# Improving the Efficiency and Accuracy of Aligning Erroneous mRNAs

TW Lam * WK Sung † Terence Yim* SM Yiu*

*Department of Computer Science
University of Hong Kong, Hong Kong

†Department of Computer Science
National University of Singapore, Singapore

*Abstract*: Locating the coding regions in a genome is one of the critical steps in understanding the genes in the genome. Aligning mRNAs with the whole genome provides clues about such locations. In eukaryotic genes, this alignment problem is complicated by the exon/intron structures. BLAT [6], one of the most popular mRNA alignment tools, was developed to address this issue. In general, for error-free mRNAs, BLAT is very efficient and accurate. However, the performance of BLAT degrades on two types of input: erroneous mRNAs and mRNAs without enough distinguishable short markers. Errors (about 10% in practice) in the mRNA as well as the genome sequence are rather common. Our experiments reveal that BLAT fails to locate mRNAs with 10% errors even if it runs in the "exhaustive" mode (i.e., a more accurate but much slower mode). For error-free mRNAs, we found that the efficiency of BLAT cannot be maintained in some rare cases which are characterized by mRNAs containing not enough distinguishable short markers; in such cases, BLAT has to run in the exhaustive mode and requires about 8 minutes per mRNA.

In this paper, we give an efficient algorithm to align the above two types of mRNAs accurately. Our algorithm makes use of approximate string matching technique based on suffix tree and takes advantage of an effective structural filtering procedure. In our experiments, our algorithm consistently outperforms BLAT on these hard cases. In particular, our algorithm is able to locate all erroneous mRNAs (with the exact exon/intron structures); for those error-free mRNAs without enough distinguishable short markers, our algorithm takes 28 seconds per mRNA on average (i.e., 16 times faster than BLAT).

*Key-words:* mRNA alignment, gene structures, exon/intron boundaries, erroneous mRNAs

## 1 Introduction

Locating the coding regions in a genome is one of the critical steps towards a better understanding of the genes in the genome. Aligning mRNAs with the whole genome provides clues about such locations. In eukaryotic genes (our paper focuses on these genes), this alignment problem is complicated by the exon/intron structures: each mRNA is, in fact, formed by the concatenation of a number of regions (called exons) in the original genome and between any two consecutive exons, there is a (possibly large) segment of nucleotides (called introns). So, the alignment between a mRNA and a eukaryotic genome usually contains a number of big gaps (introns). Most of the generic alignment tools (e.g. BLAST [1, 2], FASTA [10]) are not appropriate in this mRNA-genome alignment.

Sim4 [4] and BLAT [6] were developed to address this issue of introns. Sim4 has been shown to be slower than BLAT and is not as accurate as BLAT [5, 6]. In general, BLAT is efficient and accurate. We have tested BLAT with 26,000 error-free mRNAs, these mRNAs are extracted from the human genome based on 26,000 known genes.[1] The average length of such mRNAs is about 3000. In most cases, BLAT can align the mRNA ac-

[1]The information of these genes are based on the Human Genome Reference DNA Sequence build 34 obtained from ftp://ftp.ncbi.nih.gov/genomes/H_sapiens.

curately in the "default" mode, using about 0.5 seconds per mRNA; for some rare cases, accurate alignment can be achieved only if BLAT is adjusted to run in the "exhaustive" mode,[2] which is more accurate but much slower.

In real cases, the input mRNAs and the genome sequence may contain errors (in practice, we would assume that there are about 10% errors). We found that the accuracy of BLAT degrades drastically when erroneous mRNAs are used as input. In our experiments, we implanted 10% errors in 30 mRNAs, BLAT fails to locate most of these mRNAs even if we switch it to the "exhaustive" mode (see Table 1). And on average, BLAT takes about 5.5 minutes to align each of these mRNAs.

| Match % | Number of mRNAs located | | |
|---|---|---|---|
| | BLAT (Default) | BLAT (Exhaustive) | Our Program |
| 0% | 17 | 0 | 0 |
| 1%-10% | 9 | 23 | 0 |
| 11%-20% | 2 | 2 | 0 |
| 21%-30% | 2 | 0 | 0 |
| 31%-40% | 0 | 1 | 0 |
| 41%-50% | 0 | 1 | 0 |
| 51%-60% | 0 | 0 | 0 |
| 61%-70% | 0 | 2 | 0 |
| 71%-80% | 0 | 1 | 0 |
| 81%-90% | 0 | 0 | 30 |
| 91%-100% | 0 | 0 | 0 |
| Ave. time (per mRNA) | 2 secs | 5.5 mins | 3.8 mins |

Table 1: Performance of our program and BLAT on 30 erroneous mRNAs (Match % shows the percentage of the matched nucleotides between the mRNA and the located region in the genome). Note that since there are 10% errors in the mRNAs, the highest possible match % is 90%.

In this paper we also investigate those error-free mRNAs that make the performance of BLAT degrade. Out of the 26,000 error-free mRNAs, there are 488 mRNAs for which BLAT was not able to locate the exact exon/intron structures in its default mode (see Table 2) A detailed study reveals that these mRNAs have the characteristics that there are not enough distinguishable short markers in the mRNAs. In other words, short substrings (of length 10 or less ) of these mRNAs usually have quite a number of copies in the genome sequence. Thus, BLAT fails to identify a small set of potential locations for checking the exact location for the mRNAs, and BLAT has to run in the exhaustive mode (using about 8 minutes per mRNA) in order to align these mRNAs correctly.

In summary, there are two types of mRNAs that make the performance of BLAT degrade: (1) erroneous mRNAs for which BLAT is not able to locate the mRNA even in "exhaustive" mode, and (2) mRNAs that do not have enough distinguishable short markers for which BLAT has to use the "exhaustive" mode in order to successfully locate these mRNAs.

| Match % | Number of mRNAs located | | |
|---|---|---|---|
| | BLAT (Default) | BLAT (Exhaustive) | Our Program |
| 1%-10% | 2 | 0 | 0 |
| 11%-20% | 13 | 0 | 0 |
| 21%-30% | 38 | 0 | 0 |
| 31%-40% | 44 | 0 | 0 |
| 41%-50% | 74 | 0 | 0 |
| 51%-60% | 93 | 0 | 0 |
| 61%-70% | 101 | 0 | 0 |
| 71%-80% | 123 | 0 | 0 |
| 81%-90% | 330 | 0 | 0 |
| Over 90% | 0 | 488 | 488 |
| Ave. Time (per mRNA) | 1.5 secs | 8 mins | 28 secs |

Table 2: Performance of our program and BLAT on 488 error-free mRNAs without enough distinguishable short markers.

**Our Contributions:** In this paper, we propose an efficient and accurate alignment algorithm to work on erroneous mRNAs, as well as mRNAs without enough distinguishable short markers. Our algorithm makes use of suffix tree and approximate string matching technique to capture a complete set of candidates for possible alignment. Then we exploit some structural property of these candidates to effectively filter out most of the incorrect candidates. This is the reason why our algorithm can improve accuracy and efficiency at the same time. We have conducted experiments to compare our algorithm against BLAT on the two types of problematic inputs, confirming that our algorithm

[2]For reference, we modify parameters like `MAXDnaHits` to be `MAX_INT`.

does perform better. In particular, for the erroneous input mRNAs, our algorithm can locate all these mRNAs correctly (see Table 1 for details). For those mRNAs without enough distinguishable short markers, our algorithm is able to align them correctly about 16 times faster than BLAT (see Table 2 for details).

**Organization of the paper:** The rest of the paper is organized as follows. Section 2 describes our algorithm. The experimental results are presented in Section 3. Section 4 concludes the paper.

**Remark:** Aligning ESTs against the genome sequence is also an useful application in identifying the location of coding regions. ESTs are usually contiguous nucleotide sequences coming from a single exon or at most two exons, our algorithm and BLAT can also work on ESTs. However, the focus of our paper is on aligning a full mRNA which consists of multiple exons, so we choose to compare our algorithm with BLAT and the software tools designed mainly for EST alignment [3, 5, 7, 9] are not considered in this paper. In fact, most of these tools have only been evaluated by using ESTs, but not mRNAs.

# 2 Our Algorithm

Our algorithm consists of 3 phases in aligning a given mRNA to the genome. Roughly speaking, in Phase 1 (Candidate Generation), we partition the mRNA into substrings and search the occurrences of these substrings in the genome. These occurrences, called candidates, provide information on the possible locations of the mRNA. To ensure the accuracy, unlike some other algorithms, we aim at generating all possible candidates. However, checking all these locations may take a long time. Obviously, most of these occurrences are not related to the correct location of the mRNA. By exploiting the relationship between the ordering of the substrings and the positions of the corresponding candidates, we develop a very tight filtering condition (see Lemma 1) in Phase 2 (Filtering) to filter out the incorrect candidates which enables us to eliminate about 96% of the candidates. The remaining candidates will be evaluated in Phase 3 (Finding the alignment) to locate the correct exon/intron structures.

To handle erroneous mRNAs, in Phase 1, we do not require the occurrences of a substring to be an exact match of the substring. However, the number of candidates reported will be increased drastically. The filtering in Phase 2 is critical in eliminating most of the incorrect candidates. These two features enable our algorithm to handle erroneous mRNAs efficiently and more accurately. Note that if the input mRNA and the genome sequence are of high quality, the requirement of an exact match can be imposed to speed up the process. In this case, the filtering still plays a key role in eliminating candidates.

In the following context, let $P$ be the mRNA of length $m$ and $G$ be the genome. In addition, we have the following assumptions: (1) An exon contains at least 100 nucleotides[3] and (2) The error in each exon (in mRNA as wellas in the genome sequence) is at most 10%.

## 2.1 Phase 1 - Candidates Generation

We partition $P$ into $j$ substrings, $P_1$, $P_2$,..., $P_j$, each containing $L = \lfloor m/j \rfloor$ characters. For each $P_i$, we locate all substrings $S$ in $G$ such that the edit distance between $P_i$ and $S$ is at most $e$. We say that $S$ is located by $P_i$. Each substring $S$ is called a candidate and is represented by a tuple $(i, a, e)$ where $a$ is the starting position of $S$ in $G$.

In our implementation, we set $e = 1$ for erroneous mRNAs and $e = 0$ for high quality mRNAs. Recall that we aim at generating all possible candidates, we use an approximate string matching technique based on suffix tree [8]. By storing only the top levels (say 13 levels) of a suffix tree in memory and a suffix array in harddisk, we solve the issue of memory limitation which enables the algorithm to run on a PC.

## 2.2 Phase 2 - Filtering

The basic idea of the filtering algorithm is as follows. A candidate $(k, b, e)$ is a supporting candidate for another candidate $(i, a, e)$ if $|k - i| \times L \leq |b - a| \leq |k - i| \times L + e * |k - i - 1|$. Intuitively,

[3]This has been verified to be valid in most real data.

the two candidates are supporting candidates if they are related to the correct location of the same exon. Recall that each exon is assumed to contain at least 100 nucleotides and the error in each exon is at most 10%. By taking $L = 12$ ($L$ is the partition length) and based on the pigeon principle, we can show that the following lemma holds.

**Lemma 1.** *Let $E$ be an exon in $P$ with at least 100 nucleotides. Then, $E$ includes 7 or more 12-nucleotide partitions and these partitions can locate at least 4 supporting candidates.*

Based on the above lemma, for each candidate, we look for 3 more supporting candidates, if found, we keep the candidate, otherwise we throw it away. Note that these supporting candidates may not be consecutive. After the filtering, we can guarantee that 4 supporting candidates are left for each exon. Thus, we have the following lemma.

**Lemma 2.** *Let $E$ be an exon in $P$. Let $P_k$, $P_{k+1}$, ..., $P_{k+\ell-1}(\ell \geq 7)$ be the partitions included in $E$. Then, at least 4 supporting candidates located by these partitions are left after filtering.*

### 2.3 Phase 3 - Finding the Alignment

In this phase, we first cluster the remaining candidates based on their locations in the genome and the positions of the corresponding substrings in the mRNA. Basically, each cluster corresponds to an exon and alignment is performed inside the cluster and by extending the boundaries of the cluster. The clusters are then chained up together to form a possible location of the mRNA. The chained clusters that give the highest match percentage will be reported. For each exon, we can show that we can recover the location of the mRNA since at least one of the candidates is left after Phase 2. The details of this phase will be given in the full paper.

## 3 Experimental results

In this section, we evaluate the performance of our algorithm against that of BLAT on two sets of mRNAs: the erroneous mRNAs and the mRNAs without enough distinguishable short markers. We also show the effectiveness of our filtering procedure (Phase 2 of our algorithm). We implemented our algorithm in C and conducted our experiments on a Intel-based PC with a Pentium IV 2.4GHz CPU, 4GB main memory running in SunOS 5.8.

### 3.1 Erroneous mRNAs

We generate the erroneous mRNAs as follows. Based on the mRNAs in the GenBank published in the Human Genome Reference DNA Sequence build 34, we selected 30 mRNA sequences and introduced 10% errors to each of them. The 10% errors are introduced by mutating one nucleotide chosen randomly per every ten nucleotides. For each mRNA, we measure the percentage of matched nucleotides in the alignment of the mRNA and the located region as reported by the programs (we call this the match %). The results of the experiments is shown in Table 1. It is clear that our program is more accurate than BLAT. In terms of efficiency, our algorithm takes 3.8 minutes on average per mRNA alignment while BLAT takes 5.5 minutes in the exhaustive mode. Note that the exon/intron boundaries reported by our algorithm are all exact locations while some of the exon positions reported by BLAT are not correct.

### 3.2 mRNAs without enough distinguishable short markers

We tested BLAT with about 26000 mRNAs obtained from GenBank. These mRNAs are error free. We found that there are 488 of them for which BLAT (running in the default mode) can only report a match percentage of $\leq 80\%$ (with 300 of them have a match percentage $\leq 60\%$). In other words, BLAT fails to locate the correct exon/intron structures in these mRNAs in its default mode. These mRNAs are characterized by having a relatively high percentage of short substrings that have many occurrences in the genome. From a preliminary data analysis, we found that in these mRNAs, about 80% of the substrings of length 10 have more than 8000 occurrences in the genome while for the other mRNAs, only about

50% of the substrings have more than 8000 occurrences in the genome.

To assess our program's ability for aligning mRNAs without enough distinguishable short markers, we used these 488 mRNAs as inputs to our program. Our program can align all of them correctly with 28 seconds per mRNA[4]. In contrast, if BLAT runs in exhaustive mode, although it can align all the mRNAs with match percentage higher than 90%, it takes around 8 minutes to align each mRNA. Table 2 shows the results.

### 3.3 Filtering effectiveness

We measure the effectiveness of our filtering procedure by measuring the percentage of candidates generated in Phase 1 of our algorithm that are filtered away by our filtering procedure (we call this the effectiveness index).

Based on the two sets of experiments we have conducted, we found that the effectiveness index of our filtering procedure is about 96%. More precisely, for erroneous mRNAs, the effectiveness index is 98%, while for mRNAs without enough distinguishable short markers, the effectiveness index is 95%. In other words, the filtering is effective in both cases. Moreover, the filtering procedure takes only 20%, on average, of the total running time of the whole algorithm.

## 4 Conclusion

In this paper, we propose an efficient and more accurate algorithm for aligning erroneous mRNAs for which BLAT is not able to align them correctly. We also identified a set of mRNAs characterized by not having enough distinguishable short markers that make BLAT run slower in order to align them while our algorithm can align them efficiently and accurately. Integrating BLAT and our algorithm would be a sensible future work.

[4]Our program is switched to find exact match candidates in Phase 1 for this set of experiment.

*References:*


[1] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[2] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

[3] E. Birney and R. Durbin. Dynamite: A flexible code generating language for dynamic programming methods used in sequence comparison. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, pages 56–64, 1997.

[4] Liliana Florea, George Hartzell, Zheng Zhang, Gerald M. Rubin, and Webb Miller. A computer program for aligning a cdna sequence with a genome DNA sequence. *Genome Research*, 8:967–974, 1998.

[5] F.R. Hsu and J.F. Chen. Aligning ESTs to genome using multi-layer unique makers. In *Proceedings of the Second IEEE Computer Society Bioinformatics Conference (CSB'03)*, pages 564–566, Stanford, CA, USA, 2003.

[6] W. James Kent. BLAT - the BLAST-like alignment tool. *Genome Research*, 12:656–664, 2002.

[7] R. Mott. Est_genome: A program to align spliced DNA sequences to unspliced genomic DNA. *Computer Applications in the Biosciences*, 13:477–478, 1997.

[8] Gonzalo Navarro and Ricardo Baeza-Yates. A new indexing method for approximate string matching. In *Proceedings of the 10th Annual Symposium on Combinatorial Pattern Matching*, pages 163–185, 1999.

[9] Jun Ogasawara and Shinichi Morishita. Practical software for aligning ests to human genome. In *Proceedings of the Thirteenth Annual Symposium on Combinatorial Pattern Matching*, pages 1–16, Fukuoka, Japan, 2002.

[10] W.R. Pearson and D.J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Acad. Sci.*, 85:2444–2448, 1988.