Journal of Computer Science 7 (5): 644-650, 2011
ISSN 1549-3636

A Fast Hybrid Algorithm Approach for the Exact String Matching Problem Via Berry Ravindran and Alpha Skip Search Algorithms

Abdulwahab Ali Almazroi
Department of Computer and Information Technology, Faculty of Science and Arts at Khulais, King Abdul Aziz University, 80200, Jeddah, Saudi Arabia

Abstract: Problem statement: String matching algorithm had been an essential means for searching biological sequence database. With the constant expansion in scientific data such as DNA and Protein; the development of enhanced algorithms have even become more critical as the major concern had always been how to raise the performances of these search algorithms to meet challenges of scientific information. **Approach:** Therefore a new hybrid algorithm comprising Berry Ravindran (BR) and Alpha Skip Search (ASS) is presented. The concept is based on BR shift function and combines with ASS to ensure improved performance. **Results:** The results obtained in percentages from the proposed hybrid algorithm displayed superior results in terms of number of attempts and number of character comparisons than the original algorithms when various types of data namely DNA, Protein and English text are applied to appraise the hybrid performances. The enhancement of the proposed hybrid algorithm performs better at 71%, 60% and 63% when compared to Berry-Ravindran in DNA, Protein and English text correspondingly. Moreover the rate of enhancement over Alpha Skip Search algorithm in DNA, Protein and English text are 48%, 28% and 36% respectively. **Conclusion:** The new proposed hybrid algorithm is relevant for searching biological science sequence database and also other string search systems.

Key words: Sequence database, hybrid algorithm, string searching, Alpha Skip Search (ASS), Berry Ravindran (BR), searching biological, string matching algorithm, original algorithms

INTRODUCTION

In modern times string search algorithms continues to play a vital role for searching string applications (Almazroi and Rashid, 2011) specifically searching and matching patterns from vast and multifaceted science data. The goal of string matching algorithms (Raju and Babu, 2007; Lokman and Zain, 2010) is to discover all the occurrences of a specific known pattern Y= y1y2…ym in a text Z=z1z2…zn. The operations for all string algorithms entail the initial process by aligning the left last positions of the pattern and also the text, after that comparison is among the text and the pattern characters (Mohamed *et al.,* 2010). If a match or mismatch occurs between the pattern and the text, the pattern is shifted to the right. The processes continue till end of the pattern and also to the right end of the text is reached (Sleit *et al*., 2009; Deusdado and Paulo, 2009).

Additionally, the algorithms are made up of two phases namely pre-processing phase which examine the patterns to gain the needed information to be applied as a shift values for the pattern and the searching phases describe (Chen, 2007; Yuen *et al*., 2009; Radhakrishna *et al*., 2010) the arrangements comparison of characters in every attempt among pattern and text. The goodness of algorithms are measured in terms of number of character comparisons and number of attempts and also to ensure bigger shift values (Huang *et al*., 2008a) to boost performance. The search for improved hybrids performance (Pratumsuwan *et al.,* 2010) are leading to the development of new search algorithms, in this study a new hybrid algorithm consisting of Alpha Skip Search (ASS) and Berry-Ravindran (BR) algorithms is proposed and the aim is to extract the best features from each algorithm and combine them to enhance the performance of the new hybrid algorithm.

Mostly the algorithms distinction comes from number of shifting processes and the speed of detecting a match or a mismatch (Mohammad *et al*., 2006; Nadarajan and Zukarnain, 2008). The pre-processing phase presented by BR algorithm is the best as it ensures greater shift values. The strong point of BR is the reliability of being able to offer a greater shift value (Huang *et al*., 2008b) to shift the pattern in case of a match or mismatch. In the case of ASS algorithm searching phase, it is best known for searching strings

made up of three-character words. The advantage of ASS (Cantone *et al*., 2004) is the ability to check and verify the starting location in every attempt. However the weak spot is the event of a match or a mismatch it is limited throughout all the positions of the characters on shifting the patterns.

The ASSBR display efficient searching performance and this is due to the verifications which are carry out at every starting spot in every attempt, that in tend leads to best shift value when compared with the original algorithms, enabling the weaknesses to be overcome by the hybrid and these benefits are used in the searching phase to boost results.

Moreover some of the algorithms known to have a greater shift value (Klaib and Osborne, 2009) for example are Boyer-Moore (BM), Quick Search (QS) and BR. Another comparison will be also be conducted some algorithms such as Skip Search (SS) with the proposed hybrid algorithm, SS which is noted for small alphabets and long patterns and verify the starting positions in the text before starting the search and also Raita algorithm which is useful (Sheik *et al*., 2005) in searching patterns specially made of English texts and reality Raita has superior performance behavior in practice is due to the presence of character dependencies.

Previous work: Previous studies confirmed two algorithms combination always leads to higher search and matching performance in terms of number of attempts and number of character comparisons. Few of such hybrid algorithms are summarized as follows:

KMPBS hybrid algorithm (Xian-Feng *et al*., 2010) comprises of Boyer Moore (BM) and Knuth-Morris-Pratt (KMP) algorithms and the aim was to utilize characters which are positioned at end of the text and also the next bit of characters so as to derive maximum shifts values. The operations are done through the computations of the Next function of KMP and also the Right (p [j]) values representing BMHS algorithm at the pre-processing phase and at the searching phase the BM algorithm applied the next character to estimate the pattern string to be shifted according to the exact amount.

ZTFS (Cai *et al*., 2009) is also the combination of Zhu-Takaoka (ZT) and Fast Search (FS) algorithms. The operations for the algorithm entails two processes such as bmGs(j) for BM good suffix heuristic and ztBc(a,b) for ZT bad character heuristic representing the characters and with the ztBc(a,b) function providing the maximum shift values and moreover bmGs(j) function grant the prefix of the pattern during the pre-processing phase.

BHAC (Lin *et al*., 2011) on the other hand is composed of Backward Hashing (BH) and Aho-Corasick (AC) algorithms and it is important for tracking and also for scanning text information from virus. The intention is to index the shift table from the Prefix Sliding Window (PSW), representing the prefix for the search window. The BH also searches for longer patterns that are within PSW are examined because the shift value should not go past the PSW so this process enables time saving.

ShiftAnd-BMS algorithm (Smyth and Wang, 2009) was also implemented for searching both regular and indeterminate strings mainly in the form of English text. It is the integration of ShiftAnd and BMS (that is, the Sunday variant of the Boyer-Moore) algorithms. The improvement of this algorithm was achieved through BMS shifting and when a match occurs at end of pattern it changes to ShiftAnd matching algorithm and at the same time continue ShiftAnd matching till there is no match found at the present location, it skip to next location before finally to BMS shift.

To raise the performance another new algorithm called ZTBMH (Huang, *et al*., 2008c) was developed. The hybrid is made of Zhu-Takaoka (ZT) and Boyer-Moore-Horspool (BMH) algorithms and it is useful for searching in biological sequences database and also well-known for small alphabets and long patterns. The performance improvement was gained by applying one function at the pre-processing phase, that is, ztBc (a,b) for ZT bad character heuristic representing the characters and the ztBc(a,b) function of ZT providing the maximum shift values during the searching phase. The searching phase uses the ztBc(a,b) function which is used to compute the maximum shift value during each attempt. Whenever a mismatch occur ZT bad character shift value is used.

The BRBMH (Klaib and Osborne, 2008) is also improved performance hybrid algorithm consisting of Berry-Ravindran (BR) and Boyer-Moore Horspool (BMH) algorithms, designed especially for searching protein sequence database. The efficient performance was attributed to extraction of the good features from each algorithm involving the bad-character shift function in BMH, this defines one character that is next to the current text window and also the bad-character shift function in BR that identify two characters right next to the text window. BRBMH counts solely the shift value for the pattern characters rather than the text character and it also stores shift values in one-dimension array instead of two dimensional arrays by original algorithm BR thereby decreasing processing time.

MATERIALS AND METHODS

The technique included the review of all the search string algorithms before deciding on the two algorithms which best suit the research objectives and also the algorithms that can complement each other, hence the choice of ASS and BR. Because each algorithm has its own best features which can be leverage upon to enhance performance.

The BR bad character table is used and the bmBc (that is, Boyer Moore bad character) table applied for ASS algorithm. The main intention is to ensure greater shift values during the searching phase and BR is renowned for having the best shift values by employing the two successive characters immediately after the window. In the case of ASS algorithm, it is an improvement of Skip Search and it is useful (Cantone *et al*., 2004) for searching three character words in every attempt.

Pre-processing phase: The pre-processing phase is the initial point for constructing the algorithm and it involves the creating of BR bad character table needed for the calculating the shift values (Huang *et al*., 2008a), the bad character table for BR is represented as fellows:

$$\text{brBc}[m,n] = \min\begin{Bmatrix} 1 & \text{If } x[m-1]=m \\ m-i+1 & \text{If } x[i]\times[i+1]=mn \\ m+1 & \text{If } x[0]=n \\ m+2 & \text{Otherwise} \end{Bmatrix} \quad (1)$$

Searching phase: At this phase, the given pattern length is used to derive the necessary pattern matches from the text and the standard processes are describe as follows:

- Scrutinize the character of the text made up of m-length in order to segregate a probable location for the starting of the search spot
- Examine the last three characters that exist in the text window and if they do not exist in the pattern, the pattern is shifted by BR shift value involving the rightmost two successive letters following the assessment of character pattern length
- If the characters being scanned are found within the pattern, positioned the beginning characters seek spot with the same location within the pattern characters
- The next process is that the character comparisons are imitated from left to right of the window
- Finally, there will be a probability of a match or mismatch, when this event occurs, shift the pattern by computing the BR shift value from the two rightmost consecutive characters immediately after the window

Analysis: The proposed hybrid has $O(m+\sigma^2)$ time complexity, originating from (Huang *et al*., 2008b). The time complexity for the searching phase is demonstrated as fellows.

Lemma 1: The searching phase formula representing the time complexity for the proposed algorithm is characterize as O(mn) which indicates the worst case scenario.

Proof: The worst case derive from the proposed hybrid algorithm implies that the whole characters matches which are inside the text should not be more than m times. This issue arises when the characters of the pattern are the same characters in the text. For example given text T = “bbbbbbbbbbbbb” and pattern P = “bbbbb” and thus O(mn) is the worst case time complexity in this scenario.

Lemma 2: With the equation O(n/(m+2)) time complexity as a rule denoted to be the best case scenario.

Proof: Attempts are made on every three examining characters to verify whether the characters can be found in the pattern or otherwise, in view of this the shift value is represented as m+2 and it is calculated at the pre-processing phase of the bad character table. The best case scenario is known if the characters of the pattern are uniquely diverse from the characters in the text. For instance, when given text T = “bbbbbbbbbbbbbbbb” and pattern P = ”yyyyyyy”, the shift is symbolizes as m+2 at the searching phase that is perform at every attempt, thus the time complexity signifies O(n/(m+2)) as the best case scenario.

Another searching phase time is the average time complexity which involves alphabet size and also the character possibility occurrences of each character within the text. The equation is represented by m+2 which are derived from the maximum shift and 1-m is classified as minimum coming from the random addition in the input data. Average time is usually composed of random prediction of characters so in reality it is almost impossible to predict with accurate certainty.

Evaluation: Three types of data are used as the evaluation criteria, with each data having various degrees of pattern length and character size, before evaluation of results are describe for each algorithm, other algorithms namely Skip Search and Raita performance results are also analyzed and compared with the proposed hybrid. The three data used are, DNA, Protein and English text are the scientific benchmarks (Klaib and Osborne, 2009; Sharma and Singh, 2009) test data for testing algorithms performance, usually composed of sophisticated and gigantic data.

The hybrid and the original algorithms are executed 8 times and the size of alphabet for DNA data equal to four letters (σ = 4) and Protein data (Moreau, 2010) is equal to twenty letters (σ = 20) and 100 kinds of alphabets symbolizing every English text and even numbers and symbols and database is from Gutenberg project (Karkkainen and Na, 2006), also the test data for Protein came from Swiss-Prot Database (Klaib and Osborne, 2009). The pattern length ranges from 5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90 and 100 characters which are randomly selected.

The executing environment includes Personal Computer, Microsoft Windows Vista Service Pack 2 processor, speed of 1.93 GHz Intel Core 2 Duo Processor, 3GB of RAM and programming editor C++ 2010 Architect. Number of attempts and number of character comparisons are the performance criterion used to evaluate the hybrid (ASSBR) and the original algorithms.

Number of attempts: The number of attempt for any searching algorithm entails starting spot and then the foremost letter positioned in the pattern is plot to exact character in the text. The processes of movement persist to last position of the text and by so doing able to found the occurrence of a match or a mismatch before the searching concludes. This process is called number of attempts (Hudaib *et al*., 2008).

Number of comparisons: The number of character comparisons are also part of search algorithm performance criteria (Klaib and Osborne, 2008), the course of action involves the sections in a given text are initiated to be the first spot in the text and it is lengthen to the final character text, after that the characters are extracted individually before comparison is done among the characters and by so doing able to find the occurrences of a match or a mismatch.

RESULTS

Experimental results depict the new hybrid and the original algorithms and also that of Raita and Skip Search algorithms are highlighted with the size of the testing data of about 50MB. The numbers 5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90 and 100 characters are the diverse kinds of patterns lengths implemented in order to obtain requisite results. From Fig. 1-6, the results achieved emanated from two performance criteria areas are number of attempts and number of character comparisons.

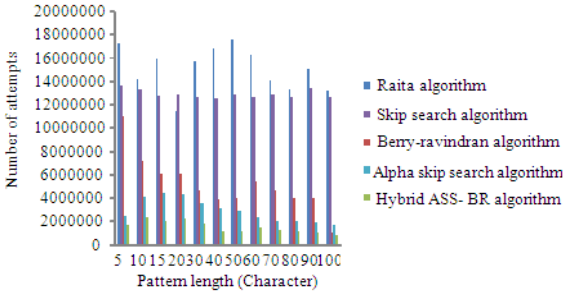

Fig. 1: Number of attempts in DNA data

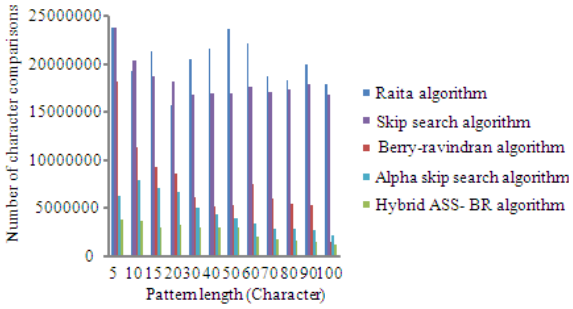

Fig. 2: Number of character comparisons in DNA data

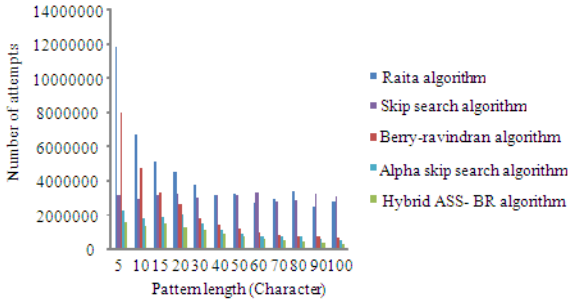

Fig. 3: Number of attempts in protein data

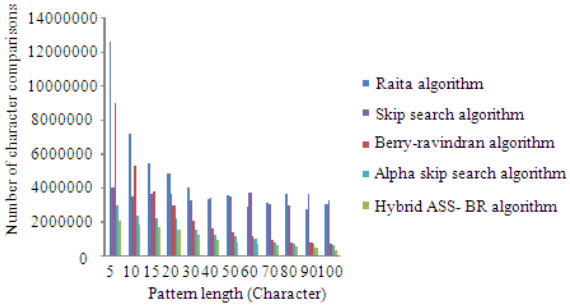

Fig. 4: Number of character comparisons in protein data

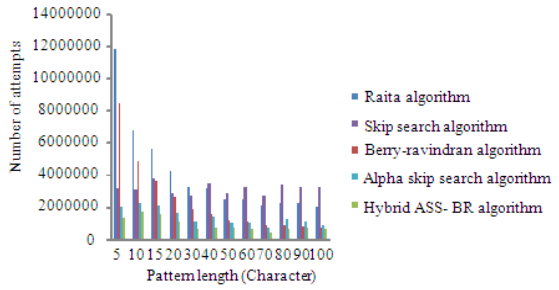

Fig. 5: Number of attempts in English text

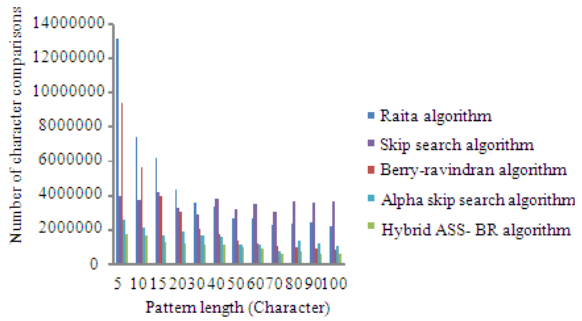

Fig. 6: Number of character comparisons in English text

After analyzing the experimental results of the algorithms, the proposed hybrid algorithm (ASSBR) displayed a far greater performance than the original algorithms in all the performance criteria's that is applied to evaluate the algorithm effectiveness no matter the type of data used. For example whether DNA, or Protein or English text data are used the hybrid algorithm still deliver the best performance than the entire original algorithms namely ASS and BR. Furthermore, also when compared with Raita algorithm and Skip Search algorithm, the proposed hybrid is still the best in terms of number of comparisons and number of attempts.

The key factor for the proposed hybrid improved performance is the attainment of a bigger shift value that is derived from the bad character table (brBc). This also attest the combination and the extraction of the good features from both algorithms have been beneficial and complimentary to the proposed hybrid, therefore the enhanced results springing up from the hybrid which might have been otherwise when the two algorithms are not compatible.

DISCUSSION

The key point of combination algorithms to form hybrid as always been to increase search performance and the choice of these algorithms to form hybrid must be carefully studied and the right ones selected. The pre-processing and the searching phases of the hybrid extraction from the original algorithms good attributes enabled increase performances.

The testing data include different types acquired various recognized resource centers such as Gutenberg project and Swiss-Prot Database for scientific data and information. However for the hybrid and the original algorithms three data are used, namely DNA, Protein and English text and the end conclusion is that the hybrid algorithm (ASSBR) performs significantly better than the original algorithms throughout and every stage of evaluation and even at different pattern lengths and sizes in the area of number of character comparisons and number of attempts.

Further results clarify the issues up, the hybrid enhancements in percentages terms are 71%, 60% and 63% over the BR and as well better at 48%, 28% and 36% when compared with ASS.

The prudent decision that can be made from the experimental results from hybrid algorithm is that, the rate of performance exceeds the use of a single string search algorithms as the number of attempts and number of comparisons are much less in the proposed hybrid. The use of biological science sequence databases are always progressing and becoming difficult, hence hybrid algorithm envisaged to be the way forward for string search algorithms developments as it enables enhanced performance due to the blending and extraction of each other's good properties, therefore offering the necessary complementary attributes and functions that enabled the formation of a high performing hybrid, for instance BR is well known for having greater shift value and ASS useful for three-character words and the benefit can be seen from the improvements in percentages of number of attempts and number of character comparisons when the hybrid compared with ASS and BR algorithms.

Having seen the efficient performance and results from research, the hybrid algorithm is highly recommended for string searching as it does offer the best of string searching and matching of patterns at much improved performance rates.

CONCLUSION

The research highlighted a new hybrid algorithm (ASSBR), which is the combination of Alpha Skip Search and Berry-Ravindran. Experimental results acquired proved that the ASSBR attained increase performances over the original algorithms in terms of number of attempts and number of character

comparisons using three types of data DNA, Protein and English text, with all the data having various kinds of pattern length and sizes. Even at any pattern length and size the hybrid is the best, thus this verifies the integration of the algorithms have derived the maximum benefits from each other.

REFERENCES

Almazroi, A.A. and N.A. Rashid, 2011. A fast hybrid algorithm for the exact string matching problem. Am. J. Eng. Applied Sci., 4: 102-107. DOI: 10.3844/ajeassp.2011.102.107

Cai, G., X. Nie and Y. Huang, 2009. A fast hybrid pattern matching algorithm for biological sequences. Proceedings of the IEEE 2nd International Conference on Biomedical Engineering and Informatics, Oct. 17-19, IEEE Xplore, Tianjin, pp: 1-5. DOI: 10.1109/BMEI.2009.5305645

Cantone, D., S. Cristofaro and S. Faro, 2004. Efficient algorithms for the δ-approximate string matching problem in musical sequences. Proceedings of the Prague Stringology Conference, (PSC'04), Universit`a di Catania, Italy, pp: 33-47.

Chen, Y., 2007. A new algorithm for subset matching problem. J. Comput. Sci., 3: 924-933. DOI: 10.3844/jcssp.2007.924.933

Deusdado, S. and C. Paulo, 2009. GRASPm: An efficient algorithm for exact pattern-matching in genomic sequences. Int. J. Bioinformat. Res. Appl., 5: 385-401. DOI: 10.1504/IJBRA.2009.027510

Huang, Y., L. Ping, X. Pan and G. Cai, 2008. A fast exact pattern matching algorithm for biological sequences. Proceedings of the International Conference on Biomedical Engineering and Informatics, May 27-30, IEEE Xplore, Sanya, pp: 8-12. DOI: 10.1109/BMEI.2008.154

Huang, Y., L. Ping, X. Pan, L. Jiang and X. Jiang, 2008. A fast improved pattern matching algorithm for biological sequences. Proceedings of the International Symposium on Computational Intelligence and Design, Oct. 17-18, IEEE Xplore, Wuhan, pp: 375-378. DOI: 10.1109/ISCID.2008.117

Huang, Y., X. Pan, Y. Gao and G. Cai, 2008. A fast pattern matching algorithm for biological sequences. Proceedings of the 2nd International Conference on Bioinformatics and Biomedical Engineering, May 16-18, IEEE Xplore, Shanghai, pp: 608-611. DOI: 10.1109/ICBBE.2008.148

Hudaib, A., R. Al-Khalid, D. Suleiman, M. Itriq and A. Al-Anani, 2008. A fast pattern matching algorithm with Two Sliding Windows (TSW). J. Comput. Sci., 4: 393-401. DOI: 10.3844/jcssp.2008.393.401

Karkkainen, J. and J.C. Na, 2006. Faster filters for approximate string matching. SIAM. http://www.siam.org/proceedings/alenex/2007/alx07_008karkkainenj.pdf

Klaib, A.F. and H. Osborne, 2008. Searching protein sequence databases using brbmh matching algorithm, Int. J. Comput. Sci. Network Secur., 8: 410-414.

Klaib, A.F. and H. Osborne, 2009. RSMA Matching Algorithm for Searching Biological Sequences. Proceedings of the 6th international conference on Innovations in information technology, Dec. 15-17, IEEE Xplore, Al Ain, pp: 195-199. DOI: 10.1109/IIT.2009.5413769

Lin, P., Y. Lin and Y. Lai, 2011. A hybrid algorithm of backward hashing and automaton tracking for virus scanning. IEEE Trans. Comput., 60: 594-601. DOI: 10.1109/TC.2010.95

Lokman, A.S. and J.M. Zain, 2010. One-match and all-match categories for keywords matching in chatbot. Am. J. Applied Sci., 7: 1406-1411. DOI: 10.3844/ajassp.2010.1406.1411

Mohamed, M.A., M.R.M. Said, K.A.M. Atan and Z.A. Zulkarnain, 2010. An improved binary method for scalar multiplication in elliptic curve cryptography. J. Math. Stat., 6: 28-33. DOI: 10.3844/jmssp.2010.28.33

Mohammad, A., O. Saleh and R.A. Abdeen, 2006. Occurrences algorithm for string searching based on brute-force algorithm. J. Comput. Sci., 2: 82-85. DOI: 10.3844/jcssp.2006.82.85

Moreau, V.H., 2010. Genomic distance between thymidylate synthase and dihydrofolate reductase genes does not correlate with phylogenetic evolution in bacteria. Am. J. Biochem. Biotechnol., 6: 35-39. DOI: 10.3844/ajbbsp.2010.35.39

Nadarajan, K. and Z.A. Zukarnain, 2008. Analysis of string matching compression algorithms. J. Comput. Sci., 4: 205-210. DOI: 10.3844/jcssp.2008.205.210

Pratumsuwan, P., S. Thongchai and S. Tansriwong, 2010. A hybrid of fuzzy and proportional-integral-derivative controller for electro-hydraulic position servo system. Energy Res. J., 1: 62-67. DOI: 10.3844/erjsp.2010.62.67

Radhakrishna, V., B. Phaneendra and V.S. Kumar, 2010. A two way pattern matching algorithm using sliding patterns. Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering, Aug. 20-22, IEEE Xplore, Chengdu, pp: 666-670. DOI: 10.1109/ICACTE.2010.5579739

Raju, S.V. and A.V. Babu, 2007. Parallel algorithms for string matching problem on single and two dimensional reconfigurable pipelined bus systems. J. Comput. Sci., 3: 754-759. DOI: 10.3844/jcssp.2007.754.759

Sharma, R. and A.K. Singh, 2009. Target identification in ory s1 pollen protein allergen from oryza sativa in the course of construction of hypoallergenic vaccines. Am. J. Infect. Dis., 5: 142-147. DOI: 10.3844/ajidsp.2009.142.147

Sheik, S.S., S.K. Aggarwal, A. Poddar, B. Sathiyabhama and N. Balakrishnan *et al*., 2005. Analysis of string-searching algorithms on biological sequence databases. Curr. Sci., 89: 368-374.

Sleit, A., W. AlMobaideen, M. Qatawneh and H. Saadeh, 2009. Efficient processing for binary submatrix matching. Am. J. Applied Sci., 6: 78-88. DOI: 10.3844/ajassp.2009.78.88

Smyth, W.F. and S. Wang, 2009. An adaptive hybrid pattern-matching algorithm on indeterminate strings, Int. J. Found. Comput. Sci., 20: 985-1004. DOI: 10.1142/S0129054109007005

Xian-Feng, H., Y. Yu-bao and X. Lu, 2010. Hybrid pattern-matching algorithm based on BM-KMP algorithm. Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering, Aug. 20-22, IEEE Xplore, Chengdu, pp: 310-313. DOI: 10.1109/ICACTE.2010.5579620

Yuen, C.T., M. Rizon, W.S. San and T.C. Seong, 2009. Facial features for template matching based face recognition. Am. J. Applied Sci., 6: 1897-1901. DOI:10.3844/ajassp.2009.1897.1901