A Very Fast Algorithm for Detecting Partially Plagiarized Documents Using FM-Index

Chang SeokOck, JongKyuSeo, Sung-Hwan Kim, and Hwan-Gue Cho

***Abstract*—Sequence alignment and fingerprinting are two of the most common methods for plagiarism detection because of their powerful performances. The disadvantage of using these methods is that if the size of the target document is increase, the string processing cost also increases. We use disk-based techniques and Genome assembly used in Next Generation Sequencing (NGS) to overcome this disadvantage. By combining the two methods, we propose a method for very-fast plagiarism detection in a large Korean corpus. The method is based on the Burrows-Wheeler Transform (BWT) and the FM-index for BWT search. For efficient detection, we extract initial consonants from the Korean corpus and build data structures for indexing the extracted initial consonants. We then split the suspected plagiarism query document into several pieces and perform the query search. Finally, we analyze the results of the search to detect the plagiarized sections. Our proposed method shows a maximum of 0.96 precision and 1.0 recall. In the future, we plan to investigate various ways of improving the search algorithm through optimization, and user-specific visualization methods.**

***Index Terms*—Burrows-wheeler transform, FM-index, plagiarism detection.**

I. Introduction

In recent years, technological research and development has attracted considerable attention owing to the importance in various fields. Researchers in various fields submit papers to prestigious journals in order to publish their findings. A company or an institute is evaluated on the basis of these published papers. This has led to fierce internal competition within the same field. Outstanding achievements or original research articles are necessary to remain competitive in such an environment. Consequently, some people resort to plagiarizing, thereby taking the credit and gaining recognition for others'. However, there are two sides to a case of suspected plagiarism: it could be a case of actual plagiarism, or an attempt to expand the scope of an existing study.

It is often difficult to distinguish malicious plagiarism from genuine research aimed at expanding the boundaries of a subject. Considerable time and effort are required to manually examine a large number of papers from various fields for plagiarism. Therefore, we propose a method to detect plagiarized sections of a document in the large amounts of Korean corpus. Our method searches for the sources of the fragments of query document in the corpus using a DNA short-read alignment method employed in Next Generation Sequencing (NGS) [1], [2]. Burrows-Wheeler Transform (BWT), which is a block-sorting algorithm [3], and FM-index data structures [4] are used to index the corpus. A disk-based BWT [5] is used to process large amounts of data, because it is difficult to apply a common BWT algorithm to high-volume corpus processing. In addition, partially plagiarized sections as well as fully plagiarized works can be detected by performing a search about query document fragments.

Our method can be used to detect plagiarized documents and sections. However, we believe that our methods should be adopted by only those having the appropriate authority to determine plagiarism. This is because an accusation of plagiarism may adversely affect the reputation of a researcher. In other words, the ultimate goal of our method is to minimize the cost of detecting plagiarism.

II. Preliminary Works

Our method to detect plagiarized sections of papers in the massive Korean corpus is based on string-processing algorithms and Korean language-processing algorithms. String-processing algorithms are used to detect plagiarized sections. Most plagiarism detection techniques utilize string-processing algorithms. In order to measure the similarity between two documents, researchers use fingerprinting [6] or complicated string-matching algorithms. Regular expressions or well-known string search algorithms such as KMP and Boyer-Moore's algorithm are used to search for identical sentences. These methods are used to find the occurrence of the identical string within the document. On the other hand, there are two well-known methods to find similar document pairs: fingerprinting, which can identify statistically similar document pairs, and Sequence alignment, which allows edit-errors of the similar sentences [7]. These methods operate efficiently for short articles, but are inefficient in terms of time and space complexity when it comes to large documents.

To overcome this drawback, we adopt BWT, a block sorting algorithm with many space-efficient features. The BWT result string has information on original document and suffix array [8]. However, a general BWT algorithm has high space complexity in processing time, so it is difficult to process large documents. Because the available memory of the system has limited capacity, we used the disk-based BWT method [5]. This technique creates a suffix array within a specified memory and stores the processed result to the disk. The information required for a new task is retrieved from the disk memory that contains previous results. The previous results are then merged with current information. This process is significantly faster than the general BWT, because

Manuscript received November 10, 2012; revised January 28, 2013.

The authors are with the Dept. of Computer Engineering, Pusan National University, Busan, South Korea (e-mail: csock@pusan.ac.kr, maniasjk@pusan.ac.kr, sunghwan@pusan.ac.kr, hgcho@pusan.ac.kr).

DOI: 10.7763/IJCCE.2013.V2.194

it utilizes the fast sequential scan property of the disk. The results generated using the method can be used to rebuild the suffix array for the search. However, as the rebuilding complexity of the suffix array is high, we propose using the FM-index data structures that can search directly inside the BWT results [4].In other words, FM-index can be created using BWT results directly without regenerating the suffix array. Thus, FM-index can be used to search large documents at high speeds and detect plagiarism by analyzing the results of the search.

We use structural characteristics of the language in order to process the Korean language. A Korean letter consists of initial consonant, medial (of Korean orthographic syllable), and final consonant. The consonant has more information than the vowel, and some letters do not have a final consonant. Therefore, it is reasonable to extract the initial consonant to compress the string. The loss of information of an original document can be minimized and the restoration of the original document from the extracted string can be prevented to protect the author's copyright.

In addition to research on the basic elements mentioned earlier, we have also incorporated researches associated with actual plagiarism detection. C. Lyon et al. proposed a method to find similar short passages in a large document [9]. The study by M. Joy *et al.* shows the pattern of plagiarism in programming assignments [10]. G. Whale proposed a method that can measure the similarity of programs [11]. Lyon uses fingerprinting and statistical pattern recognition to measure the similarity of short passages. Joy shows how actual students plagiarize their programming assignments and how one can detect these plagiarized assignments. Warwick approach, which is an incremental comparison method, is a core of the method. Whale measured the similarity of the attribute counts and structures from source code analysis.

Methods described above are summarized in Table I.

TABLE I: DIVERSE METHODS FOR PLAGIARISM DETECTION

Field	Method	Description
String Processing	A. Amir [6]	Via Parikh mapping
	T. Smith [7]	Local alignment
	M. Burrows [3]	Block-sorting
	U. Manber [8]	Suffix array
	P. Ferragina [4]	FM-index
	P. Ferragina [5]	Disk-based BWT
Plagiarism Detection	C. Lyon [8]	Fingerprinting
	M. Joy [5]	Warwick approach
	G. Whale [11]	Attribute counts
	M. J. Wise [12]	Improve Whale's method

III. SKIN EXTRACTION OF KOREAN TEXT

Structurally, a Korean character, as shown in Fig. 1, consists of initial consonant, medial (vowel), and final consonant. Fig. 1shows that a Korean character can have a total of six structures, and all the structures have initial consonant and medial obligatorily.

There are nineteen initial consonant characters, twenty-one medial characters, and twenty-eight final consonant characters including ellipsis. There are two individual drawbacks of the medial and the final consonant: The medial has less information of the original word than the initial consonant, and the final consonant may be omitted. On the other hand, all Korean letters must have an initial consonant and it has more information of the word than the others do. However, it is difficult to restore from a document that consists only of initial consonants, because there are many possible combinations. If we use the document, we can reduce the information loss of the original documents and cannot infringe their copyright.

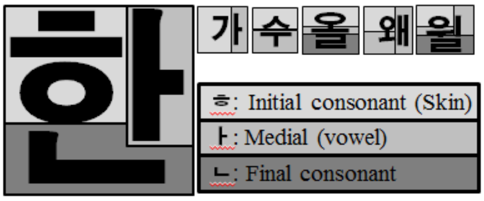


Fig. 1. A structure of a Korean character. There are total of six structures.

Having a small number of alphabets in the string processing is advantageous. However, if the number of the alphabets is too small, the amount of information also decreases, and this can affect the accuracy of the search results. Thus, we use not only the initial consonant but also both numbers and English letters. In case of Chinese letters, we translate them into Korean letters and extract their initial consonants. Then the number of the alphabets is about fifty. The size of the alphabets is neither too small nor too large. We call the initial consonant a skin, which means a representative expression of an original letter and call the process extracting the initial consonant from the corpus as skin extraction.

The extraction algorithm consists of two phases. The first phase is a transformation. A Korean letter is converted into a skin, and all the English letters are converted to lowercase. In addition, white space such as space and return character, and numbers are retained in order to conserve the information. The second step is the mapping phase. The extracted skins are mapped to a number, which can be expressed as a byte having value between0 to 255. Through this process, a Korean letter can be represented using just one byte. The mapping phase increases the efficiency of the post-processing by reducing the size of the documents.

IV. PLAGIARIZED SECTION DETECTION

A large Korean corpus is a collection of a large number of documents, and can be viewed as a document database. After performing the skin extraction for this document database, we detect the plagiarized sections by querying the suspected document. The query document also should be extracted to skin document and fragmented into small query pieces for detecting plagiarized sections.

The skin extraction is needed to search query strings, and the fragmentation is needed to detect partially plagiarized sections. By searching the sources of the query fragments in the database, and analyzing a distribution of the sources, we can find similar documents and plagiarized sections. However, because all the processes produce results with sources expressed as numbers, it is difficult to determine whether the actual plagiarism is only from this particular number represented information. After the plagiarized sections are detected by the analysis of the distribution, and the information of the source has been retrieved, the results have to be visualized to determine whether it is an actual plagiarism.

A. Query Document Fragmentation

A query document to examine plagiarism is usually a single document of small size. We cannot find the partially plagiarized sections using the entire original query document. Therefore, we split the query document into small fragments to detect the sections.

There are two methods to split a query document. The first, *k*-mer (*n*-gram) analysis is used to retrieve strings. It splits the target document into sets of *k* characters, starting from the initial position. This method has a drawback that it takes a lot of time to search the pieces because of the large number of duplicate characters. Even so, a detailed and accurate search is possible. The second method, Genome assembly makes several duplicates, and then splits the copies randomly. This method takes less time than the *k*-mer analysis, because it generates a smaller number of query pieces. However, it has slightly less accuracy than the accuracy of *k*-mer analysis. Fig. 2 shows an example of the skin extraction and the query document fragmentation.

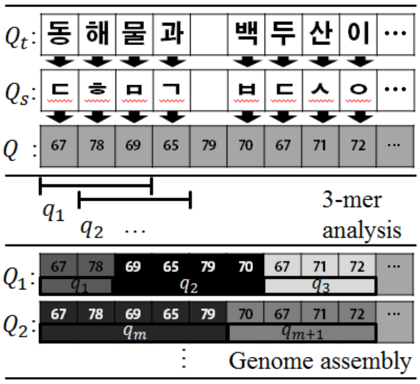
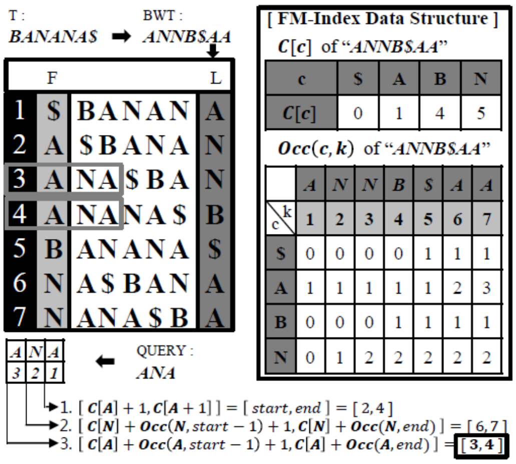


Fig. 2. An example of query document fragmentation.

In the Fig. 2, Q_t=*동해물과백두산이* is a query string,Q_s, is the skin and *Q* is the string mapped as1 byte. If we set *k*=3, a result of *k*-mer analysis is a set of query fragments, $\{q_1,\ q_2, \ldots\}$(as depicted in the middle of Fig. 2). If the number of letters in a document is *n*, the number of fragments is (*n*-*k*+1). On the other hand, if we split *Q* with the Genome assembly, the result is $\{q_1,\ q_2,\ q_3, \ldots, q_m, \ldots\}$(at the bottom in Fig. 2). The number of the results of Genome assembly depends on the cut length and the number of duplicates. If the cut length and the number of duplicates are moderately small, the number of query pieces is generally smaller than the number of pieces of *k*-mer analysis.

B. Similar Section Detection

We should find the sources of query fragments that appear in the document database to detect plagiarized sections. We adopt the disk-based BWT and FM-index to process a large document database. If disk-based BWT cannot process entire job on the limited memory in the processing time, it generates temporary results and stores them on the disk. Then the temporary results are merged into a final result. For this reason, it is possible to implement it on a system with a limited memory such as a PC. In addition, we can get advantages that the information of a document database and aligned characters by the features of BWT. As mentioned earlier, we use the FM-index to search using the BWT string without restoring the suffix array. The FM-index consists of the number of cumulative occurrences of the alphabet at a specific source, and the total number of times each letter of the alphabet occurs in the BWT string. Fig. 3 shows an example of search using FM-index.



c	$	A	B	N
C[c]	0	1	4	5

	A	N	N	B	$	A	A
c \ k	1	2	3	4	5	6	7
$	0	0	0	0	1	1	1
A	1	1	1	1	1	2	3
B	0	0	0	1	1	1	1
N	0	1	2	2	2	2	2

1. $[\, C[A]+1, C[A+1]\,] = [\, start, end\,] = [\,2,4\,]$
2. $[\, C[N]+Occ(N, start-1)+1, C[N]+Occ(N, end)\,] = [\,6,7\,]$
3. $[\, C[A]+Occ(A, start-1)+1, C[A]+Occ(A, end)\,] = \boxed{[\,3,4\,]}$

Fig. 3. An example of query search using BWT and FM-index for text T=*BANANA$*.The *$* is 'EOF' character.

Burrows-Wheeler transform is done on a sample text *T=BANANA$* by sorting all rotations of the text in lexicographic order, and then taking the last column, *L* [1, n] = *ANNB$AA*. FM-index data structures, *C*[*c*] and Occ (*c*, *k*), also should be built. For each character *c* in the alphabet, The C[*c*] table contains the number of occurrences of lexically smaller characters in the text. The function Occ (*c*, *k*) computes the number of occurrences of character c in column [1, *k*]. After that, we search the query string Query=*ANA* using the data structures. The search process starts with '*A*', which is the last letter of the query string. The result of the search is the interval [3, 4]. In addition, we can see the query string, "*ANA*"atF [3, 4]. Finally, to find the occurrence sources of values of the interval [3, 4] in the original text T, FM-index perform a backtracking until *L*[*i*]='*$*'.

$$\begin{cases} 1.\, L[3] = N, C[N] + Occ[N,3] = 7, \\ 2.\, L[7] = A, C[A] + Occ[A,7] = 4, \\ 3.\, L[4] = B, C[B] + Occ[B,4] = 5, \\ \qquad 4.\, L[5] = \$ \end{cases}$$

In the above example, to reach the EOF letter*$*, four iterations such as *L*[3], *L*[7], *L*[4], and *L*[5], are required. The count 'four' is an occurrence source of query, which is "*ANA*" in the text.

The sources collected after search using FM-index are used to analyze similar sections. We analyze the sections by computing the density of the sources and consider the sections to be plagiarized. In order to detect more precisely, we calculate the density from the information of a position, and filter the sections that have more than a certain density threshold. Here we use incremental density method to compute the density. It calculates the density of *i*th position, d'_i, by using (1) and (2). Note that the *i*th position is P_i, and the duplicate of *i*th position is F_i. $d_i$is calculated with both the *i*th position and*i+1*th position progressively. (1≤*i*≤*n*, *n* is the number of the search results)

$$d_i = \frac{\sum of\ frequencies}{Gap} = \frac{F_i + F_{i+1}}{P_{i+1} - P_i + 1} \quad (1)$$

d_i is computed by using (1). We can then getd'_i, by normalizingd_i. (M is the number of maximum duplicates in a location, T is a size of the original skin document, and α is constant weight)

$$d'_i = \frac{\alpha}{M}\left(\frac{F_i + F_{i+1}}{P_{i+1} - P_i + 1} - \left(\frac{F_i + F_{i+1}}{T}\right)\right) \quad (2)$$

d'_i is calculated incrementally from i=s=1 by using (2).We then apply the threshold value to the density. If the density, d'_{s+k} at i=e=(s+k) is lower than the threshold value, the interval [s, e] is considered to be a similar section.

V. Performance Evaluation

We evaluated the performances of the proposed method in detecting plagiarized sections. First, we evaluated the skin extraction method. The advantages of this method are compression and protection of the original document. Next, we assessed the time taken to build a document database and to search query fragments. Finally, we evaluated the precision and recall of the detected plagiarized sections.

A. Document Compression

Compressing a large corpus is an essential part of facilitating the post-processing. In this experiment, we determined the effectiveness of compression of the skin extraction method. It means that we do not use an actual compression algorithm to the compression because doing that requires a decompression before the original data can be used.

A Korean character is represented by 2 bytes in ASCII. In addition, the alphabet size to express the initial consonant in Korean is small. Thus, we get the advantages of compression using the properties of the Korean letter. The compression efficiency for each input file is shown in Table II.

TABLE II: Experimental Result of Data Compression

Input	Size (KB)	Skin Size (KB)	Compressibility (%)
D1	1,011	567	43.92
D2	2,221	1,247	43.85
D3	7,008	3,875	44.71
D4	63,767	35,337	44.58
D5	101,752	56,246	44.72
D6	375,154	207,983	44.56
D7	1,145,097	596,099	47.94
		Average	44.89

TABLE III: Experimental Results of Restoration from Skin

k	1	2	3	4	5	6	7	8	9	10	15
P	0.00	0.00	0.00	0.00	0.02	0.10	0.33	0.65	0.88	0.96	0.99

From Table II, we can see that the method shows an average compression efficiency of 44.89%. Next, we evaluated the protection effectiveness. To do this, we performed k-mer analysis from k=1 to k=15 on a large corpus approximately 1GB. We then analyzed the number of different consonants in each k. The experiment showed that there are many possible combinations to restore the original document. The experimental result of the restoration is shown in Table III(where Pstands for precision).

For k=10, the precision is more than 0.9. However, when k<10, we can see that it is difficult to restore an original string from a skin string whose length is k.

B. Query Search

The search process consists of fragmenting the query document and finding the source of the query fragments. There are two methods to make the fragments, k-mer and Genome assembly. We also used the FM-index data structures to improve the efficiency of the searches.

The experimental data, a large corpus, comprising various documents and documents for a query search, is about 1GB. The documents included in the corpus to determine whether they are plagiarized were used as query documents. We also created a plagiarism scale from 0 to 9 for the query documents depending on the degree of deformation per query document. Distance 0 implies an original query document for the completely plagiarized document, and distance 9 implies a heavily modified query document for a partial plagiarism. Table IV shows the experimental results of k-mer analysis using these input data when k=30.

TABLE IV: Experimental Result of Query Search

Input	# of Docs.	Plagiarism Dist.	Avg. Search (sec.)
E01	50	0	4.508
E02	50	1	3.843
E03	50	2	3.430
E04	50	3	3.247
E05	50	4	3.111
E06	50	5	3.016
E07	50	6	2.964
E08	50	7	2.908
E09	50	8	2.885
E10	50	9	2.848
Total	500	Average	3.276

Table IV shows the query retrieval results of fifty documents each at every plagiarism distance using the proposed method. In case of the documents included in the corpus without any deformation (E01), the search process takes long time because there are many occurrences of the fragments. On the other hand, in case of the heavily modified document, the process takes comparatively shorter time because there are fewer occurrences. In addition, we can see that the search time is very fast from the result.

C. Plagiarism Detection

In order to determine plagiarism, we need to detect similar sections using the query search and analyze the results comprehensively. We have used to the Incremental Density method to detect them. Precision and recall are the two out puts of the experiment. These are calculated by checking the result section that is included in the real document range of a corpus. In this experiment, k-mer analysis has the value from k=10 to k=30. It is difficult to find sections precisely with the value k lower than 10, and there are no changes for values of k higher than 30. The analysis result of detecting plagiarized section is shown in Fig. 4.

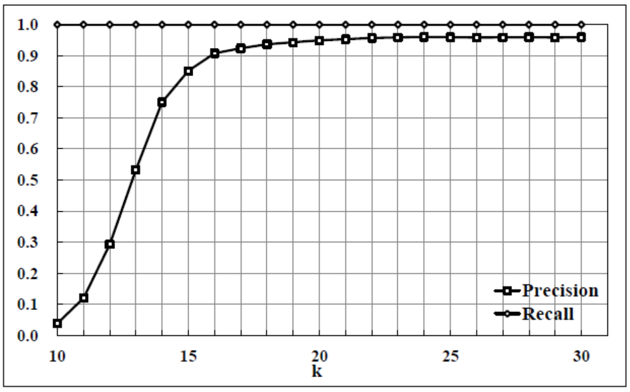


Fig. 4. A Plagiarism detection result graph for various **k**.

From Fig. 4, the precision is observed to be higher than 0.9 when *k*≥16. However, when *k*=10, the precision is very low because the probability of random occurrence in a corpus is high. Nevertheless, the actual plagiarized sections are also detected, because the precision value is not zero when *k*=10. In other words, the value of the precision is low just because unwanted sections are detected. Furthermore, the recall is always 1, because the fragment that exists on a corpus should always be found by the search process without fail.

VI. Conclusions

This paper proposed a method for plagiarized section detection in a large Korean corpus. The method, which involves the application of the short-read alignment (Genome assembly) used in NGS, can rapidly detect plagiarized sections. However, it is not a simple search technique. It utilizes the properties of a Korean language for searching, protection, and compression. In addition, disk-based BWT and FM-index increase the space utilization performance of the search. However, the methods proposed in this paper do not determine a case of actual plagiarism. We believe that our method should be employed by only those having the appropriate authority to determine plagiarism. This is because an accusation of plagiarism may adversely affect the reputation of a researcher. In conclusion, our method can reduce the time, space and labor required to detect plagiarized documents in a large document database.

The contributions of this paper can be summarized as follows:

- Compression and protection of the original document using the skin extraction method.
- Minimizing the space complexity of indexing using disk-based BWT.
- Minimizing the time and space complexity of the search using FM-index.
- Detecting similar sections using the incremental density method.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No.2012-0005518)

References

[1] H. Li and R. Durbin, “Fast and accurate short read alignment with burrows wheeler transform,” *Bioinformatics*, vol. 25, no. 14, pp. 1754-1760, July 2009.

[2] B. Langmead, C. Trapnell, M. Pop, and S. Salzberg, “Ultrafast and memory-efficient alignment of short dna sequences to the human genome,” *Genome Biology*, vol. 10, no. 3, pp. R25, 2009.

[3] M. Burrows and D. J. Wheeler, *A block-sorting lossless data compression algorithm*, Technical report, 1994.

[4] P. Ferragina and G. Manzini, “Opportunistic data structures with applications,” in *Proc. 41st Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, pp. 390, 2000.

[5] P. Ferragina, T. Gagie, and G. Manzini, “Lightweight data indexing and compression in external memory,” in *Proc. 9th Latin American conference on Theoretical Informatics*, Berlin, Heidelberg, 2010, pp. 697-710.

[6] A. Amir, A. Apostolico, G. M. Landau, and G. Satta, “Efficient text fingerprinting via parikh mapping,” *Journal of Discrete Algorithms*, vol. 1, no. 5-6, pp. 409-421, Oct. 2003.

[7] T. Smith and M. Waterman, “Identification of common molecular subsequences,” *Journal of Molecular Biology*, vol. 147, no. 1, pp. 195-197, 1981.

[8] U. Manber and G. Myers, “Suffix arrays: a new method for on-line string searches,” in *Proc. 1st annual ACM-SIAM symposium on Discrete algorithms*, Philadelphia, PA, USA, 1990, pp. 319-327.

[9] C. Lyon, J. Malcolm, and B. Dickerson, “Detecting short passages of similar text in large document collections,” in *Proc. 2001 Conference on Empirical Methods in Natural Language Processing*, 2001, pp. 118-125.

[10] M. Joy and M. Luck, *Plagiarism in programming assignments*, Technical report, Coventry, UK, 1998.

[11] G. Whale, “Identification of program similarity in large populations,” *Comput. J.*, vol. 33, no. 2, pp. 140-146, Apr. 1990.

Chang SeokOck is a M.S. student in Pusan National University. He received the B.S degree from Pusan National University. His research interests are HCI and String Processing.

Jong KyuSeo is a M.S. student in Pusan National University. He received the B.S degree from Pusan National University. His research interests are information retrieval and string sequence processing.

Sung-Hwan Kimis a M.S. student in Pusan National University. He received the B.S. degree from Pusan National University. His research interests are information retrieval and sequence processing.

Hwan-Gue Cho is a professor in Pusan National University. He received the B.S. degree from Seoul National University, Korea, and the M.S and Ph.D. degrees from Korea Advanced Institute of Science and Technology, Korea. His research interests are computer algorithms and bioinformatics.