A Novel Method Providing Exact SNP IDs from Sequences

Hsueh-Wei Chang, Yu-Huei Cheng, Tai-Chen Chen, Cheng-San Yang, Li-Yeh Chuang, and Cheng-Hong Yang, *Member, IAENG*

***Abstract*—Single-nucleotide polymorphisms (SNPs) are the most common type of DNA sequence variation. An SNP is the substitution of a single base in the sequence for one that is different from that present in the majority of the population. SNPs were very important for personalized medicine, especially for association studies. Each SNP has an ID number (rs#) in dbSNP of NCBI, providing the information for SNP genotype and frequency of many populations. However, many previous association studies provide only the SNP nucleotide position or primer sequences, without giving an SNP ID of NCBI. In this study, we built the dbSNP, SNP fasta and SNP flanking marker databases for the rat, mouse and human organisms from the NCBI databases. Boyer-Moore algorithm, dynamic programming method and database technologies were applied and integrated to identify the SNP IDs within input sequences. Therefore, we proposed a novel method to provide efficient, exact and stable output for SNP IDs discovery from a sequence. It also constitutes a novel application to identify SNP IDs from the literatures for systematic association studies.**

***Index Terms*—SNP, SNP flanking marker, Boyer-Moore algorithm, dynamic programming, database.**

I. Introduction

Single nucleotide polymorphisms (SNPs) are the most common polymorphisms among the genomes of many species. The definition of SNP is a variation of the DNA sequence at the frequency larger than 1% allele of a population. Recently, SNPs are widely applied to personalized medicine [1, 2]. Many methodologies are reported or reviewed for genetic association studies [3-5], however, most of the previously reported SNPs are written in nucleotide/amino acid position formats without providing an SNP ID. For example, C1772T and G1790A SNPs in exon 12 of the hypoxia inducible factor-1alpha (HIF) gene are reported to be associated with the renal cell carcinoma phenotype [6], and TNF gene polymorphisms -857, -863, and -1031 in the TNF gene are analyzed in the osteoporosis association study [7]. Without the SNP ID of NCBI, the associated SNPs are hard to be analyzed or organized to systemic databasing.

Recently, SNP-BLAST [9] was developed by coupling the NCBI dbSNP [8] with a BLAST program of NCBI. SNP-BLAST is designed to perform the BLAST function among various SNP databanks for many species. The BLAST program of NCBI uses heuristic algorithms, which are less time-consuming and simple, to search for homologous sequences across species in GenBank. However, it cannot provide exact SNP IDs by inputting sequences. When using the blastn function of SNP-BLAST with or without megablast to perform BLAST for a partial sequence, results do not always show the SNP rs# within the input sequence. Even using megablast with IUPAC format sequences, it often shows "No significant similarity found", such as rs8169551 (rat), rs7288968 (human) and rs2096600 (human) etc. BLAT [10] in UCSC Genome Browser uses the index to find regions in the genome likely to be homologous to the query sequence. In our experiences, BLAT is more accurate and faster than other existing alignment tools. It rapidly scans for relatively short matches (hits), and extends these into high-scoring pairs (HSPs). However, it usually hits so many sequences distributed in different chromosomes and sometimes the result doesn't show the originally entered rs# in selecting the option of the SNPs of the title is "Variation and Repeats", such as rs8167868 (rat), rs2096600 (human), and rs2844864 (human)…etc. Previously, we utilized a Boyer-Moore algorithm [11] to match sequences with the SNP fasta sequence database for the human, mouse and rat genomes. However, the problems of nucleotide change, insertion or deletion in sequences were not addressed in this method. This method cannot provide the SNP IDs. Accordingly, in-del (insertion and deletion) sequences were not acceptable. In order to solve this problem, a dynamic programming method [12] was chosen. However, this method occupies too much memory and is time-consuming when applying to the huge human SNP database; therefore it is impracticable. Finally, we took notice of Uni Marker [13] and generated the following idea. We used SNP flanking markers that are extracted from SNP fasta sequence and then they

Hsueh-Wei Chang is with the Faculty of Biomedical Science and Environmental Biology, Kaohsiung Medical University, Taiwan (e-mail: changhw@kmu.edu.tw).

Yu-Huei Cheng is with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Taiwan (e-mail: yuhuei.cheng@gmail.com).

Tai-Chen Chen is with the Ministry of Education, Taiwan. (e-mail: Janson123@moe.gov.tw)

Cheng-San Yang is with the Department of Plastic Surgery, Chiayi Christian Hospital, Taiwan.

Li-Yeh Chuang is with the Department of Chemical Engineering, I-Shou University, Kaohsiung, Taiwan (email: chuang@isu.edu.tw)

Cheng-Hong Yang is with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Taiwan (e-mail: chyang@cc.kuas.edu.tw).

combined Boyer-Moore algorithm to search markers in the query sequences to identify possible SNPs. Then, we employed a dynamic programming to validate these SNPs to obtain exact SNP IDs. The proposed method greatly reduces matched time and memory space. The experimental results show that our proposed approach is efficient, exact and stable. Thus, it is a valuable approach when identifying SNP IDs from the literature, and could greatly improve the efficiency of systematic association studies.

II. Methods

This integrated approach is proposed for effective, stable and exact. It is based on SNP fasta database, and using Boyer-Moore algorithm and dynamic programming method. The following will illustrate the implementation.

2.1 The application of the Boyer-Moore algorithm

We use a Boyer-Moore algorithm to search for SNP flanking markers in a sequence. The Boyer-Moore algorithm usually matches from right to left, which is in contrast to the usual methods. However, the average search efficiency of the Boyer-Moore algorithm is superior to Knuth-Morris-Pratt algorithms and Brute Force algorithms. These three methods are briefly described and compared below.

(1) Brute Force algorithms- match forms from left to right and one by one for all text. If some error occurs in the matching process, the matching pattern window will shift one position in order to match the next character in the text. It will take the time complexity is O(mn).

(2) Knuth-Morris-Pratt algorithms- match from left to right. In the process, the phase in advance will take O(m) space complexity and time complexity and the phase of search will take O(m+n) time complexity.

(3) Boyer-Moore algorithms- match from right to left. The pretreatment stage take O(m+σ) space and time complexity. σ is the bad-character shift function which is stored in the size of table and the best perform efficiency is O(n/m) time complexity.

Boyer-Moore algorithms use a bad-character shift function and a good-suffix shift function. Fig. 1 describes the process of the Boyer-Moore algorithm's bad-character shift, in which T represents a text, and P represents the pattern to be aligned. As shown in Fig. 1-(1), P is aligned from left to right; P(12)=T(13),P(11)=T(12), but P(10) ≠ T(11), which means the position within P(10) and T(11) mismatched. By using a bad-character shift rule, the mismatch can be shown to occur in P, in our case P(10). Then, searching from the left of P(10) , the same character mismatch is shown for T(11), i.e. P(7)= T(11). At this stage, the bad-character shift rule will move the P window and align P(7) to T(11) as shown in Fig. 1-(2). After that, the alignment from right and left of P(12) and T(16) will start again.

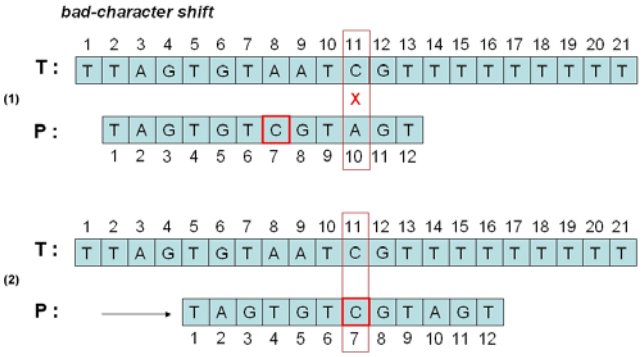


Fig. 1. The bad-character shift process

The good-suffix shift rule is divided into a good-suffix shift1 and a good-suffix shift2. The process for the good-suffix shift1 is described in Fig. 2. In Fig. 2-(1), P is aligned from right to left, P(12)=T(13), P(11)=T(12), but P(10) ≠ T(11). This means that a mismatch is present within P(10) and T(11). Good-suffix shift1 then searches from the right of the P mismatch position, that is from the right of the character of P(10) and finds the match T(12, 13), which is a suffix string of P, P(12, 13). Also, the right character of the P suffix string can not be the same as the mismatch P(11). As shown in Fig. 2-(1), P(8,9) is the suffix string found, but since P(7)=P(10), the search process continues from the left until P(5,6) and P(4)≠P(11) are found. The good-suffix shift1 rule will then move the P window and align P(4) to T(11) as shown in Fig. 2-(2). However, if no suffix string can be found in P, but the prefix string is the suffix substring of the suffix string in P, good-suffix shift2 is implemented. Fig. 3-(1) shows that P(8) mismatches T(9), and P(9, 12) is the suffix string of P. The suffix string P(1, 3) matches the suffix string P(9, 12), i.e. P(1, 3)=P(10, 12)=T(11, 13). Therefore, the good-suffix shift2 rule will move the P window and align P(1) to T(11) as shown in Fig. 3-(2). After that, alignment from right to left of P(12) and T(22) continues.

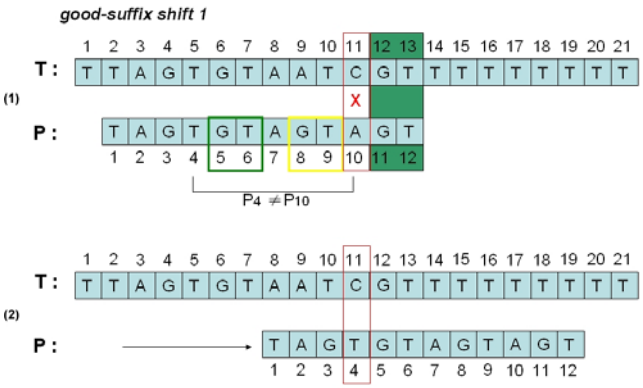


Fig. 2. Good-suffix shift1 process

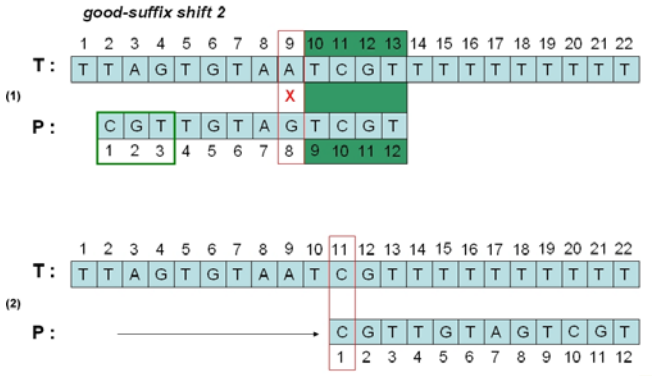


Fig. 3. Good-suffix shift2 process

When using a Boyer-Moore algorithm to select possible SNPs from the SNP fasta sequences database by query sequence, the following three conditions have to be considered:

Condition 1. Sequence only match SNP flanking marker 3', but SNP flanking marker 5' is mismatched. The SNP flanking marker 5' could possibly appear near the left side of the sequences, it resulted in SNP flanking marker 5' could not been matched, as shown in Fig. 4. This condition will be candidate of possible SNPs.

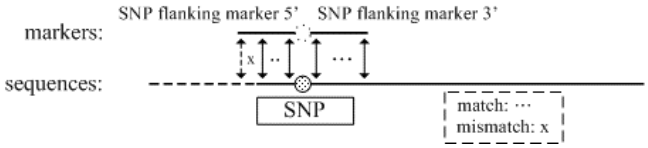


Fig. 4. Sequence only matches to SNP flanking marker 3'.

Condition 2. Sequence only match SNP flanking marker 5', but SNP flanking marker 3' is mismatched. The SNP flanking marker 3' may appear at the right side of the sequences, it resulted in SNP flanking marker 3' could not been matched, as shown in Fig. 5. This condition will be candidate of possible SNPs.

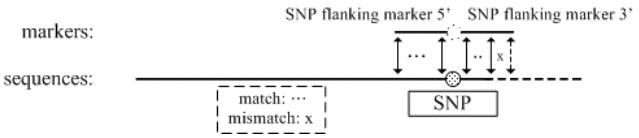


Fig. 5. Sequence only matches to SNP flanking marker 5'.

Condition 3. Sequence matches to SNP flanking marker 5' and SNP flanking marker 3'. In this case, two possibilities exist: (a) a SNP exists within the sequences, as shown in Fig. 6. It will be candidate of possible SNPs. (b) a SNP does not exist within the sequences, but SNP flanking markers exist, as shown in Fig. 7 and Fig. 8. In Fig. 7 and Fig. 8, the SNP flanking marker 5' and the SNP flanking marker 3' are separated from each other, so the existence of a SNP is impossible. We eliminate it from the candidate of possible SNPs.

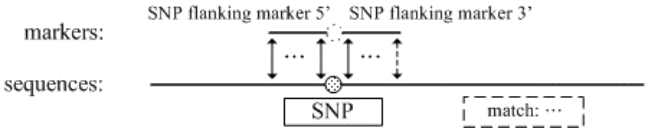


Fig. 6. SNP exists within sequence.

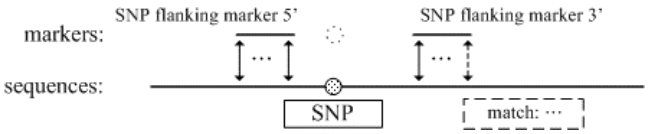


Fig. 7. SNP does not exist within sequence, because the distance of the matched SNP flanking markers.

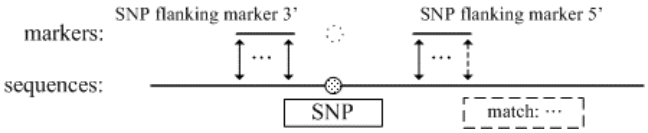
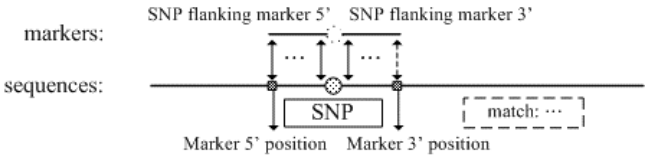


Fig. 8. SNP does not exist within sequence, because the orientation and distance of the matched SNP flanking markers.

Possible SNPs will be selected by a criterion. The discriminable criterion is presented below and illustrated in Fig. 9.

if ((marker 5' position + marker 5' length + 1) == marker 3' position) (1)

If above formula (1) is confirmed, the sequence will possibly contain a SNP that corresponding one of SNP fasta sequences database. The "+1" of this formula (1) represents the base of the SNP.



(Marker 5' position + marker 5' length + 1) = Marker 3' position

Fig. 9. Discriminable criterion for possible SNPs.

2.2 The Revise of SNP flanking marker

Because of the exact character matching of a Boyer-Moore algorithm, we must consider three conditions when applying SNP flanking markers. These three conditions are illustrated below:

Condition 1. SNP flanking marker 5' has one SNP and upward in it, which will result in mismatch using Boyer-Moore algorithm. And the SNP flanking marker 3' is at the right side of the sequence and mismatched. It is illustrated in Fig. 10. This condition is not any SNPs found.

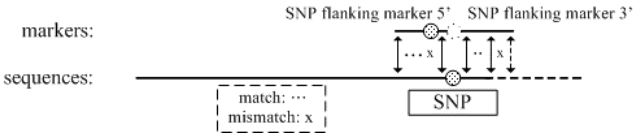


Fig. 10. SNP flanking marker 5' contains SNPs in it and SNP flanking marker 3' is not matched to the sequence, it is not any SNPs found.

Condition 2. SNP flanking marker 3' has one SNP and upward in it, which will result in a mismatch using Boyer-Moore algorithm. And the SNP flanking marker 5' is at the left side of the sequence and mismatched. It is illustrated in Fig. 11. Again, no SNPs is found in this condition.

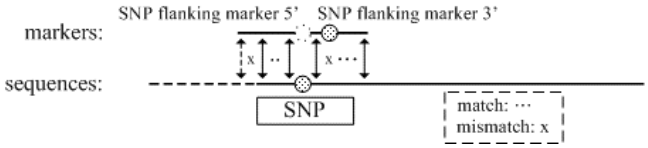


Fig. 11. SNP flanking marker 3' contains SNPs in it and SNP flanking marker 5' is not matched to the sequence, it is also no SNPs found.

Condition 3. Both SNP flanking marker 5'and SNP flanking marker 3' contain SNPs within them. This will result in no markers to match using Boyer-Moore algorithm, but actually SNP markers exist in sequence as shown in Fig. 12. It still no SNP is found.

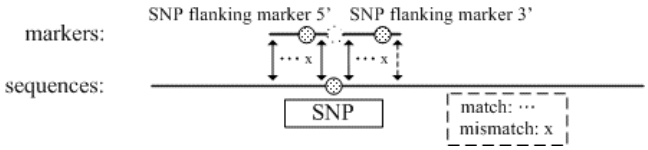


Fig. 12. Both SNP flanking marker 5' and SNP flanking marker 3' contain SNPs within them, but no SNP is found.

In order to improve the above faults, we constructed a revised SNP flanking marker table. It uses the SNP chromosome position from dbSNP to find existing SNPs within the SNP flanking marker 5'and SNP flanking marker 3'. For example, under Condition3 shown in Fig. 12, the flanking marker 5' of SNP2 contains SNP1 and flanking marker 3' of SNP2 contains SNP3, respectively. A search process for the flanking markers of SNP2 using the Boyer-Moore algorithm will result in a failure. Therefore, we through the revised SNP flanking marker table to correct the condition. As shown in Table 1, the flanking marker 5' of SNP2 contains SNP1 and the flanking marker 3' of SNP2 contains SNP3. In this case, the SNP will be considered a possible SNP.

Table 1. Example of the revised SNP flanking marker table

SNPs	SNP flanking marker 5'	SNP flanking marker 3'
SNP1	none	SNP2
SNP2	SNP1	SNP3
SNP3	SNP2	none

2.4 Alignment using Dynamic programming

Through the steps described above, possible SNPs within query sequence can be retrieved. However, the query sequence must match with the fasta sequence, only SNP flanking markers matched can not prove the existence of a SNP in sequences. If nucleotide bases outside the SNP flanking marker can not be matched to the SNP fasta sequences, the above effort is futile. The SNP flanking marker is too short to make a complete estimate. Consequently, we employ a dynamic programming method to match with fasta sequences of the possible SNPs in order to discover valid SNPs. The dynamic programming method contains an error tolerant function which resolves problems associated with changes, insertions or deletions in sequences. The corresponding SNP fasta sequences will provide the SNP ID. It works as follows. First, the SNP fasta sequences and the input sequences of the suffix edit distance E(i, j) is calculated. Suppose Tj is the SNP fasta sequences, j = 1, 2, …, n, where n is the SNP fasta sequences' length. P_i is a user's input sequences, i = 1, 2, …, m, and m is the user's input sequences length. The procedure for the suffix edit distance is given below.

```
// initialization
1:  for i←0 to m do
2:      E(i, 0)←i
3:  next i
4:  for j←0 to n do
5:      E(0, j)←0
6:  next j

// suffix edit distance E(i, j)
7:  for i←1 to m do
8:      for j←0 to n do
9:          if (T(j) = P(i)) then
10:             E(i, j)←(i-1, j-1)
11:         else
12:             min←MIN[E(i-1, j), E(i, j-1)]
13:             E(i, j)←min + 1
14:         end if
15:     next j
16: next i
17: return E(i, j)
```

In order to obtain partially homologous sequences, the maximum tolerance error rate for the input sequences is accepted. Once the error count is equal to or smaller than the maximum tolerance error rate, the input sequences is aligned successfully to the SNP fasta sequences.

Maximum tolerant error number = (input sequence length)* (tolerant error rate) (2)

The homologous sequences can be found by using previously obtained suffix edit distances E(i, j) and the maximum tolerance error number based on backward dynamic programming. Once the suffix edit distance E(i, j) is smaller than or equal to the maximum tolerance error number, it is processed. The backward sequences are the homologous sequences that fit with the analogue. For example, if input sequences contain the bases (nucleotides) TAGC, the maximum tolerance error rate is 20%. When the input sequences are aligned with SNP fasta sequences of 10 bps, e.g.

TGGATACCAT, the maximum tolerance error number is 10 * 0.2 = 2. In other words, only two or fewer error alignments are allowed in this case (Fig. 13). The boldface arrows in Fig. 13 indicate the output of an agreeable homologous alignment; the homologous sequences are (1)TG (2)TGG (3)TGGA and (4)TA.

		T	G	G	A	T	A	C	C	A	T
	0	0	0	0	0	0	0	0	0	0	0
T	1	0	1	1	1	0	1	1	1	1	0
A	2	1	2	2	1	1	0	1	2	1	1
G	3	2	1	2	2	2	1	2	3	2	2
G	4	3	**2**	**1**	**2**	3	**2**	3	4	3	3
			(1)	**(2)**	**(3)**		**(4)**				

Fig. 13. Homologous alignment and possible homologous sequences

III. Results and Discussion

This research utilizes the NCBI SNP [14] rs_fasta sequences database, which contains the Human (ftp://ftp.ncbi.nih.gov/snp/organisms/human_9606/), Mouse (ftp://ftp.ncbi.nih.gov/snp/organisms/mouse_10090/), and Rat (ftp://ftp.ncbi.nih.gov/snp/organisms/rat_10116/) genomes. To implement the proposed method, a SNP flanking marker database must be built with data from the SNP fasta sequences database. In order to ensure that exact SNP IDs can be found, the selection of the length of the SNP flanking marker is important. When using shorter SNP flanking markers, possible SNPs are more rapidly identified by using Boyer-Moore algorithm, but many of the select SNPs are insignificant. These insignificant SNPs will increase the load for the following process of determining exact SNP IDs. Longer SNP flanking marker will fail to obtain SNP IDs using the Boyer-Moore algorithm, because sequence may contain changes, i.e. an insertion or a deletion, or long markers may contain SNPs with high frequency. Therefore, this research adopted a length of 10 bps of SNP flanking sequences of the fasta database as a standard for the SNP flanking marker length. Although, the marker length influences the matching results, it is compensated by the revised SNP flanking marker table that we will introduce following. Chromosome position of the table SNPContigLoc in dbSNP [8] b126 was employed to find SNPs within the SNP flanking marker, and then build the revised SNP flanking marker table.

The proposed approach using Microsoft Windows XP, a 3.4G MHZ processor, 1GB of RAM memory, and JRE (Java Runtime Environment) with a maximum JAVA heap size of 800MB to discover SNP rs28909981 [Homo sapiens]. We mainly aimed at the following three sequences:

Sequence 1.
AAGAGAAAGTTTCAAGATCTTCTGT**S**TGAGGAAAAT
GAATCCACAGCTCTA

Sequence 2.
AAGAGAAAGTTTCAAGATCTTCTGT**C**TGAGGAAAAT
GAATCCACAGCTCTA

Sequence 3.
AAGAGAAAGTTTCAAGATCTTCTGT**G**TGAGGAAAAT
GAATCCACAGCTCTA

(1) For test sequence 1, we set the dynamic programming method with error tolerant bases = 0. rs28909981 was successfully identified and had 27 SNP flanking marker matches. Run time was 2844 millisecond.

(2) For test sequence 2, we set the dynamic programming method with an error tolerant bases = 1, because the C allele was mismatched with the SNP in fasta sequence. rs28909981 and rs17883172 were identified and had 36 SNP flanking marker matches. Run time was 3313 millisecond. rs17883172 is similar to rs28909981. The rs17883172 sequence was as follows:
GAGAAAGTTTCAAGATCTTCTGT**C**T**R**AGGAAAATGA
ATCCACAGCTCTACC
The **C** allele represents SNP rs28909981. We still search rs28909981 successfully and discovered SNP rs17883172 in this sequence.

(3) For test sequence 3, we set the dynamic programming method with error tolerant bass = 1, because the G allele is mismatched with the SNP in fasta sequence. The result finds rs28909981 successfully and had 34 SNP flanking marker matches. Run time was 3141 millisecond.

(4) For test sequence 1, we adjusted the dynamic programming method with error tolerant bases = 5. rs28909981 and rs17883172 could be found, and 27 SNP flanking marker matches were identified. Run time was 2750 millisecond. We also discovered that test sequence 2 and sequence 3 with error tolerant bases = 5 still find rs28909981 and rs17883172.

The results described above show that the presented approach indeed provides exact SNP IDs from sequences. The advantages of this approach are effective, stable and exact. It seeks through SNP fasta database and only aims at specific database. By the property, it reduces the unknown errors and performs the more exact output. The proposed approach can be used for specialized application of SNP IDs identification. It will help biologists to find SNP IDs within input sequences and have the chance to find invalidated SNPs. It also is useful in SNP association studies.

IV. Conclusion

SNPs are essential for personalized medicine. In order to identify SNP ID within input sequences, this research proposes the use of SNP flanking markers and combines Boyer-Moore

algorithm with dynamic programming to provide exact SNP IDs from sequences. The NCBI dbSNP, SNP fasta and SNP flanking sequences of 10 bps for the rat, mouse, and human organisms were mainly built, improving on our previously proposed methods. After implementation, verified SNP IDs could be obtained from sequences in a fast and efficient way. This integrated approach constitutes a novel application to identify SNP IDs, and can be used for systematic association studies.

REFERENCES

[1] Erichsen HC, Chanock SJ: SNPs in cancer research and treatment. Br J Cancer 2004, 90(4):747-751.

[2] Suh Y, Vijg J: SNP discovery in associating genetic variation with human disease phenotypes. Mutat Res 2005, 573(1-2):41-53.

[3] Lunn DJ, Whittaker JC, Best N: A Bayesian toolkit for genetic association studies. Genet Epidemiol 2006, 30(3):231-247.

[4] Newton-Cheh C, Hirschhorn JN: Genetic association studies of complex traits: design and analysis issues. Mutat Res 2005, 573(1-2):54-69.

[5] Su SC, Kuo CC, Chen T: Inference of missing SNPs and information quantity measurements for haplotype blocks. Bioinformatics 2005, 21(9):2001-2007.

[6] Ollerenshaw M, Page T, Hammonds J, Demaine A: Polymorphisms in the hypoxia inducible factor-1alpha gene (HIF1A) are associated with the renal cell carcinoma phenotype. Cancer Genet Cytogenet 2004, 153(2):122-126.

[7] Furuta I, Kobayashi N, Fujino T, Kobamatsu Y, Shirogane T, Yaegashi M, Sakuragi N, Cho K, Yamada H, Okuyama K et al: Bone mineral density of the lumbar spine is associated with TNF gene polymorphisms in early postmenopausal Japanese women. Calcif Tissue Int 2004, 74(6):509-515.

[8] Sherry ST, Ward MH, Kholodov M, Baker J, Phan L, Smigielski EM, Sirotkin K: dbSNP: the NCBI database of genetic variation. Nucleic Acids Res 2001, 29(1):308-311. [http://www.ncbi.nlm.nih.gov/SNP/]

[9] SNP BLAST. [http://www.ncbi.nlm.nih.gov/SNP/snp_blastByOrg.cgi]

[10] Kent WJ: BLAT—The BLAST-Like Alignment Tool. Genome Res. 2002 12: 656-664.

[11] Christian Charras et Thierry Lecroq, Handbook of Exact String Matching Algorithms, King's College London Publications, 2004.

[12] Eddy SR: What is dynamic programming? Nat Biotechnol 2004, 22(7):909-910.

[13] Chen LYY, Lu SH, Shih ESC and Hwang MJ: Single Nucleotide Polymorphism Mapping Using Genome-Wide Unique Sequences. Genome Res. 2002, 12: 1106-1111.

[14] Altschul SF, Gish W, Miller W, Myers EW & Lipman DJ: Basic local alignment search tool. J. Mol. Biol. 1990, 215:403-410.