ESTs) are single sequencing reads from cDNA clones. Even though ESTs resources are plagued by problems such as poor sequence quality and intronic contamination, they are still important tools for exon finding (Burke et al., 1998) and detection of alternative splicing (Kan et al., 2001) at the present time. Biologists assemble them into consensus sequence in order to form EST contigs and analyse alternative splicing variants (Burke et al., 1998; Coward et al., 2002; Modrek and Lee, 2001). Expressed sequence tags (ESTs) from normal and tumour tissues have been deposited in public databases (Xie et al., 2002). With the availability of a nearly complete sequence of the human genome, aligning expressed sequence tags (EST) to the genomic sequence has become a practical and powerful strategy for gene prediction. Elucidating gene structure is a complex problem requiring the identification of splice junctions, gene boundaries, and alternative splicing variants (Kan et al., 2001). Recent EST based studies have revealed that alternate transcription, including alternative splicing, polyadenylation and transcription start sites, occurs within most parts of human genes. Transcripts form surveys have yet to integrate the genomic context, expression, frequency, and contribution to protein diversity of isoform variation (Hide et al., 2001). The problem of using expressed sequence tags (ESTs) for genomic DNA annotation and prediction of exon-intron structure is not trivial, even when all splicing sites are correctly defined. One of the main difficulties is that a considerable number of ESTs map to intronic regions, or could be products of aberrant or incomplete splicing (Mironov and Gelfand, 1999; Modrek and Lee, 2001). Moreover, the computational challenges of finding all alternative splicing variants can be understood if one considers a gene with 20 exons with one alternative splicing site per exon. In this case, the number of potential splicing variants are at least 2^{20}.

We can take advantage of the notion of the *splicing graph* (Heber et al., 2002) built from available EST and cDNA data. The graph conveniently encodes all potential splicing variants and shows the relationships between different transcripts implying the over-all structure of transcripts. Here the idea is to abandon the linear sequence representation of each transcript and replace it with a directed graph called *splicing graph* representation where each transcript corresponds to a path in the graph. Splicing graphs may be rather complicated. As an example, the gene model of the *Drosophila Dscam* gene implies roughly 38,000 potential transcripts (Graveley, 2001). The benefit of the *splicing graph* approach is that it takes into account all ESTs derived from different transcripts, which cover a given position (vertex) rather than only ESTs derived from a single transcript as in the conventional approach.

Linear and integer programming have proved valuable for modelling many and diverse types of problems in planning, routing, scheduling, assignment, and design. For example, linear programming techniques have been used in some bioinformatics area for analysing and optimising children's diets, in neural networks for radiotherapy treatment planning, in the assessment of the profit from strategies to reduce the prevalence of Staphylococcus aureus mastitis, etc.

Human and mouse genomes share similar long-range sequence organisation, while most of their genes are homologous. Since alternative splicing is a frequent and important aspect of gene regulation, it is of interest to assess the level of conservation of alternative splicings (Thanaraj et al., 2003). Past researches show that analysing the EST data is most helpful not only in gene expressions but also in the qualitative and quantitative analysis of alternative splicings.

In this paper, we use the idea of splicing graphs and present methods of performing quantitative analysis for all possible alternative splicing forms. Furthermore, we utilise the method of parallelising these approaches by sharing the works to the available multi-processors. Our graphtheoretical approach basically exploits all possible directed paths starting from the source of the (probabilistic weighted) splicing graph, and correctly reports all variants of splicing forms as well as their corresponding probabilities. The method we proposed is also closely related to the notation of Eulerian superpaths problem (Mulyukov and Pevzner, 2002; Pevzner et al., 2001a, 2001b; Pevzner and Tang, 2001).

2 METHOD

At first, the frequency count of each directed edge (E_i, … E_j) is calculated from the EST database. Then, a number of alternative splicing forms (x_1, ..., x_A) is derived by using the procedure ASFFIND, illustrated in Figure 1, in our system. Furthermore, each alternative splicing form, E_i, is treated as an input of the LP solving system to finally obtain its corresponding probability. As an example, for a given alternative splicing graph with 13 nodes (exons) and 18 edges (alternative splicing sites), as shown in Figure 2, the corresponding 18 possible alternative splicing forms are listed as following:

1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 13
1 – 3 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 13
1 – 2 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 13
1 – 2 – 5 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 13
1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 13
1 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 13
1 – 2 – 3 – 5 – 12 – 13
1 – 3 – 12 – 13
1 – 2 – 4 – 5 – 12 – 13
1 – 2 – 5 – 12 – 13
1 – 2 – 3 – 4 – 5 – 12 – 13
1 – 3 – 4 – 5 – 12 – 13
1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13
1 – 3 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13
1 – 2 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13
1 – 2 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13
1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13
1 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13

```
DFS(G)
Input: Alternative splicing sites V = ⟨v1, v2, ..., vn⟩.
Output: All putative alternative splicing forms ASFs.
for each vertex u ∈ V[G]
    do color[u] ← WHITE; π[u] ← NIL;
time ← 0;
for each u ∈ V[G]
    do if color[u] = WHITE
        then DFS-VISIT(u)

DFS-VISIT(u)
color[u] ← GRAY;    ▷ While vertex u has just been discovered (u, v)
d[u] ← time ← time + 1;
for each vertex v ∈ Adj[u]    ▷ Explore edge (u, v).
    do if color[v] ← WHITE
        then π[v] ← u
             DFS-VISIT(u)
color[u] ← BLACK;
f[u] ← time ← time + 1;

ASF-FIND(G)
call DFS(G) to compute finishing times f[v] for each vertex v.
as each vertex is finished, insert it onto the front of a linked list.
for each v ∈ V do out-list(v) ← NIL;
return the linked list of vertices.    ▷ Topological sort order (u, v).
out-list(s) ← ⟨s⟩
for each vertex u in the topological sorted ordering do
    if u is sink
        then Output out-list(u)
    else
        for each vertex v ∈ Adj[u] do    ▷ u is predecessor of v.
            APPEND(out-list(u) ∘ v, out-list(v))    ▷ Append u's out-list to v's out-list.
```

Figure 1 *Finding all putative alternative splicing forms*

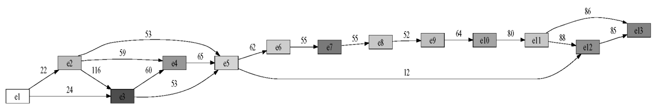

Figure 2 *A splicing graph*

2.1 Finding alternative splicing form by using alternative splicing graphs

Given a directed acyclic graph $G = (V, E)$ with a source vertex $s \in V$, the vertex set V represents the state and E represents the transition probability from one state to another. For a (connected) path of G, starting from s, we use $N(p)$ to denote the set of all immediate vertices following the *last* vertex of the path p; when p is just a single vertex $v \in V$, it follows that $N(v)$ is the neighbouring set of v. Let $p_1, \ldots, p_n$ be the set of all RNA transcripts for a given gene of interest. Each transcript p_i corresponds to a set of genomic positions (also called *exons*) $V_i \subset V$; note that $V_i \neq V_j$ for $i \neq j$. It follows that the set of all transcribed position $V = V_1 \cup V_2 \cup \cdots \cup V_n$ is the union of all sets V_i. The *splicing graph* G is the directed graph on the set of transcribed sites V that contains a directed edge (u, v) if and only if u and v are consecutive positions in one of the transcripts. Every transcript can be viewed as a path in the splicing graph G, and the whole graph G is the union of all paths. Each transcribed site has at least one emission probability from one transcribed site to next.

Here in our model, we assume the transcribing process is a probabilistically independent random process. It follows that all possible path variants and their corresponding probabilities can be deduced from the random model. A higher probability in one of these paths implies a more probable putative alternative splicing forms can be produced by them, resulting in a quantitative analysis of different ASFs. These possible alternative splicing form in following steps. First, a depth first search can be performed on G to obtain the topological sorted ordering of these transcribed sites. Following the topological sorted ordering, we check whether the endpoint of a vector is sink. Here a sink vertex is one without descendants. Associated each vector with an predescendant-list to store its visited predescendants, if the vector is sink, output its corresponding visited list; otherwise, the predecessors of this vector's output-list should be appended to its children's visited list. The detailed description of our algorithm is shown in Figure 1.

Definition 1 (weighted alternative splicing graph)**:** An edge-weighted directed acyclic splicing graph $G = (V, E)$ is a weighted alternative splicing graph if G has a single starting (source) vertex $s \in V$, and each edge e of G is associated with a probability $0 \leq \text{Prob}(e) \leq 1$, such that $\forall u \in V$, we must have $\sum_{v \in N(u.)} \text{Prob}(uv) = 1$.

Definition 2 (weighted alternative splicing forms problem)**:** Given a weighted alternative splicing graph $G = (V, E)$, the alternative splicing forms problem is to find all possible alternative splicing graph forms with their associated probabilities.

Let $\pi = \langle s, v_1, \ldots, v_k \rangle$ be a path of a weighted alternative splicing graph G starting from source s. Denote the probability associated with the path π by $\text{Prob}(\pi) = \text{Prob}(sv_1) \cdot \text{Prob}(v_1v_2) \ldots \text{Prob}(v_{k-1}v_k)$ It follows that

Lemma 1: *Given a weighted alternative splicing graph G with the source s, let $P = \{\pi_1, \pi_2, \ldots, \pi_p\}$ be all distinct paths from the source s of equal length n. It follows that* $\sum_{i=1}^{p} \text{Prob}(\pi_i) = 1$ *for every $n \geq 1$.*

Proof. The lemma can be easily proved by induction on the paths of length k starting from s. The condition obviously holds if $n = 1$ by definition. Assuming the property holds for length k. Now consider the case of all paths of length $k + 1$. Let the set of all paths of length k be $\{a_1, \ldots, a_x\}$, and we have $\text{Prob}(a_1) + \cdots + \text{Prob}(a_x) = 1$ by inductive hypothesis. Note that the set of all paths of length $k + 1$ will be $\{a_1 \circ N(a_1)\} \cup \cdots \cup \{a_k \circ N(a_x)\}$. Let $q(\pi)$ denote the last vertex of a path π. By independent property, we have the sum of probabilities for paths of length $k + 1$ being $\sum_{v \in N(a_1)} \text{Prob}(a_1 \circ v) + \cdots + \sum_{v \in N(a_x)} \text{Prob}(a_x \circ v) = \text{Prob}(a_1) \cdot \sum_{v \in N(a_1)} \text{Prob}(q(a_1)v) + \cdots + \text{Prob}(a_x) \cdot \sum_{v \in N(a_x)} \text{Prob}(q(a_x)v) = \text{Prob}(a_1) + \cdots + \text{Prob}(a_x) = 1$ since $\sum_{v \in N(u)} \text{Prob}(uv) = 1$ for any vertex $u \in V$ by the definition of weighted alternative splicing graphs.

Theorem 1: *Let $U \subset V$, $U = \{u_1, u_2, \ldots, u_m\}$, be the set of all sinks within a weighted alternative splicing graph G. Let $P = \{\pi_1, \pi_2, \ldots, \pi_n\}$ be all distinct paths from the source s to sinks, then* $\sum_{i=1}^{p} \text{Prob}(\pi_i) = 1$.

Proof: Given a weighted alternative splicing graph G, without loss of generality, we can add self-loops to all its sink vertices as shown in Figure 3, and obtain an augmented graph G'. Let l denote the length of the longest path of P. Note that the set of all paths of length l, P', is just the set of paths P such that some shorter paths of P are lengthened by appending repeated sinks. However, by lemma 1, we have $\sum_{\pi \in P'} \text{Prob}(\pi) = 1$, it follows that $\sum_{\pi \in P} \text{Prob}(\pi) = 1$.

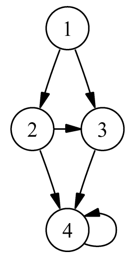

Figure 3 *Adding self-loops to sinks of a DAG*

Theorem 2: *Give an alternative splicing graph. We can correctly compute all possible alternative splicing forms in time linearly propositional to the size of its alternative splicing forms.*

Proof: We prove the correctness of this theorem using the Theorem 1 and Lemma 1, Consider the algorithm shown in Figure 1. The graph $G = (V, E)$ is represented using adjacency lists. The colour of each vertex $u \in V$ is stored in the variable *colour*$[u]$, and the predecessor of u is stored in the variable $\pi[u]$. If u has no predecessor, the $\pi[u] = \text{NIL}$.

Each vertex v has two timestamps, the first timestamp $d[v]$ records when v is first discovered (and greyed), and the second timestamp $f[v]$ records when the search finishes examining v's adjacency list (and blackens v) . During an execution of DFS, the loops on lines 1–2 and lines 5–7 of DFS take time $\Theta(|V|)$, exclusive of the calls to DFS-VISIT(v). The procedure DFS-VISIT(v) is called exactly once for each vertex $v \in V$, since DFS-VISIT(v) is invoked on white vertices and the first thing it does is paint the vertex grey. During an execution of DFS-VISIT(v), the loop on lines 3–6 is executed $\Theta(|E|)$ times.

Finally, the procedure ASF-FIND takes $\Theta(|V| + |E|)$ for the topological sort, and it takes time linearly proportional to the output size of the alternative splicing forms since it can be seen that each element of the output uses constant time in appending the out-lists of predecessor vectors onto the out-lists of its children vector, performed on line 11.

2.2 Parallel alternative splicing finding

As we mentioned before, finding all possible alternative splicing variants is a computational challenge. To overcome this, we can parallelise our algorithm based on Depth-First Search and then distribute the search jobs among various processors. We exploit the method proposed by Rao and Kumar (1987a, 1987b) to achieve the target. The detailed procedure of the modified algorithm is shown in Figure 4. In the parallel implementation, each processor performs DFS on a disjoint part of search space. When a processor finishes searching its part of the search space, it requires an unsearched part from other processors. If the search space is finite and the problems have solutions, then all the processors eventually run out of work, and the algorithm terminates. As each processor searches the tree, unexplored nodes can be stored as a stack. Each processor has its own local stack on which it executes 0. When local stack is empty, it takes some of the unvisited candidate from the other process's stack. From then on, the search space is divided and distributed. We use P_i and *stack*[i] to denote the ith processor and the stack of ith processor. When the search is finished, GETWORK() is called to get more work. If new work is not received, then a TERMINATION-TEST() routine is called to see whether all other processors have finished. Moreover, procedure GETWORK is architecture dependent. It can be developed for shared address space platforms and message passing platforms respectively (Rao and Kumar, 1987b). Two characteristics of parallel DFS are critical to determining its performance. One is the strategy for splitting work at a processor, and the other is the scheme to determine the donor processor when a processor becomes idle. Apart from this, Parallel search algorithms incur overhead from several sources. These include communication overhead, idle time due to load imbalance, and contention for shared data structures. These important parameters of parallel DFS are discussed in this book (Grama et al., 2003). In order to avoid lengthy writing, we will not discuss the detail here.

```
PARA-DFS: For each processor P_i:
while (not terminated) do
    if (stack[i] = empty) then GETWORK();
    while (stack[i] ≠ empty) do
        AFS-FIND (stack[i]);
        GETWORK();
    TERMINATION-TEST();
```

Figure 4 *Basic parallel DFS routine in each of processors*

2.3 Quantity analysis

Our analysis is based on two major types of data: EST database and alternative splicing form. Within our system, we keep a set of EST database that is relevant to our targeted analysed domains of putative alternative splicing forms. While the putative alternative splicing forms is either generated by some systematical methods or proposed by biologists based on their professional judgements. Here we briefly describe the notations used by our proposed model before we get into the details of our methods.

Exons are areas of genomes that can be translated to produce mRNA and finally transcribed to proteins. Different combinations of exons can be spliced together to produce different mRNA isoforms of a gene, encoding structurally and functionally different protein products (Xu et al., 2002). We use e_i to denote a possibly expressed exons in the final alternative spicing forms, and let $\Sigma = \{e_1, e_2, \ldots, e_n\}$ denote the set of all different exons. The locations and the precise segment of exons can be obtained from the EST database. Here we denote an EST by symbol y, conceptually just a substring of an ASF. Note that an alternative splicing form is composed by linking several alternative splicing sites. Thus an alternative splicing site can be found by checking the EST database and aligning the EST sequence to the genome. In particular, an EST containing two consecutive exons define an alternative splicing site on the genome. Finally, an alternative splicing form is just a list of exons, altogether constitute an expressed mRNA. Thus each alternative splicing form (ASF) is denoted by $a = \langle e_1, e_2, \ldots, e_k \rangle$, or in short, $a \in \Sigma^*$.

Our method assumes the availability of two sets of data: a set of biological meaningful alternative splicing forms, consists of all possible different exons compositions for one gene: $A = \{a_1, a_2, \ldots, a_{|A|}\}$; $a_i \in \Sigma^*$, and an EST database, briefly denoted by a set of ESTs, $Y = \{y_1, y_2, \ldots, y_{|Y|}\}$. This information is retrieved from quantitative denoted dbEST. Here, the number of occurrence of an EST y is denoted by $N(y)$. Furthermore, let $p(y \mid a)$ denote the probability of picking the EST y out of one particular alternative splicing form a, and $p(e \mid a)$ denote the probability of picking the exon e out of one particular alternative splicing form a.

The problem of quantifying a given set of alternative splicing forms A is to compute the expressing probability of each single ASF $a \in A$. Since the only quantitative information we have in the sequence database is the quantitative denoted dbEST. We assume that laboratorial process of transcribing an EST sequence from the alternative splicing form is a random process with some associated probability density function. It follows that the different quantitative expressions of ESTs are caused by the combination of different quantitative expressions of original ASFs as well as the random process guarded by the distribution of the probability density function.

Since there is no obvious ways of obtaining the putative probability distribution of the underlying random process, as an initial guess, we use the uniform distribution for the first model. Thus the underlying probability density function has the following form:

$$P(e \mid a) = |e|/|a| \tag{1}$$

By this way, the probability will change along with length of alternative splicing form closely. The longer the alternative splicing form the more probability it has. Although we did not know what the real model of an alternative splicing form is we can imagine that every alternative splicing form will have a particular model to fit itself.

Many possible underlying probability density functions are possible. Other than the linear (uniform) model, on our system, we also use probability density function of different quadratic forms, as well as other mathematically interesting distribution functions. In the future, molecular biologists may have further progress on understanding the underlying quantitative mechanism of extracting ESTs out of an ASF. This information can be easily incorporated into our system to achieve higher order of accuracy.

2.4 Linear programming

Depending upon the biological knowledge concerning the EST expressing probability distributions, our system gathers a list of candidate probability density functions chosen by users. From the chosen density function *f*, and for each exon *e*, we have the following equalities (with a, a_i being an ASF):

$$P(e \mid \mathrm{a}) = \int_{\mathrm{Start}(e)}^{\mathrm{End}(e)} f(x)\mathrm{d}x \tag{2}$$

$$P(a_i)P(e \mid a_i) = P(e, a_i) \tag{3}$$

$$\sum_{a_i \in A(e)} P(a_i)P(e|a_i) = \sum_{a_i \in A(e)} P(e, a_i) = P(e) \tag{4}$$

Here $P(e \mid a)$ denote the probability of an exon e being expressed from a particular alternative splicing form a, and $A(e)$ is the set of alternative splicing forms containing exon e. We can formulate these equalities as a set of linear equations by treating these $P(a_i)$s as variables, and $P(e \mid a_i)$s as constants determined from the given density function *f*. The following linear programming system is obtained:

$$\left\{ \sum_{a_i \in A(e)} P(a)P(e \mid a) = P(e) \,\middle|\, e \in e(Y) \right\} \tag{5}$$

here $e(Y)$ denote the starting exons of the set of ESTs Y. Note that the desired quantitative information $P(a_i)$s constitutes a probability function, and we need to add two explicit conditions:

- $\forall a_i 1 \geq P(a_i) \geq 0$
- $\sum_{a_i \in A} P(a_i) = 1$.

With the needed linear programming system model at hand, all it remains to find all $P(a)$ for each $a \in A$ by any conventional LP solver.

2.5 Preliminary experiments and result

To validate our approach we applied it to get the human adenylosuccinate lyase (ADSL) gene (Jaeken and van den Berghe, 1984; Sebesta et al., 1997; Kmoch et al., 2002). Adenylosuccinate lyase (ADSL) is a bi-functional enzyme acting in de novo purine nucleotide recycling. To date, about 50 patients have been diagnosed worldwide and reports on about half of them have been published. The disease usually appears within the first months of life with neurological involvement. Our input data come from UniGene clusters of ESTs (Boguski and Schuler, 1995; Schuler, 1997; Schuler et al., 1996; Wheeler et al., 2003). It contains 13 exons of overall length about 2 kb. In order to compare the accuracy of our approach with *splicing graphs* approach that were studied by Kmoch et al. (2002). We store our input data into our database and find all possible alternative splicing sites. Two examples follow. Then we compute all possible alternative splicing forms at our bioinformatics workstation. At the present time, our preliminary experiments are based on our sequential algorithm, we will make use of the parallel version to compare the experimented results and the performance in the future.

Currently our system adapt the GNU Octave system as our LP solver. Octave is an open source numerical analysis software package that is very popular among scientists and engineers. It provides a convenient command line interface for solving linear and nonlinear problems numerically, and for performing other numerical experiments using a language that is mostly compatible with Matlab. It may also be used as a batch-oriented language. Octave has extensive tools for solving common numerical linear algebra problems. Octave is not only easily adaptable but also compatible with Matlab mathematical tools. Furthermore, it has parallel version that can run over a cluster of machines or multiple processors using MPI (Snir et al., 1996). Octave is thus chosen as our linear programming system solver.

Moreover, we utilise the open source licensed software called Graphviz to draw the alternative splicing graphs. Graphviz is a set of graph drawing tools for Unix or MS-Windows implemented by AT&T Research. It has five components to make hierarchical layouts of trees, DAGS (directed acyclic graphs), virtual physical (spring model) layouts of undirected graphs successful. More importantly, it is distributed without cost in many common platforms. Graphviz not only create convenient graph drawing systems and web services but also manipulate graph structures and generate graph layouts. Graphviz is thus chosen and installed in our bioinformatic workstations.

In the meantime, we have cooperated with Dr. Fong-Rong Hsu and used their EST database (Hsu et al., 2004) in choosing the suitable alternative splicing sites as well as the putative alternative splicing forms. Making the decision of choosing of a putative alternative splicing site is a difficult task. We align the EST position to the genomic data, and count the number of EST at the same position. If the number of EST greater than five or given by user, we choose it be a segment and connect the overlap one. After making choice of segment, then count EST by each new segment and you will get the new number of EST of each segment. By the uniform distribution, each segment will have it own probability with length.

Finally, this linear programming system is solved by Octave with two added conditions:

- all variants are between zero and one
- add all of variant are among one.

The probability of these alternative splicing forms is then deduced by the LP solver. Furthermore, we will use another model to try to incorporate the verified data given by biologists and publish our system in the coming weeks.

3 DISCUSSION

Bioinformatics has become an essential tool not only for basic research but also for applied research in biotechnology and biomedical sciences. Currently, the field is undergoing an enormous expansion, as witnessed by the dramatic increase in the number of advertisements for bioinformatics personnel during the past year (Rehm, 2001).

We have shown here that our method, quantities analysis of alternative splicing form, is a new approach to detect alternative splicing form. Our results can be useful to biologists in several ways. First, this method can be used not only in analysed alternative splicing but also in Motif. Second, it will enhance alternative splicing database to become high correctly. We introduced a novel statistical approach for estimating the frequency of alternative splicing events based on observed frequencies in ESTs.

In contrast to other traditional methods, our approach does not need large sets of training data to construct species-specific models of genes or assemble ESTs into linear sequences, we take advantage of our algorithm and alternative splicing sites acquiring from UniGene clusters of ESTs to calculate all possible paths and their probabilities.

The increasing prevalence of alternative splicing at higher EST coverage may simply be attributable to an increased chance of detecting low-probability events with continued sampling (Kan et al., 2002). We set the length as the probability of each segment. Finally, we put this data into our linear programming system to calculate the probability of alternative splicing form. Table 1 is our result output from our linear programming system. This experimentation is very unsophisticated at choose exons. Table 1 shows that the higher probability of alternative splicing form is because of the length of segment it has. The equation below is the set of linear programming.

Table 1 *For example, we use adenylosuccinate lyase (ADSL) gene to detect our alternative splicing variant's probability*

Adenylosuccinate lyase (ADSL) gene										
SN	*Variants*	*Exon1*	*Exon2*	*Exon3*	*Exon4*	*Exon5*	*Exon11*	*Exon12*	*Exon13*	*Probability*
1	$1 \times 3 \times 5 \times 12$	0.333	0.000	0.051	0.000	0.333	0.000	0.282	0.000	0
2	2×5	0.000	0.316	0.000	0.000	0.684	0.000	0.000	0.000	0
3	2×4	0.000	0.545	0.000	0.455	0.000	0.000	0.000	0.000	0.152777778
4	11×13	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.305555556
5	$1 \times 3 \times 5$	0.464	0.000	0.071	0.000	0.464	0.000	0.000	0.000	0
6	$3 \times 5 \times 12$	0.000	0.000	0.077	0.000	0.500	0.000	0.423	0.000	0
7	1×3	0.867	0.000	0.133	0.000	0.000	0.000	0.000	0.000	0.208333333
8	3×5	0.000	0.000	0.133	0.000	0.867	0.000	0.000	0.000	0
9	5×12	0.000	0.000	0.000	0.000	0.542	0.000	0.458	0.000	0.333333333

Many possible progressive directions of our system remains to be exploited. In particular, the linear programming system could produce several set of possible results based on the given loose constraints. Such a situation is most commonly resulted by too many alternative splicing forms suggested by undecided biological data. It is thus desirable for human intervention to incorporate some guided information from additional biological knowledge, and to adjust our model of alternative splicing.

Table 2 represent all possible alternative splicing site of the adenylosuccinate lyase (ADSL) gene. Theoretically two types of alternative splicing events might exist, one generated randomly and one generated through regulated process. Spurious events are expected to occur at lower frequencies than regulated events because biological processes have inherent error rates that are difficult to quantify and could be highly variable. In order to avoid the danger of eliminating biologically meaningful information, we will conserve all possible alternative splicing variant and its probability. In the future, we plan to implement our algorithm and calculate all probabilities of alternative splicing variant. Our final destination is providing the program for biologists on our website (Forum n.d.) and our system will build at http://bioinfo.cs.pu.eud.tw/ASF/.

Table 2 *All possible alternative splicing sites (ASS) within adenylosuccinate lyase (ADSL) gene obtain form EST database (Hsu et al., 2004)*

Adenylosuccinate lyase (ADSL) gene				
Exon	*Segment start*	*Segment end*	*Length*	*Count of EST start*
1	19957347	19957499	152	13
2	19960620	19960823	203	6
3	19963861	19963905	44	2
4	19965036	19965115	79	5
5	19969652	19969823	171	13
6	19970048	19970094	46	4
7	19971190	19971280	90	10
8	19972061	19972130	69	7
9	19972276	19972423	147	25
10	19973769	19973859	90	22
11	19975064	19975153	89	22
12	19975668	19975844	176	11
13	19977224	19977535	311	0

ACKNOWLEDGEMENTS

The authors are sincerely grateful to Dr. Yin-Te Tsai for constructive comments and valuable suggestions. We thank Dr. Fang-Rong Hsu for providing the experimental data in their EST database. Most of all, we thank Dr. Hwan-You Chang, Dr. Hwei-Ling Peng and her PhD student Ying-Tsong Chen for biological knowledge about EST and alternative splicing. Finally, we thank many instructive discussions and information regarding the biological processes of alternative splicings within the Providence University Bioinfomatics forum (Forum n.d.). The work is supported in part by the National Science Council, Taiwan, ROC, grant NSC 92-2213-E-126-001.

REFERENCES

Boguski, M.S. and Schuler, G.D. (1995) 'Establishing a human transcript map', *Nature Genetics*, Vol. 10, pp.369–371.

Burke, J., Wang, H., Hide, W. and Davison, D. (1998) 'Alternative gene form discovery and candidate gene selection from gene indexing projects', *Genome Res.*, Vol. 8, pp.276–290.

Coward, E., Haas, S. and Vingron, M. (2002) 'Splicenest: visualizing gene structure and alternative splicing based on EST clusters', *Trends Genet.*, Vol. 18, pp.53–55.

Forum, PUB (nd), http://bioinfo.cs.pu.edu.tw.

Grama, A., Gupta, A., Karypis, G. and Kumar, V. (2003) *Introduction to Parallel Computing, 2/E*, Addison Wesley.

Graveley, B. (2001) 'Alternative splicing: increasing diversity in the proteomic world', *Trends Genet.*, Vol. 17.

Heber, S., Alekseyev, M., Sze, S-H., Tang, H. and Pevzner, P.A. (2002) 'Splicing graphs and est assembly problem', *Bioinformatics*, pp.181–188.

Heuze, N., Olayat, S., Gutman, N., Zani, M-L. and Courty, Y. (1999) 'Molecular cloning and expression of an alternative hklk3 transcript coding for a variant protein of prostate-specific antigen', *Cancer Res.*, Vol. 59, pp.2820–2824.

Hide, W.A., Babenko, V.N., van Heusden, P.A., Seoighe, C. and Kelso, J.F. (2001) 'The contribution of exon-skipping events on chromosome 22 to protein coding diversity', *Genome Res.*, Vol. 11, pp.1848–1853.

Hsu, F.R., Shih, M-Y. and Liu, J-H. (2004) 'Genome-wide alternative splicing events detection through analysis of large scale ESTs', *IEEE Fourth Symposium on Bioinformatics and Bioengineering*, pp.310–318.

Jaeken, J. and van den Berghe, G. (1984) 'An infantile autistic syndrome characterised by the presence of succinylpurines in body fluids', *Lancet*, Vol. 2, pp.1058–1061.

Kan, Z., Rouchka, E.C., Gish, W.R. and J, D. (2001) 'Gene structure prediction and alternative splicing analysis using genomically aligned EST', *Genome Res.*, Vol. 11, pp.889–900.

Kan, Z., States, D. and Gish, W. (2002) 'Selecting for functional alternative splices in ESTs', *Genome Res.*, Vol. 12, pp.1837–1845.

Kmoch, S., Hartmannova, H., Stiburkova, B., Krijt, J., Zikanova, M. and Sebesta, I. (2002) 'Human adenylosuccinate lyase (ADSL), cloning and characterization of full-length cDNA and its isoform, gene structure and molecular basis for ADSL deficiency in six patients', *Hum. Mol. Genet.*, Vol. 9, pp.1501–1513.

Lopez, A. (1998) 'Alternative splicing of pre-mrna: developmental consequences and mechanisms of regulation', *Annu. Rev. Genet.*, Vol. 32, pp.279–305.

Mironov, A. and Gelfand, J.F.M. (1999) 'Frequent alternative splicing of human genes', *Genome Research*, Vol. 9, pp.1288–1293.

Modrek, B. and Lee, C. (2001) 'A genomic view of alternative splicing', *Nature Genet.*, Vol. 30, pp.13–19.

Snir, M., Otto, S.W., Huss-Lederman, S., Walker, D.W. and Dongarra, J. (1996) *MPI: The Complete Reference*, MIT Press.

Mulyukov, Z. and Pevzner, P.A. (2002) 'Euler-*pcr*: finishing experiments for repeat resolution', *Proceedings of the Pacific Symposium on Biocomputing*, pp.199–210.

Paquola, A.C., Nishyiama Jr., M. Reis, E.M., da Silva, A.M. and Verjovski-Almeida, S. (2003) 'ESTweb: bioinformatics services for EST sequencing projects', *Bioinformatics*, Vol. 19, pp.1587–1588.

Pevzner, P.A. and Tang, H. (2001) 'Fragment assembly with double-barrelled data', *Bioinformatics*, Vol. 1, pp.S225–S233.

Pevzner, P.A., Tang, H. and Waterman, M.S. (2001a) 'A new approach to fragment assembly in DNA sequencing', *Proceedings of the Fifth Annual International Conference on Computational Molecular Biology*, pp.256–267.

Pevzner, P., Tang, H. and Waterman, M. (2001b) 'An Eulerian path approach to DNA fragment assembly', *Proc. Natl. Acad. Sci.*, USA, Vol. 98, pp.9748–9753.

Rao, V.N. and Kumar, V. (1987a) 'Parallel depth first search, part II: Analysis', *Int. J. Parallel Programming*, Vol. 16, pp.501–519.

Rao, V.N. and Kumar, V. (1987b) 'Parallel depth first search, part I: implementation', *Int. J. Parallel Programming*, Vol. 16, pp.479–499.

Rehm, B. (2001) 'Bioinformatic tools for DNA/protein sequence analysis, functional assignment of genes and protein classification', *Appl. Microbiol. Biotechnol.*, Vol. 57, pp.579–592.

Schuler, G.D. (1997) 'Pieces of the puzzle: expressed sequence tags and the catalog of human genes', *J. Mol. Med*., Vol. 75, pp.694–698.

Schuler, G.D. *et al*. (1996) 'A gene map of the human genome', *Science*, Vol. 274, pp.540–546.

Sebesta, I., Krijt, J., Kmoch, S., Hartmannova, H., Wojda, M. and Zeman, J. (1997) 'Adenylosuccinase deficiency: clinical and biochemical findings in 5 Czech patients', *J. Inherit. Metab. Dis.*, Vol. 20, pp.343–344.

Thanaraj, T., Clark, F. and Muilu, J. (2003) 'Conservation of human alternative splice events in mouse', *Nucleic Acids Res.*, Vol. 31, pp.2544–2552.

Wheeler, D.L., Chappey, C., Lash, A.E., Leipe, D.D., Madden, T.L., Schuler, G.D., Tatusova, T.A. and Rapp, B.A. (2003) 'Database resources of the national center for biotechnology', *Nucl. Acids Res*., Vol. 31, pp.28–33.

Xie, H., Yong Zhu, W., Wasserman, A., Grebinskiy, V., Olson, A. and Mintz, L. (2002) 'Computational analysis of alternative splicing using EST tissue information', *Elsevier Science*, USA, Vol. 80, pp.326–330.

Xu, Q., Modrek, B. and Lee, C. (2002) 'Genome-wide detection of tissue-specific alternative splicing in the human transcriptome', *Nucleic Acids Res.*, Vol. 30, pp.3754–3766.