Aligning ESTs to Genome Using Low Frequency High Density Index

F. R. Hsu Shu Ying Sue

Department of Information Engineering and Computer Science, Feng Chia University, Taiwan

frhsu@fcu.edu.tw *shuyingsue@gmail.com*

ABSTRACT

The development of computer and information technology facilitates many researches of biology that employ computer software. In this paper, we provide a method that can align the ESTs to the genome. Yet, the human genome contains repetitive sequences that hold one-tenth of the human genome. And in the past, most of the associated researches cannot handle those repetitive sequences well, and even cannot deal with those sequences. Hence, our research hopes to handle both those repetitive and unique sequences in the genome to make all ESTs can be aligned to the correct regions. In this way, we can employ the results to have an advance research and analysis. We provide different strategies that can save time and get results within an acceptable correctness to align a single EST and the entire ESTs in dbEST to the genome. We consider the low frequency and high density index problem to provide the EST to locate to the genome. And then, we propose a heuristic algorithm and employ MUGUP to check our research with different test sets of ESTs.

1: Introduction

Although the Human Genome Project (HGP) was completed in 2003, the data will keep on being analyzed for many years. However, with the developmental bioinformatics and progress of related technology, some researches like genomic analysis can be accelerated by using those tools.

In genome research, mapping and aligning cDNA sequences to the genome is important. The cDNA sequences contain both message RNAs(mRNAs) and expressed sequence tags (ESTs). An EST is a short sequence of DNA. It can be taken from a cDNA copy of an mRNA. The EST can be used to discover the mechanisms of life such as alternative splicing site decision, gene structure prediction [5] (Hsu et al., 2004), single nucleotide polymorphism (SNP) [2] (Chen et al., 2002) and so on.

Take an EST for example, we need to decompose the EST into exons and align each exon to the genome sequence. However, mapping and aligning of the millions of ESTs in dbEST to the human genomic sequence is an arduous task. Because ESTs are a maximum of tens of thousands of bases long, and the human genomic sequences are about 3 billon bases long. Hence, pre-processing the genomic sequences is important. If the genomic sequences are not pre-processed, it will cost much calculation time to align a long EST to the genome. In order to shorten the overall calculation time, the genomic sequences should be pre-processed. All fast alignment tools can be divided into two parts. Fist, a "search stage," to detect regions of the two sequences which are probable to be homologous. And then, an "alignment stage" is executed to check those regions and produce alignment for those regions which are indeed homologous according to some criteria. BLAT [6] (Kent, 2002), SQUALL [7] (Needleman et al., 1970), MUGUP [4] (Hsu et al., 2003) and GMAP [9] (Wu et al., 2005) are tools that pre-process the genomic sequence before the alignment.

However, SIM4 [3] (Florea et al., 1998), and Spidey [8] (Wheelan et al., 2001) are alignment tools that do not pre-process the genomic sequence. They will cost much calculation time to align the cDNA to the genome. Thus, BLAST [1] usually is used to pre-process the genomic sequence to determine the HSPs before executing SIM4 or Spidey.

Although pre-processing the genomic sequences can reduce the time in aligning the cDNA to the genome, it is also important to process the genomic sequences adequately. For example, SQUALL [7] (Needleman et al., 1970) finds all 21-mers occur in the genome and create a MapTable to store the occurrences and positions. Then, for an EST, SQUALL searches 21-mers that are StartSet and EndSet both ends of the EST in the MapTable to combine the possible positions of the EST in the genome. However, mapping the EST just use both ends of the EST may be not accurate enough. The sequencing errors may result in the EST be mapped in wrong positions. In addition, MUGUP [4] (Hsu et al., 2003) processes the genomic sequences by finding all unique markers (UMs) [2] (Chen et al., 2002) that each one is a subsequence that occurs once in the genome and creates an index table. For an EST, MUGUP will map the EST by the UMs occur in the EST. However, if an EST does not contain any UMs, MUGUP cannot map and align it. In other words, if all the markers occur in an EST are repetitive, MUGUP cannot handle the sequence. However, if MUGUP align ESTs by using

all-marker(AM) table that contain information about all markers in the genome, it may cost much time to complete it as a large number of ESTs need to be aligned.

Hence, in this paper, we will consider the repetitive sequences in the genome and we will propose an adequate algorithm to pre-process the genomic sequences. Besides, we are interested about the EST to genome alignment problem in this research.

In this paper, we want to deal with two problems. First, we expect to deal with the problem of single EST alignment. Second, we expect to discover a strategy for aligning the entire ESTs in dbEST to the genome. For the two problems, our objective is to get the result with high correctness within reasonable time. Due to both the large amount of the human genome and ESTs, it is a difficult task to align the EST(s) to the genome for related research. If we use all the occurrences in the genome to be an index, the cost of aligning the EST(s) may be unexpected. However, if we just use all unique occurrences in the genome to be an index, there are some regions in the genome cannot be aligned. Thus, a strategy for selecting an appropriate index to represent the genome is a way to deal with the problem of ESTs alignment.

Hence, we design an algorithm to select the index to map and align the EST to the genome. The index generated by our algorithm is called low-frequency and high-density (LFHD) index. We will define the LFHD index problem in section 2.1 and present the LFHD index selection algorithm in section 2.2. Finally, we will present the information about LFHD index with an adjustable parameter K in section 3.1, the comparison with MUGUP [4] and GMAP [9] in section 3.2 and the results of aligning the single EST to genome and the entire ESTs in dbEST to genome in section 3.3 and 3.4 separately.

2: Methods

In this section, we describe how we pre-process the genomic sequences by implementing our algorithm. First, we will define the LFHD index problem in section 2.1, and then we present the algorithm step by step in section 2.2.

2.1: LFHD index problem

Consider the LFHD index problem." Given a genome, $G=\{g_1, g_2, g_3, ..., g_m\}(g_a \in \{A, T, C, G\})$, and we expect to get an LFHD index of genome, $M = \{m_1, m_2, m_3, ..., m_n\}(m_b \in k\text{-mer in } G$, and k-mer is a consecutive sequence that is k bases long in $G)$. And fitting that the sum of $f(i_b)$ is minimum, where $f(i_b)$ is the frequency of i_b in the genome, and $D(i_b, i_{b+1}) \leqq K$, where $D(i_b, i_{b+1})$ is the distance between adjacent i_b and i_{b+1} and K is a threshold of a distance. Because this is a difficult problem, we think it may be an NP-Complete problem. Thus, we propose a heuristic algorithm to handle the "LFHD index problem."

2.2: LFHD index selection algorithm

In this research, the property of UM [2] is used as a foundation to proceed to the algorithm. In the beginning, a sufficient large value of k as the length of UM, denote as UM_k needs to be determined. In MUGUP [4], they discovered that when the length of UM is increased to 28, the number of UMs in the human genome saturates. The number increases rapidly from the length of UMs equals to 15 to 23. Hence, we choose k as 28 and we will find all the UMs that each is 28 bases long in the human genome, denote as UM_{28}.

Let I denote the set of intervals where the distance of two consecutive selected markers is larger than K. Consider a marker m. Let $freq(m)$ denote its frequency in the genome. Let $count(m,I)$ denote the number of appearance of m in current I. Let $R(m)$ denote $count(m,I) \diagup freq(m)$.

Step 1, find all the UM_{28} in the genome and then count the intervals between the two adjacent UM_{28}. If the interval is bigger than a threshold value K, add it to a set of intervals I. After scanning all the intervals, I is constructed.

Step 2, search the 28-mer that occur only twice in the genome, denoted as 2M. For each 2M, if it occurs twice in I, add it to the set of index foremost. At the same time, I will be updated. When selecting a marker that exists in an interval, the interval can be separated into two parts by the marker. Thus, if the interval is small than K, the interval needs to be removed from I. After that, choose the 2M occur only once in I and update I.

Step 3, search the 28-mer that occurs thrice in the genome, denoted as 3M. Select the 3M occurs thrice in I, and then select the 3M occurs twice in I. Finally, choose the 3M occurs once in I. Similarly, I need to be updated when selecting a marker.

Step 4, after dealing with the 2Ms and 3Ms, give each marker a ratio R. Then, limit R to equal to 0.75. Add those markers that fit the limitation and update I. After that, limit R to equal to 0.5 and repeat the same action.

Step 5, scan the remaining intervals in I and select remaining markers exist in I. After doing this step, the selection of the LFHD index is finished.

Step 6, collect the LFHD index, and employ MUGUP to construct the table of LFHD index for mapping and aligning.

When completing the above steps, the algorithm terminates. Note that K is determined by user and the fit values will be proposed in the next section.

3: Results

In this section, we present the result of selecting the LFHD index with different K-values and the number of selected markers in particular steps in the algorithm. And we compare LFHD index selection algorithm by using different K-values with MUGUP [4] (Hsu et al., 2003). In addition, we propose the results of aligning the single

ESTs to genome and the entire ESTs in dbEST to genome. In this paper, we utilize two test sets to execute our algorithm.

(1) Employ UM table of MUGUP to align ESTs to the genome. And randomly choose 1,000 ESTs that the UM table cannot align to be a set. (2) Randomly choose 24,000 ESTs that the alignment score are less than 94 to be a set.

3.1: LFHD index selection algorithm with different *K*-values

We use *K* equals to 30, 40 and 50 to execute our algorithm. First, we count the two adjacent UMs in the human genome, and collect the intervals that bigger than *K*. The total length of intervals with different *K* is as shown in Table 1. When *K* equals to 30, 12% of the genomic sequences are repetitive. When *K* equals to 40, 11% of the genomic sequences are repetitive. And when *K* equals to 50, 10% of the genomic sequences are repetitive. In this research, we choose some markers in these repetitive regions to be the index. We show the markers selected in the algorithm with different *K*-values as shown in Table 2.

K-value	Length(base)
30	347,953,541
40	316,330,892
50	286,695,461

Table 1. The original length of intervals with different *K*-values

Table	UM	2M	3M	Remaining
AM	241,494,2414	81,647,716	29,939,688	328,698,008
K=30	24,14,942414	11,926,006	6,333,201	208,796,065
K=40	24,14,942414	9,541,774	5,036,409	53,903,763
K=50	241,494,2414	7,506,546	4,110,513	20,598,800

Table 2. The number of select markers in particular steps of our algorithm with different *K*-values

3.2: Comparisons with MUGUP and GMAP

We employ test set (1) to execute our algorithm with *K* equals to 30, 40 and 50 and the results as shown in Table 3. “Successful rate” means the percent of ESTs have alignment results, “Score” means the percentage of matching to the genome of an EST and “Not found” means an EST cannot be aligned to any regions.

Tools	Successful rate(%)	Score ≧ 94(%)	Not found(%)
K=30	83.7	42	16.3
K=40	83.1	42	16.9
K=50	83	41	17
GMAP	89.7	38.7	10.3

Table 3. The results of aligning test set (1) by LFHD index with different *K*-values and GMAP

In the previous study, MUGUP [4] (Hsu et al., 2003) run 6,868,818 ESTs and find that 210,546 ESTs cannot be aligned as shown in Figure 1. Hence, we estimate how many results we can get from those 210,546 ESTs according to Table 3.

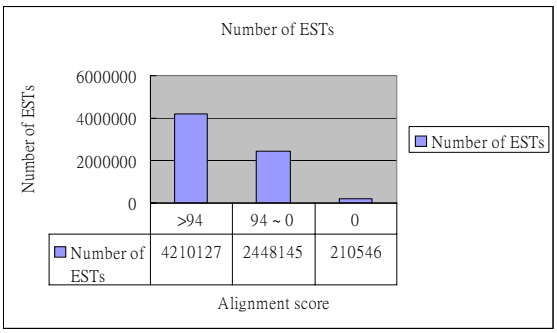

Figure 1. The number of alignment scores of 6,868,818 ESTs with different limitations

We calculate that when using *K* equals to 30, there are 176,227 ESTs can be aligned. When using *K* equals to 40, there are 174,963 EST can be aligned. When using *K* equals to 50, there are 174,753 ESTs can be aligned. And GMAP can align 188,860 ESTs. But the number of ESTs of GMAP [9] that each score is bigger than or equal to 94 are less than LFHD index with different *K*-values.

3.3: Aligning the single EST to genome

We employ test set (1) and (2) to execute our algorithm with different *K*-values. And we compare those results with AM table of MUGUP [4] (Hsu et al., 2003) as shown in Table 4 and Table 5. “Same” means that they align to the same position and have the same alignment score. “Different” means that they align to the different positions. “Not found” means that both of the two tables cannot align the EST to any regions. We can find that the correctness can be adjusted by the parameter *K*. And we compute the average time for aligning an EST to genome for the three kinds of tables as shown in Table 6. When aligning a single EST to genome with UM table, AM table, LFHD index table with *K* equals to 30,40 and 50 and GMAP [9], it will cost 4, 42, 25, 24, 20, and 17 seconds separately. We can find that when using LFHD index table with a suitable parameter *K*, it will cost much less time than aligning with AM table. Hence, we propose that aligning a single EST to genome by LFHD index with a parameter *K*. Although GMAP just cost 17 seconds to align a single EST to the genome, it cost more time to align the entire ESTs in dbEST than using our strategy.

3.4: Aligning the entire ESTs in dbEST to genome

In this research, aligning the entire ESTs in dbEST to genome is one of the important problems. We use the above results to estimate time for align large amount of ESTs to genome. According to the previous study, if we align 6,868,818 ESTs to the genome by all-marker table, it must cost 80,136 hours to complete the alignment. It takes a lot of time.

K-value	Same(%)	Different(%)	Not found(%)
30	91	8	1
40	90	9	1
50	89	10	1

Table 4. Comparison between aligning test set (1) by LFHD index table with different K-values and AM table

K-value	Same(%)	Different(%)	Not found(%)
30	92	7	1
40	91	8	1
50	89	10	1

Table 5. Comparison between aligning test set (2) by LFHD index table with different K-values and AM table

Tools	Time(sec)
UM	4
AM	42
K=30	25
K=40	24
K=50	20
GMAP	17

Table 6. The average time for aligning an EST to genome by using AM table, LFHD index table with different K-values and GMAP

Table	Estimative time(hours)	Successful rate(%)	Score≧94(%)
UM	7,632	96.90	61.30
AM	80,136	97.30	69.26
UM+ K=30	26,095	97.21	69.15
UM+ K=40	25,357	97.19	69.13
UM+ K=50	22,402	97.17	69.00
GMAP	32,436	95.70	69.50

Table 7. The estimative time for aligning the entire ESTs in dbEST to genome by employing different tools

However, if we align 6,868,818 ESTs to the genome with UM table first, and we filter out ESTs that each alignment score is less than 94. And, there are 2,658,691 ESTs that scores are less than 94. And then, we realign those ESTs to the genome with LFHD index table. Then, we can reduce more time cost than aligning with all-marker table.

When we use *K* equals to 30 to choose the LFHD index, the average time of aligning an EST cost 25 seconds. Hence, it may cost 18,463 hours to align 2,658,691 ESTs to the genome. When *K* equals to 40, the average time of aligning an EST cost 24 seconds. And it may cost 17,724 hours to align those ESTs. Besides, when we use *K* equals to 50 to choose the LFHD index, the average time of aligning an EST cost 20 seconds. And it may cost 14,770 hours to align those ESTs to the genome. The estimative time of aligning the entire EST in dbEST to the genome with UM table, all-marker table, LFHD index with different *K*-values and GMAP [9] are as shown in Table 7.

According to the estimation, we know that aligning with LFHD index table can reduce more time than all-marker table. And the correctness is acceptable. And, we can find that aligning the entire ESTs in dbEST to genome with all-marker table cost much time. This is a hard work to cost so much time to align those ESTs. And we also find that GMAP will cost more time than our strategy to get better results. In addition, if we align those ESTs with Sim4, it would cost time that we cannot expect. Sim4 do not pre-process the genome. Hence, it is not practicable to align so many ESTs with Sim4.

Hence, we propose a strategy for aligning the entire ESTs in dbEST to genome. First, we align the entire ESTs with UM table. We filter out those ESTs that each of the alignment score is less than 94. And then, we realign them with the table of LFHD index with a parameter *K*. This strategy would reduce much time to get more useful ESTs.

4: Reference

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

[2] Chen,L.Y., Lu,S.H., Shih,E.S. and Huang,M.J. (2002) Single Nucleotide Polymorphism Mapping Using Genome-Wide Unique Sequences. *Genome Research*, **12**, 1106-1111.

[3] Florea,L., Hartzell,G., Zhang,Z., Rubin,G.M. and Miller,W. (1998) A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Research*, **8**, 967-974.

[4] Hsu,F.R. and Chen,J.F. (2003) Aligning ESTs to Genome Using Multi-Layer Unique Markers. *IEEE Computational Systems Bioinformatics Conference '03*, pp. 564-567.

[5] Hsu,F.R., Chang,H.Y., Lin,Y.L., Tsai,Y.T., Peng,H.L., Chen,Y.T., Chen,C.F., Cheng,C.Y., Liu,C.H. and Shih,M.Y. (2004) Genome-Wide Alternative Splicing Events Detection through Analysis of Large Scale ESTs. *IEEE Fourth Bioinformation Symposium on Bioinformatics and Bioengineering*, pp. 310-316.

[6] Kent,W.J. (2002) BLAT – The BLAST Like Alignment Tool. *Genome Research*, **12**, 656-664.

[7] Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino

acid sequence of two proteins. *Journal of Molecular Biology*, **48**, 443-453.
[8] Wheelan,S.J., Church,D.M., and Ostell,J.M. (2001) Spidey: a tool for mRNA-to-genomic alignments. *Genome Research*, **11**, 1952-1957.
[9] Wu,T.D. and Watanabe,C.K. (2005) GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics*, **21**, 1859-1875.