Single Nucleotide Polymorphism Mapping Using Multi-Layer Unique Markers

Fang Rong Hsu[1,*] , Wei-Chung Shia[1], J. F. Chen[2], and Fang Ming Hsu[3]

[1]Department of Information Engineering and Computer Science

Feng Chia University

Taichung, Taiwan

[*]frhsu@fcu.edu.tw

[2]Department of Computer Science and Information Engineering

Asia University

Taichung, Taiwan

jfchen@asia.edu.tw

[3]Department of Information Management,

National Dong Hwa University

Hwalien, Taiwan

fmhsu@mail.ndhu.edu.tw

Received 4 June 2007; Accepted 10 August 2007

Abstract. As a great number of the genomic DNA sequences can now be generated in a day, the demand for a very fast and accurate method for positioning genomic DNA sequences on a genome is high. A unique marker is a sequence which appears only once in a genome. The unique marker method is an efficient method to perform this task. The amount of time needed for positioning genomic DNA sequences grows rapidly when we use longer unique markers. On the other hand, the success rate of short unique markers is not very high. In this paper, we propose a multi-layer genome-wide unique marker positioning (MUGUP) technology for obtaining a high rate of accuracy as well as a high speed of computation. Our method combines the benefits of both short and long unique markers. We also compared our method with two other famous methods: BLAST and SSAHA. Our method was found much faster than these two methods.

Keywords: Bioinformatics, SNP, Alignment, Multilayer Unique Marker (MUM), Indexing

1 Introduction

DNA sequences can be obtained experimentally using expressed sequence tags (EST) and single nucleotide polymorphisms (SNP). "Polymorphism" within the acronym SNP refers to the change of a nucleotide within the DNA sequence, such as going from "Adenine (A)" to "Thymine (T)". The information SNPs provides can be used to predict a patient's possible reaction to certain medicines by analyzing the link between the characteristics of the medicine and the genes of the person [1]. Many new SNP sequences are found in laboratories each day. The SNP positioning problem occurs in the step of locating the position of a SNP on the genome. Formally, the SNP positioning problem is defined as follows:

If a genomic sequence G = (g1, g2,…,gn), a SNP sequence S = (S1, S2, …, Sm), and the SNP position i on S are given, find the corresponding address of i on G.

Several methods to solve problems regarding the position of a tag on a genome have been provided, the most well known method being BLAST [2]. This method has been further improved by several researches [3] [4] [5]. In 2001, Ning et al.[6] proposed a hashing technology to solve this problem. Their method (SSAHA) runs much faster than BLAST while RAM is large enough to load the entire hash table.

A unique marker is a sequence which appears on a genome only once. For example, a 14-mers unique marker consists of 14 continuous bps appearing on the genome only once. Theoretically, one unique marker is sufficient for positioning the sequence that includes the SNP. The idea of using unique markers to map SNPs was first

* Correspondence author

suggested by Chen et al. Their unique marker (UM) method was the first that used personal computers to position the entire sequence of the human genome in dbSNP {http://www.ncbi.nlm.nih.gov/SNP}.

dbSNP is an NCBI research project, and this database to serve as a central repository for both single base nucleotide subsitutions and short deletion and insertion polymorphisms, and dbSNP distinguishes a report of how to assay a SNP from the use of that SNP with individuals and populations [7]. Hence, we also use the data from dbSNP to our experimental dataset.

Using unique markers to map SNPs on an entire genome, the amount of unique markers needs to be large enough to cover the whole genome. Generally speaking, shorter unique markers are faster in positioning; however, their success rate in positioning is lower. Based on this theory, Chen et al. recommended to the use of unique markers with a length of 15 bps as a landmark for positioning in order to obtain sufficient data in an acceptable time. But for increasing the success rate for positioning, longer unique markers must be used. Yet, the time needed grows exponentially in that case. A comparison of the efficiency between unique markers with different lengths is shown in Table 1 and Fig. 1.

Table 1. Comparison of execution time and success rates for different lengths of unique markers (adapted from Chen et al. 2002, p. 1107)

Length of unique markers	Number of unique markers on genome (Build 28)	The ratio of execution time contrast with 14-mers unique markers	Success rate
13	2,440,788	0.375	13.3%
14	30,234,168	1	57.5%
15	162,253,846	2.5	81.4%
16	646,229,602	17.5	88.3%

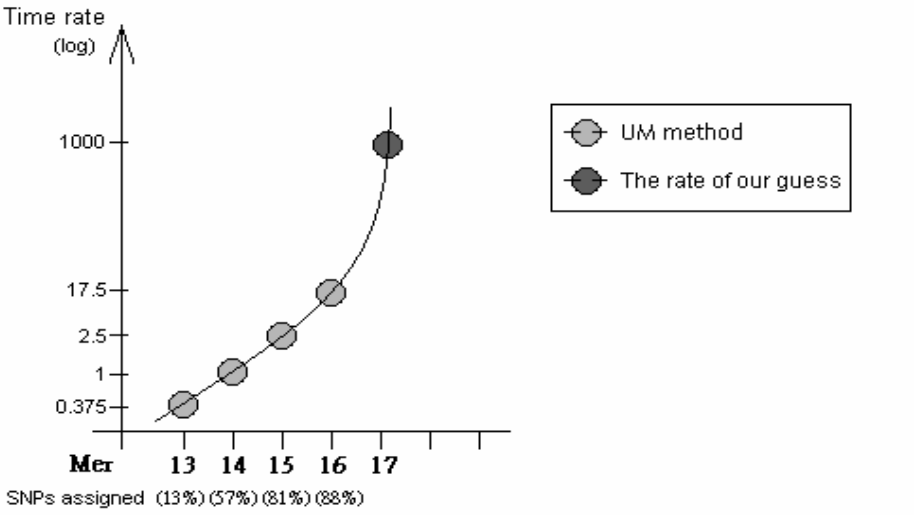

Fig. 1. Effect of lengths of unique makers on time rate

In this paper, we propose the MUlti-layer Genome wide Unique marker Positioning technology (MUGUP) to position SNPs on a genome. Our method combines the benefits of both short and long unique markers. For the human genome, we constructed a 14-mers, a 21-mers and a 28-mers unique marker table to form a 3-layer unique marker table. By using this multi-layer unique marker table, we could successfully map all SNPs in dbSNP to the genome. The time for SNP positioning by using the traditional one-layer unique marker table increases rapidly when the length of a unique marker increases. Surprisingly, by using the multi-layer unique marker table, our algorithm for solving SNP positioning problem was even faster than others that use only a 14-mers unique marker table. Besides, our SNP assigning rate was 99.9%, thus much higher than that of the 14-mers unique marker table (57.5%). Using four personal computers with Pentium III (1.0 GHz, 512M RAM), 9.0 million SNP sequences (dbSNP build 121, http://www.ncbi.nlm.nih.gov/SNP) were mapped using this method with a success rate of 99.9% in 51.9 hours.

Randomly selecting 10 SNP sequences from each chromosome, we mapped these 240 sequences using MUGUP, SSAHA and BLAST with Celeron (2.2G Hz, 1G RAM). On average, it took 0.23, 1050 and 2640 seconds to map a single sequence by using MUGUP, SSAHA and BLAST respectively. Therefore, our method is much faster than traditional methods.

2 Methods

2.1 Properties and distribution of unique markers

First, the length of unique markers used for mapping SNPs on the genome has to be decided. In the human genome build 34, there is no unique marker shorter than 10-mers. The amount of unique markers increases rapidly when more than 14-mers are used, but this rate saturates when the length is increased to more than 23-mers as shown in Table 2 and Fig. 2.

Table 2. Number of unique markers found on the human genome for different lengths of the markers

Mer	Number	Mer	Number	Mer	Number
10	857	17	1,334,437,758	24	2,363,762,938
11	242,023	18	1,802,518,644	25	2,365,922,143
12	3,287,574	19	2,039,323,565	26	2,368,517,308
13	18,771,860	20	2,147,586,807	27	2,374,140,722
14	40,261,150	21	2,194,639,732	28	2,374,587,718
15	185,253,818	22	2,317,292,056		
16	666,608,575	23	2,350,639,379		

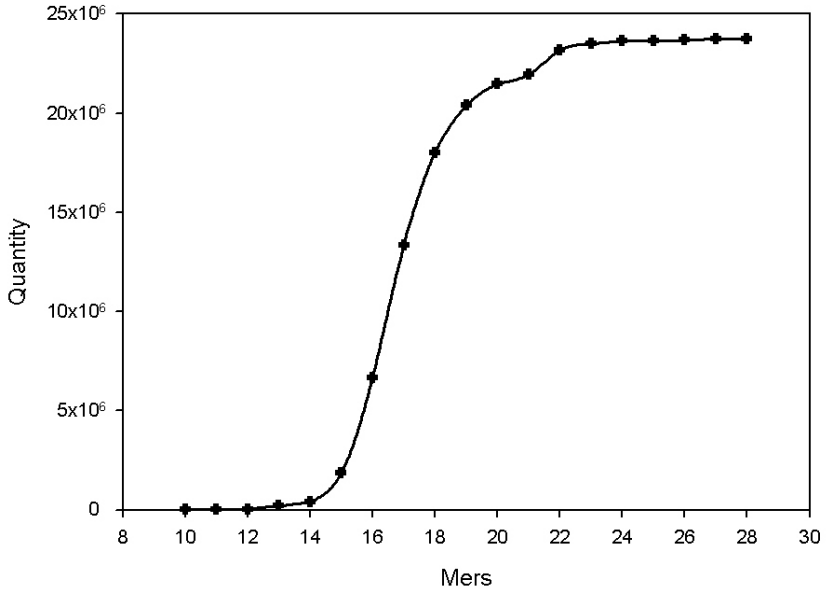

Fig. 2. Amount of unique markers in correspondence to their length

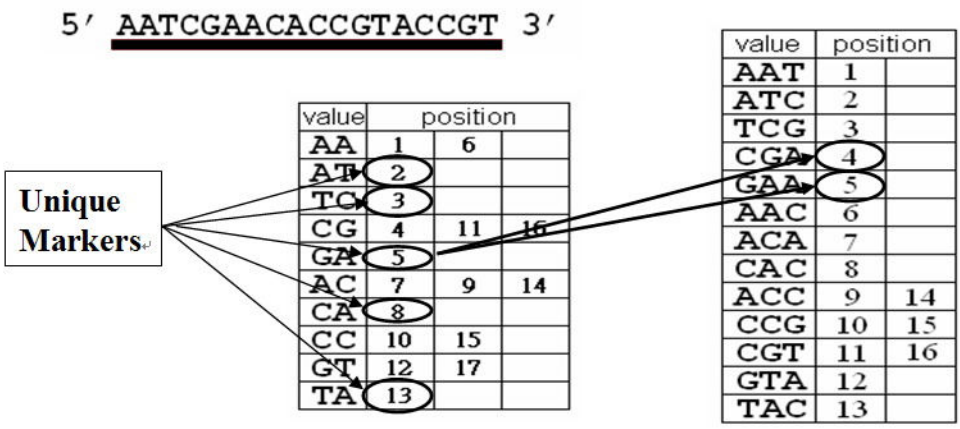

Fig. 3. Any 3-mers marker containing a 2-mers unique marker is also unique

Any sequence containing a unique marker is also a unique marker. For example, in Fig. 3, if the sequence "AATCGAACACCGTACCGT" is given, "GA" is a 2-mers unique marker. Thus the 3-mers "CGA" and "GAA" which contain "GA" are unique as well. Note that each 14-mers unique marker is contained in eight 21-mers unique markers.

2.2 Elongation of unique matching markers

In order to speed up processing, our algorithm uses the idea of elongation. In order to elongate, the unique matching marker have first to be defined as follows:

Let $G[i, j]$ and S[k, l] denote (gi, gi+1, …, gj) and (sk, sk+1, …, sl), respectively. Suppose G[i, i+m] is a m+1-mers unique marker and is identical to S[k, k+m], then G[i, i+m] and S[k, k+m] form a unique matching marker.

In fact, sk-1 may also equal gi-1. Similarly, sk+m+1 may equal gk+m+1. In our algorithm, we find the maximal unique matching marker (MUM) by elongating both sides. The elongation process can save a lot of computation time.

2.3 Clustering of maximal unique matching markers

After finding all MUMs using our algorithm, these MUMs are sorted according to their position on the genome. MUMs may be located on different contigs.

Contig can be seems to the unit of chromosome when we assemble the genomic. The sequence length of sequencing result is usually about to 600 nucleotides and the length of a chromosome usually between millions to the hundred millions nucleotides. Hence, we need assemble the EST sequence to the small part, and take these parts to the full chromosome. These small parts of sequences called the contigs.

MUMs located on different contigs are considered as belonging to different clusters. In some cases, e.g. if transposition or reversal between the genomes occurs, some positions are not in ascending order in that cluster. After positioning MUMs by their positions on the genome, we employ a variation of the "Longest Increasing Subsequence" algorithm (LIS) [8] to find the longest set of MUMs occurring in ascending order both on *G* and *S*. For instance, if the order of positions is given by the sequence {1, 2, 10, 4, 5, 8, 6, 7, 9, 3}, the result of LIS is {1, 2, 4, 5, 6, 7, 9}.

2.4 Multi-Layer Unique Markers

Let U_k denote the set of unique markers of *G* with the length *k*. The larger *k*, the higher the probability of *S* to contain any element in U_k. Yet, the larger *k*, the higher the number of elements in U_k will become. The amount of time to map *S* to *G* grows rapidly. In order to have the benefits of both shorter and longer UMs, we introduced the concept of multi-layer unique markers. For the human genome, a 3-layers unique marker table can be constructed as follows: The three layers consist of U_{14}, U_{21} and U_{28}. Why choose the 14, 21, and 28 as the length is because we use 7-mers to the smallest unit to record a unique marker. Use short length of unique marker to build the table will have the less result, and use too long length of marker will increase the size of search table. In Fig. 2, we can found if the length of unique marker more than 28, the quantity of unique markers will not increase, but

it will increase the search table size. And if choose length of unique marker less than 14-mers, the quantity of unique markers will not decrease continuously. Hence, we choose the 14, 21 and 28 as our pattern length.

Elements in U_{14} are sorted in the first layer. Each entry of the first layer consists of three fields: the unique markers, the position of the unique markers on G and a pointer pointing to the next layer. The second and third layers are constructed in the same way. Note that it is possible to store the entire first layer data in the RAM of a personal computer. In Fig. 4, a conceptual view of the first two layers is shown.

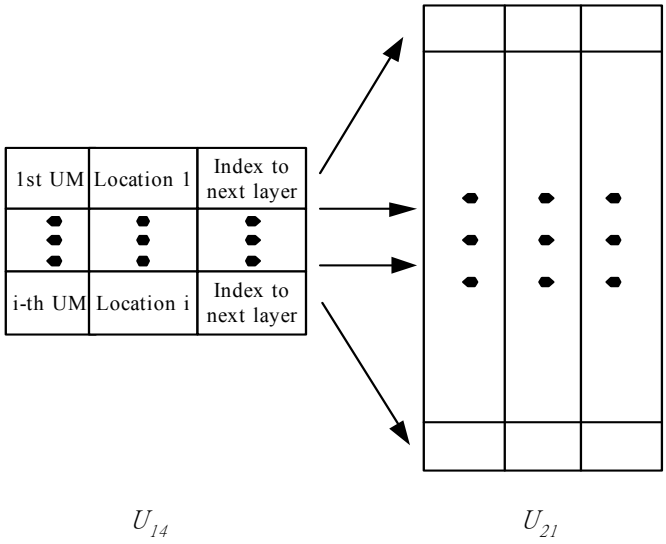

Fig. 4. Conceptual view of the first two layers of multi-layer unique markers

The reason why we did not just use longer unique markers to map SNPs was the huge amount of computation time needed. Besides, due to nucleotide polymorphism and sequencing errors, shorter unique markers sometimes can provide valuable information that longer unique markers can not. Normally, any 28-mers containing a 14-mers unique marker is also a unique marker. We may say that it is sufficient to search for U_{28} only. However, suppose there are some sequencing errors on the 28-mers unique marker region but no sequencing error on the 14-mers UM region (see Fig. 5), then the 28-mers UM cannot be used for mapping S to G. Yet, the 14-mers UM can be used to map S to G. Therefore, in this case, the 14-mers UM provides valuable positioning information while the 28-mers UM does not.

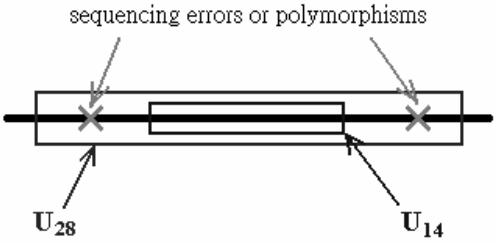

Fig. 5. Possibility of U14 providing valuable information that U28 does not

2.5 Positioning a SNP sequence on a genome

Suppose we have built a multi-layer unique marker table as stated above, if a SNP sequence S, is given, then the position S on the genome G is determined as follows: By scanning S, we find UMs on S and its corresponding address on G (see Fig. 6). Thus, UMs can be used to map S. With our method, we search for all MUMs containing UMs in U_{14}, U_{21} or U_{28}. Starting from S[1,14], we first search on the first layer. If none is found, we try to find S[1,21] in the second layer. If none is found, we try to find S[1,28] in the third layer. If a matched unique marker is found, we elongate it to find a MUM.

By elongation, we can skip the unnecessary query of the unique marker table. The flow chart of finding MUMs starting from S_i is shown in Fig. 7. Scanning S, all MUMs can be found. By applying the LIS algorithm, we find the longest MUM cluster which assign SNP sequence at the same location. In this way, S can be positioned on G more efficiently.

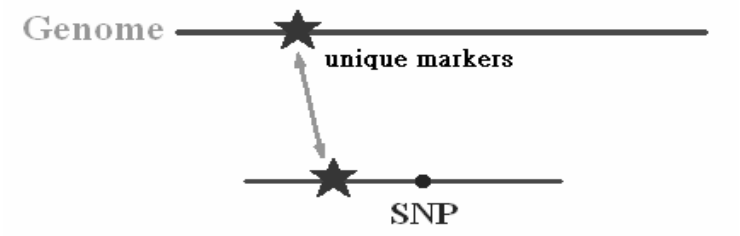

Fig. 6. Positioning a SNP sequence on a genome

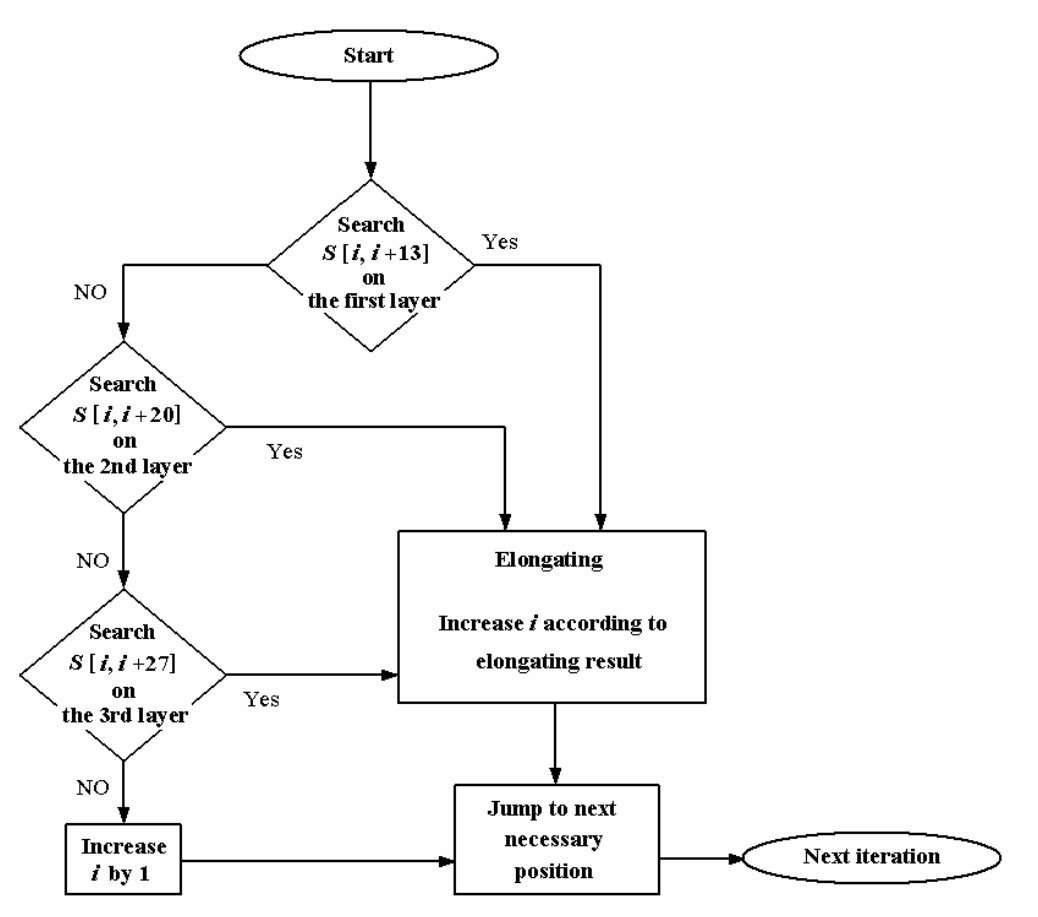

Fig. 7. The flow chart of finding MUMs starting from Si

3 Algorithms

In this section, we will describe our algorithms step by step. Our algorithms have two phases: the preprocessing phase and the query phase. In the preprocessing phase, our algorithm constructs the 3-layer unique marker table of the genomic sequence *G*. In the query phase, our algorithm positions a SNP sequence *S* on *G* by using the 3-layer unique markers.

3.1 Algorithms of building the multi-layer unique marker table

Step 1: Convert every 7-mers of *G* into a two bytes integer

For each 7-mers of G, we convert its DNA sequence into the corresponding two bytes integer. Because 47 < 216, we can use two bytes to represent a 7-mers. Note that if any 7-mers contains any unknown base-pair, we use -1 to represent this 7-mers.

Step 2: Sort all 28-mers of G

By combining four consecutive non-overlapping 7-mers, we get the corresponding value of every 28-mers of G. In order to avoid maintaining huge files and to speed up, we sort all 28-mers of G as follows. By hashing, we partition all 28-mers into 1024 files. Data in the same file have the same first 5-mers. Therefore, only the last 23-mers need to be stored in files. Then we perform sorting in each file.

Step 3: Create multi-layer unique marker table

By keeping records of the data which appear on G exactly once, we get U28 from all the sorted 28-mers of G. By examining the first 21-mers of U28, we can construct U21. We can construct U14 in the same way. Then, we construct a multi-layer unique marker table as follows. U14, U21, and U28 are kept in the first, second and third layer, respectively. Each entry of the first layer consists of three fields: the unique marker, the position of the unique marker on *G* and a pointer pointing to the next layer. The second and third layers are constructed in the same way. Note that it is possible to store the entire first layer data in the RAM of a personal computer.

3.2 Algorithms of mapping SNP on a Genome

Step 1: Convert each 7-mers of S into a two bytes integer

Similar to the first step of building a multi-layer unique marker table, we convert every 7-mers of S into a two bytes integer.

Step 2: Find MUMs

Starting from i = 1, we search for the MUM containing S[i, i + 13] as follows: First, we search for S[i, i + 13] on the first layer of the multi-layer unique marker table. If found, we elongate both sides of S[i, i + 13] to find the MUM. Otherwise, we use the searching result as index to search for S[i, i + 20] on the second layer of the multi-layer unique marker table. If found, we elongate both sides of S[i, i + 20] to find the MUM. If we can't find any, we use the current searching result as index to search for S[i, i + 27] on the third layer of the multi-layer unique marker table. If found, we elongate both sides of S[i, i + 27] to find MUM. Otherwise, if no element in U14, U21 or U28 is found at the current position, we increase i by 1.

If there is a MUM found at the current position, we increase i according to the result of the elongation. The flow chart of finding all MUMs is shown in Fig. 7. Note that the value of S[i, i + 13] can be computed by combining S[i, i + 6] and S[i + 7, i + 13] which are pre-computed in the first step. The value of S[i, i + 20] and S[i, i + 27] are computed in the same way. By this method, all MUMs containing any element in U14, U21, and U28 can be found.

Step 3: Find the largest MUM cluster

Once we have found all MUMs in the previous step, we use the LIS algorithm to find the largest MUM cluster. Then, we compute the total length of the largest MUM cluster and its corresponding address on *G*.

4 Result and Discussion

Using the traditional UM method (U_{14}), on average it takes 28.93 seconds to map 100 SNP sequences. In the UM method, every 14-mers needs to be scanned in *S*. In our method, once one unique marker is found, we elongate its both sides to get a MUM. On average, it takes 23.04 seconds to map 100 SNP sequences by using the U_{14} table with elongation. We also build a 2-layer (U_{14} and U_{21}) unique marker table. Since, it needs less searches on the 2-layer unique marker table, it runs faster than using U_{14} only. On average, it takes 12.2 seconds to map 100 SNP sequences by using the 2-layer unique markers table with elongation. On average, it takes only 8.3 seconds to map 100 SNP sequences, using the 3-layer (U_{14}, U_{21}, and U_{28}) unique marker table with elongation. These results are shown in Fig. 8.

By using a personal computer, it took 207 hours to map the entire dbSNP (Corresponding dbSNP release: Version 121, include 9,015,165 sequences) using the 3-layer unique marker table. The success rates for U_{14}, U_{21}, and U_{28} were 57%, 96.8%, and 99.9%, respectively. The comparison of the rate of time used and success rates for both methods is given in Fig. 9.

In dbSNP, there were 8,682,269 SNPs assigned by NCBI. Those SNPs were also 100% assigned by MUGUP. There were 332,896 SNPs not assigned by NCBI. Our MUGUP method successfully assigned 325,405 these SNPs and only 7,491 SNPs were not assigned by MUGUP. (See Table 3.)

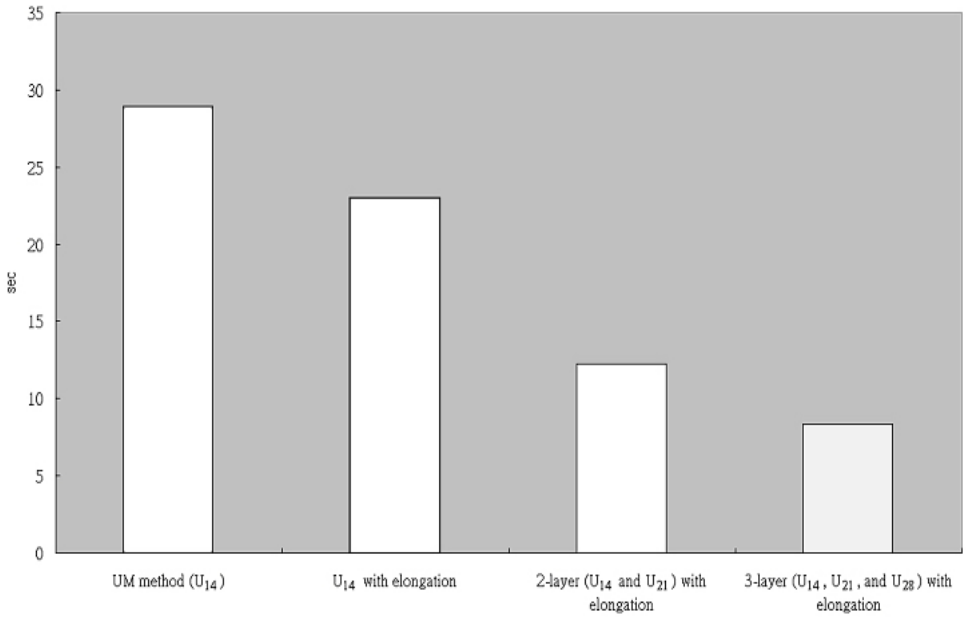

Fig. 8. Time used for positioning 100 SNP sequences with different methods

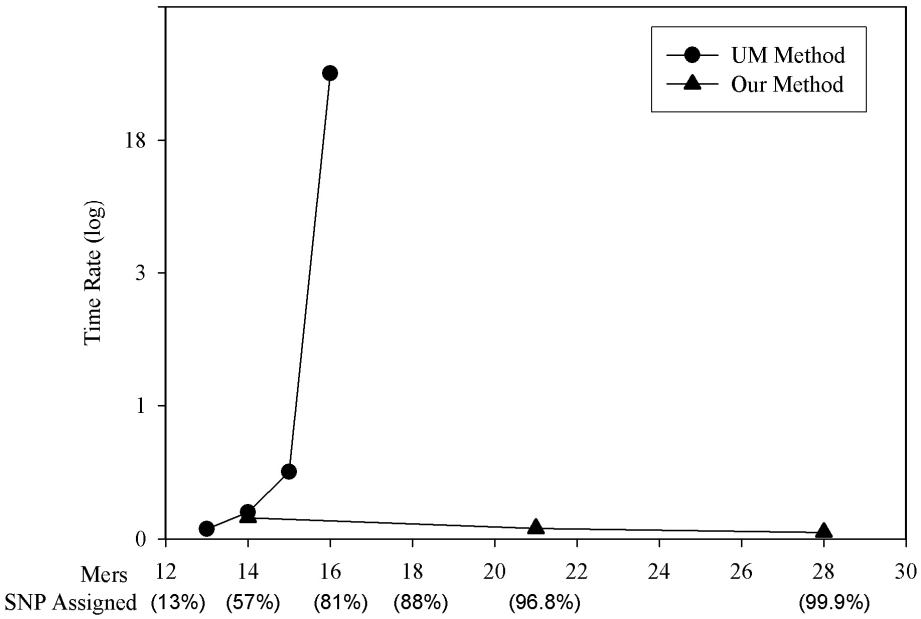

Fig. 9. Comparison of our MUGUP method and the UM method

Table 3. Performance of the MUGUP Method

No. of SNPs in dbSNP	9,015,165	%
Assigned by NCBI	8,682,269	96.31%
Not assigned by NCBI	332,896	3.69%
Assigned by MUGUP	9,007,674	99.92%
Not assigned by MUGP	7491	0.08%

Table 4. Positional Offsets between MUGUP and NCBI assignments

Offset(bp)	SNPs	%
0	8014573	92.31 %
1	331472	3.82%
2	20684	0.24%
3	8184	0.09%
4	7825	0.09%
5	3069	0.04%
6	4131	0.05%
7	2246	0.03%
8	1599	0.02%
9	831	0.01%
≧10	286632	3.30%

For those SNPs successfully assigned by MUGUP and NCBI, the agreement with the NCBI method was very high. In the compassion with the assignments reported in NCBI, the positional offsets between MUGUP and NCBI assignment is shown in Table 4.

In order to compare our method with BLAST and SSAHA, we randomly selected 10 SNP sequence from each human chromosome. Since BLAST and SSAHA need larger RAM to execute, we used Celeron 2.2 GHz with 1G RAM to perform mapping. On average, it took 0.23, 1050 and 2640 seconds to map a single SNP sequence by using MUGUP, SSAHA and BLAST respectively. Therefore, our method is much faster than these two methods.

Our method combines the benefits of both short and long unique markers and shows high efficiency and accuracy. Yet, since there are some homologous regions in the genome, the UM based method fails when a SNP is located at such a region. We have analyzed the distribution of unique markers on the human genome. For U_{14}, we have computed the distance between every two consecutive unique markers. We also performed the same task for U_{28}. The results are shown in Table 5. In the human genome, there are 397,505 (396,501+1,004) intervals larger than 153 bases containing not one 28-mers UM (see Table 3). For the SNP sequences located at these regions, a UM based method with 28-mers UM can not be used to locate them. Thus a new indexing technique is needed to overcome this shortcoming of the UM based method.

Table 5. Distance distribution of two consecutive Ums

Distances	1	2~9	10~153	154~5,913	5,914~409,113	>409,113
No. of 28-mers	2,342,490,011	23,446,619	7,946,503	396,501	1,004	0
No. of 14-mers	16,817,723	11,163,327	7,083,478	5,191,683	3,132	0

Reference

[1] G. Marth, R. Yeh, M. Minton, R. Donaldson, Q. Li, S. Duan, R. Davenport, R.D. Miller, and P. Y., Kwok, "Single-nucleotide polymorphisms in the public domain: How useful are they?", *Nature Geneitics*, Vol.27, pp. 371-372, 2001.

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

[3] Y. Y. Chen, S. H. Lu,, S. C. Shih, and M. J. Hwang, "Single Nucleotide Polymorphism Mapping Using Genome-Wide Unique Sequences," *Genome Research*, Vol.12, pp.1106–1111, 2002.

[4] J. Ogasawara, and S. Morishita, "*Fast and sensitive algorithm for aligning ESTs to Human Genome*," Proc. of IEEE Computational Systems Bioinformatics Conference 2002. pp. 43–53, 2002.

[5] T. J. Chuang, , W. C. Lin, H. C. Lee, C. W. Wang, K. L. Hsiao, Z. Wang, H. D. Shieh, S. C. Lin, and L. Y. Chang, "A Complexity Reduction Algorithm for Analysis and Annotation of Large Genomic Sequences," *Genome Research,* Vol.13, pp.313–322, 2003.

[6] Z. Ning, A. J. Cox, and J. C. Mullikin, "SSAHA: A fast search method for large DNA databases," *Genome Research*, Vol.11, pp.1725–1729, 2001

[7] D. G. Wang,, J. B. Fan, C. J. Siao, A. Berno, P. Young, R. Sapolsky, G. Ghandour, N. Perkins, E. Winchester, J. Spencer, et al., "Large-scale identification, mapping, and genotyping of single-nucleotide polymorphisms in the human genome," *Science*, Vol.280, pp.1077–1082, 1998.

[8] D. Gusfield, *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*, Cambridge University Press, New York, 1997.

[9] L. C. Bailey Jr., S. Fischer, J. Schug, J. Crabtree, M. Gibson, and G. C. Overton, "GAIA: Framework annotation of genomic sequence," *Genome Research,* Vol.8, pp.234–250, 1998.

[10] E. Birney, and R. Durbin, "Dynamite: a flexible code generating language for dynamic programming methods used in sequence comparison," *Proc. Fifth Int. Conf. Intelligent Systems Mol. Biol*, Vol.5, pp.55-64, 1997.

[11] C. B. Burge, and S. Karlin, "Finding the genes in genomic DNA," *Current Opinion in Structural Biology*, Vol.8, pp.346–354, 1998.

[12] K. M. Chao, J. Zhang, J. Ostell, and W. Miller, "A tool for aligning very similar DNA sequences," *Computer Applications in the Biosciences,* Vol.13, pp.75–80, 1997.

[13] K. M. Chao, J. Zhang, J. Ostell, and W. Miller, "A local alignment tool for very long DNA sequences," *Computer Applications in the Biosciences,* Vol.11, pp.147–153, 1995.

[14] A. L. Delcher, S. Kasif, R. D. Fleischmann, J. Peterson, O. White, and S. L. Salzberg, "Alignment of whole genomes," *Nucleic Acids Research*, Vol.27, pp.2369–2376, 1999.

[15] L. Florea, G. Hartzell, Z. Zhang, G. M. Rubin, and W. Miller, "A computer program for aligning a cDNA sequence with a genomic DNA sequence," *Genome Research,* Vol.8, pp.967–974, 1998.

[16] M. S. Gelfand, A. A. Mironov, and P. A. Pevzner, "Spliced alignment: A new approach to gene recognition," *Proceedings of the National Academy of Sciences*, Vol.93, pp.9061–9066, 1996.

[17] W. Kent, "BLAT - The BLAST-Like Alignment Tool," *Genome Research,* Vol.12, pp.656-664, 2002.

[18] E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C., Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, et al., "Initial sequencing and analysis of the human genome," *Nature*, Vol.409, pp.860–921, 2001.

[19] F. Liang, I. Holt, G. Pertea, S. Karamycheva, S. L. Salzberg, and J. Quackenbush, "Gene Index analysis of the human genome estimates approximately 120,000 genes," *Natural Genetics*, Vol.25, pp.239–240, 2000.

[20] A. V. Lukashin, and M. Borodovsky, "GeneMark.hmm: New solutions for gene finding," *Nucleic Acids Research*, Vol.264, pp.1107–1115, 1998.

[21] L. Pachter, S. Batzoglou, V. I. Spitkovsky, E. Banks, E. S. Lander, D. J. Kleitman, and B. Berger, "A dictionary-based approach for gene annotation," *Journal of Computational Biology,* Vol.6, pp.419–430, 1999.

[22] S. Rogic, A. Mackworth, and F. Ouellette, "Evaluation of gene finding programs," *Genome Research*, Vol.11, pp.817-832, 2001.

[23] S. H. Sze, and P. A. Pevzner, "Las Vegas algorithms for gene recognition: Suboptimal and error-tolerant spliced alignment," *Journal of Computational Biology*, Vol.4, pp.297–309, 1997.

[24] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, et al., "The sequence of the human genome," *Science*, Vol.291, pp.1304–1351, 2001.

[25] Z. Zhang, S. Schwartz, L. Wagner, and W. Miller, "A greedy algorithm for aligning DNA sequences," *Journal of Computational Biology,* Vol.7, pp.203–214, 2000.