International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:1, 2009

Fast Database Indexing for Large Protein Sequence Collections Using Parallel N-Gram Transformation Algorithm

Jehad A. H. Hammad, and Nur'Aini binti Abdul Rashid

***Abstract*—**With the rapid development in the field of life sciences and the flooding of genomic information, the need for faster and scalable searching methods has become urgent. One of the approaches that were investigated is indexing. The indexing methods have been categorized into three categories which are the length-based index algorithms, transformation-based algorithms and mixed techniques-based algorithms. In this research, we focused on the transformation based methods. We embedded the N-gram method into the transformation-based method to build an inverted index table. We then applied the parallel methods to speed up the index building time and to reduce the overall retrieval time when querying the genomic database. Our experiments show that the use of N-Gram transformation algorithm is an economical solution; it saves time and space too. The result shows that the size of the index is smaller than the size of the dataset when the size of N-Gram is 5 and 6. The parallel N-Gram transformation algorithm's results indicate that the uses of parallel programming with large dataset are promising which can be improved further.

***Keywords*—**Biological sequence, Database index, N-gram indexing, Parallel computing, Sequence retrieval.

I. Introduction

BIOINFORMATICS is one of the modern sciences that are a promising field and constantly evolving. Analyzing DNA and protein sequence is one of the main challenges in the bioinformatics field because of the large amount of biological data. The size of these databases increased exponentially, and the retrieval of the sequence becomes one of the main fields of study. The need to develop efficient genomic searching and retrieval tools became extremely important because it is a most time consuming process which required large working memory, high storage size and fast CPU speed. One of the methods that can help increase the speed of the searching and accessing database is database indexing. In general the main reason for using the database indexing technique is for faster searching and less computation cycles of CPU. Many of the computer scientists have made a lot of effort in the previous years to develop tools for faster searching. In this research, we propose to enhance an N-Gram transformation method by parallelize it in order to improve the computation time in building the index for large protein sequence database. We will justify our work by providing brief methodology followed by quantitative results in the form of plotted graph.

Authors are with School of Computer Sciences, University Sains Malaysia, USM, Penang, Malaysia (e-mail: Jhammad35@hotmail.com, nuraini@webmail.cs.usm.my).

II. Related Work

The special transformation based index algorithm used special transformation technique (likewise wavelet, metric analysis, genomic statistics etc) to construct an index. In this method, the sequence should be changed into vector with respect to time and frequency. The advantage of this kind of method is that it skips most of the unrelated sequences and decomposes the real search problem to only fractions of the original database.

Now we will discuss about some of the important techniques used by the transformation based index algorithm. Among the first is CAFÉ [31] partitioned based approach. In this approach, to rank the similarity to a query sequence, coarse searching with inverted index is used. The database subset with the query is used in a subsequent fine search (locally aligned). The index mainly consists of three components which are search structure, inverted lists and mapping table. To make index size more manageable, this method uses compression scheme. Importantly, CAFÉ method consists of coarse and fine search is marginally less accurate than BLAST1 and FASTA. From search point of view, CAFE is 8 times faster and efficient than the BLAST2 [29, 30]. PropSearch tool is proposed by the scientific and research society [13], have basic idea in database search is to utilize conserved properties in the similar structures. PropSearch have characteristics (such as hydrophobicity) carry more weight than lesser indicators. [13].

Bitmap indexing structure method is used to condense and encode the database sequences to smaller index sequence. The main advantage of this approach is reduces the response time but it adds more space overhead. The components of the algorithm are BIS construction procedure, filtering phase and result analysis (based on cost model) [3].

In the matrix space indexing techniques, [28] investigated two important techniques in order to support for fast query evaluation for local alignment. Incrementally Decreasing Cover-based (IDC) [14] algorithm extracts the homology candidates from genomic databases. New concept was introduced that has two-phase filtration namely "annote query

sequence" and "finding homology candidates". One of the important characteristics of this algorithm is that it is lossless filtration algorithm [14].

N-gram can be defined as a sequence of n consecutive characters. The N-Gram method is used in the various field of science including statistical natural language processing and genetic sequence analysis. The items of this method can be letters, words or protein sequence depending upon the application [9].

We can generate the set of N-Gram results by moving a window of n boxes on the text body. This can be done as sliding the window with the approximate length of n characters by one character in the specified text and each time record the characters in the window. This phenomenon is called "1-Sliding Technique" [21].

For that N-gram based index algorithm will be our choice solution for our research problem. [17] Mentioned in their future work that they will utilize the power of parallel computing architecture in order to get the efficient result for index based algorithm. Therefore in our work, we will use the parallel programming architecture to build the index in multi-core architecture.

III. Methodology

Our methodology includes two phases which are building the index sequentially, and the parallelization of the N-gram transformation algorithm. Fig. 1 shows the general scheme of our system architecture. There are four main steps in the indexing systems which are given below:

1. To download protein sequence database from public domain (Swiss-Prot)
2. To build the index using Transformation Based algorithm called N-Gram
3. To overlap the search's query using the N-Gram algorithm
4. Finally to search the overlapped query in the index and retrieve it.

Apart from the index systems main attributes, there are two matters to consider when parallelizing any existing sequential algorithm.

1. The first is whether to use data decomposition or task decomposition
2. Thread communication between each task in the algorithm.

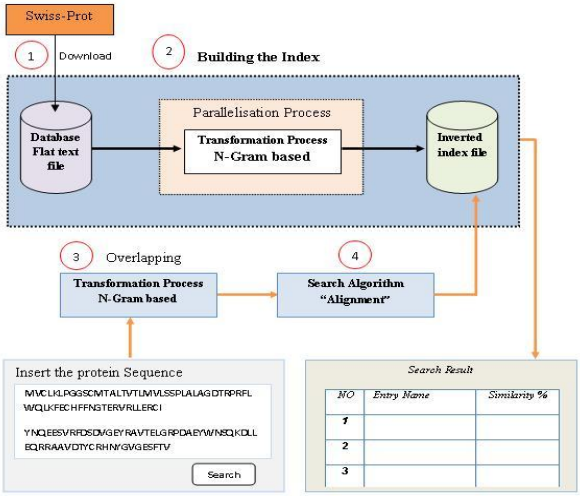

Fig. 1 General scheme

A. Design of Parallel Algorithm

In this stage we decide how to decompose computational activities and data into several small tasks.

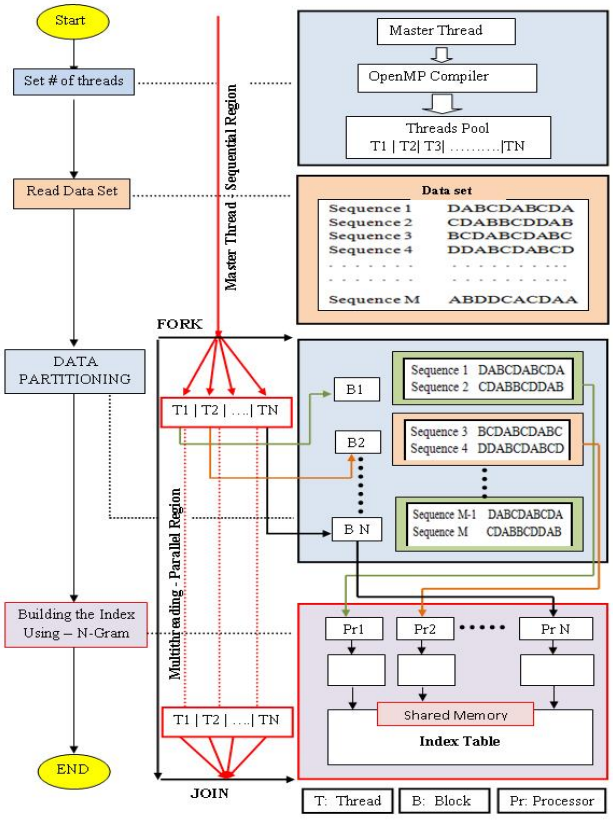

Fig. 2 Parallel N-Gram Transformation Algorithm Architecture

While we were at this stage we focused on how to decompose data and on the computational activities to perform a concurrent and a scalable program. Figure 2 shows the exact mapping of the algorithms to the multi processors.

Firstly, the number of thread has been initialized then the sequential process of the algorithm where data is read from the database. Afterwards the data is partition and distribute to each thread where thread executed the process of the overlapping for each block of data to build the index.

B. Data Partitioning

There are two basic ways to partition computational work among parallel tasks: domain (data) decomposition and functional decomposition. We will use the domain decomposition as a partitioning method in our research because we are dealing with independent data, and there is no relation between the different chunks of the data. This process breaks the data associated with a problem into discreet "chunks" of work that can be distributed into multiple tasks. In our problem we distributed the database sequences among the processors. The number of the chunks depends on the number of the processors. In our research we will use 1, 2, 4, and 8 processor. The process is more elaborated in the figure 3 which shows the data decomposition process.

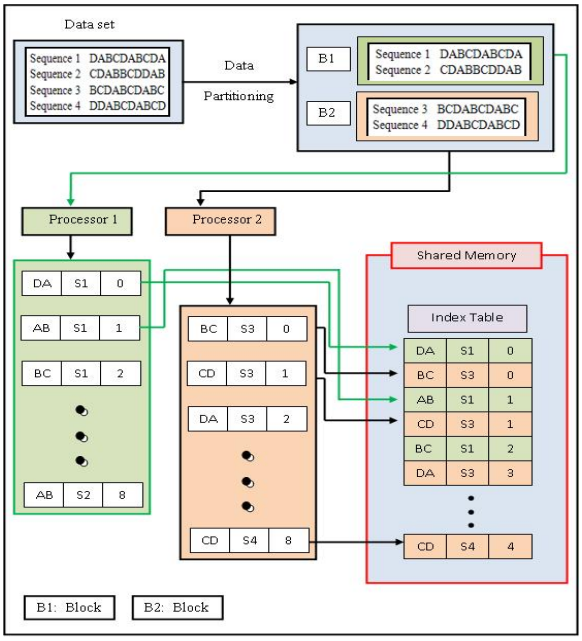

Fig. 3 Data Partitioning

IV. Result and Discussions

There are two phases of the quantitative research evaluation we made. The first one was the sequential experiment whereby at this phase we ran the program with one processor and at different word's length (gram) including 2,3,4,5 and 6 grams, with databases containing different number of sequences including 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 1000,1500, and 10000 sequences. The second phase is the parallel experiment which used 2, 4 and 8 processors with the same range of word's length and number of sequences used in sequential experiments. The aim of our experiments is to calculate the time taken to build the index sequentially and to calculate the speedup and efficiency values which will be used to evaluate the performance of the parallelization process.

A. Space Cost

The results shown in Table I and Figure 4 can be classified into 3 categories. When we used the 2 and 3 gram size words, the size of the index was greater than the size of the database. This happened because when we used the 2 and 3 gram size words the number of the words that was generated will be large and each index word need to record both the entry name and offset which need more space. In this case the space cost of the index is high. When we used the 4 gram size word the results we got is irregular. We encountered two cases, firstly with database sizes 22 KB, 68 KB, 107 KB, 153 KB, 192 KB and 4248 KB the size of the index was greater than the size of the database, but with database sizes 47 KB, 85 KB, 132 KB, 170 KB, 217 KB, 425 KB and 642 KB the size of the index was smaller than the database size. When we used 5 and 6 grams size words, the index was smaller than the database size for all database sizes. Hence space was saved.

TABLE I
INDEX SIZE IN KILOBYTES (KB)

# Sequences	Database size	2 Gram	3 Gram	4 Gram	5 Gram	6 Gram
50	22	263	48.6	34.1	32.3	32
100	47	548	68	36.1	33.2	32.5
150	68	824	135	81.4	71.4	64.8
200	85	972	144	81.8	71.4	64.8
250	107	1237	192	116	103	96.8
300	132	1480	212	118	104	97.3
350	153	1798	279	163	142	129
400	170	1945	288	163	142	129
450	192	2209	337	197	175	161
500	217	2494	356	199	176	162
1000	425	4820	716	402	350	314
1500	642	7261	1056	597	517	472
10000	4248	49702	7557	4525	4019	3745

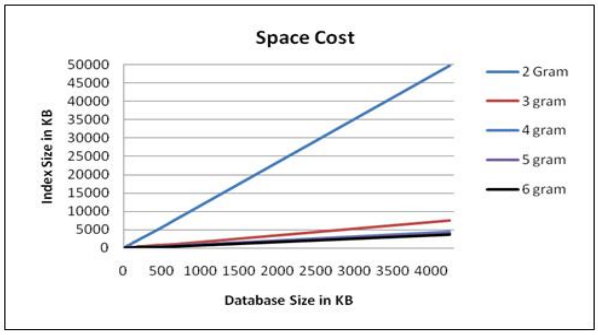

Fig. 4 Space Cost

Based on the behavior of the results we can conclude that the bigger the size of n-gram, the more the space is saved.

B. Time Cost

The results in Table II show that we get the minimum average time cost for building index when we used 8 threads with the 2 gram size word, and we get the maximum of the average time cost when we used 1 thread with the 6 gram size word. Based on the results shown in Figure 5, we can conclude that using 2 gram with 8 threads will be most

economical in index building instead of using 6 gram size word with 8 threads.

The results in Figure 5 shows that when we used a small size dataset, the sequential gave us a minimum of the average time cost compared with the parallel and by increasing the dataset size and number of the threads, the average of the time cost began to decrease, and we obtained the best result when we used 8 threads. By this, we can conclude that the sequential is economical in terms of time cost with small dataset and the results indicate that the parallel is more economical with the big size dataset. When the data is 10000 sequences, the best times is when we run it on 8 threads while with the smalls data size, the best time cost is with 1 thread.

TABLE II
AVERAGE TIME COST USING 1, 2, 4 AND 8 THREADS

Gram	1 Thread	2 Thread	4 Thread	8 Thread
2	22	263	48.6	34.1
3	47	548	68	36.1
4	68	824	135	81.4
5	85	972	144	81.8
6	107	1237	192	116

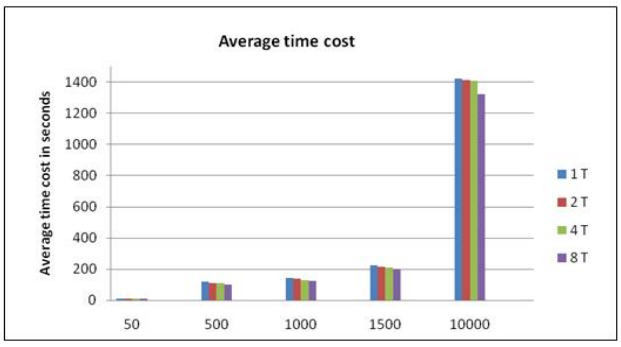

Fig. 5 Average Time Cost

C. Speed up

The results in Figure 6 show that the average values of the speed up was within the range from 1.014 to 1.066 .The highest average of the speed up was when we used 8 treads with a big size dataset, but when we decreased the number of the threads the average also decreased. By this, we can conclude that the use of the parallel methods is economical with a large dataset. Likewise, the sequential algorithm works best with small dataset.

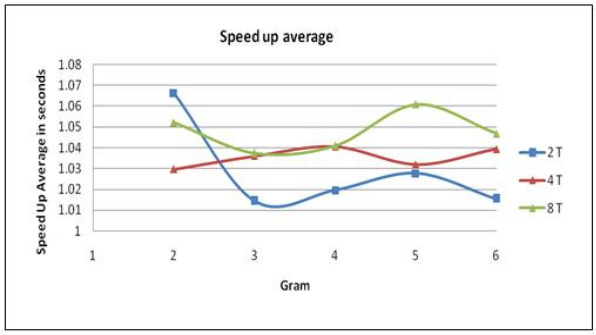

Fig. 6 Speed Up Average

D. Efficiency

The results in Figure 7 show that the average values of the efficiency were within the range from 0.393 to 0.630. The higher averages of the efficiency were when we used 2 Threads and it decreased when we increased the number of threads. By this, we can conclude that we failed to get high value efficiency but the maximum value was achieved when we used 2 Threads and the lowest value was achieved when we used 8 Thread. The reason is that we failed to get high speed up when we used a small size dataset.

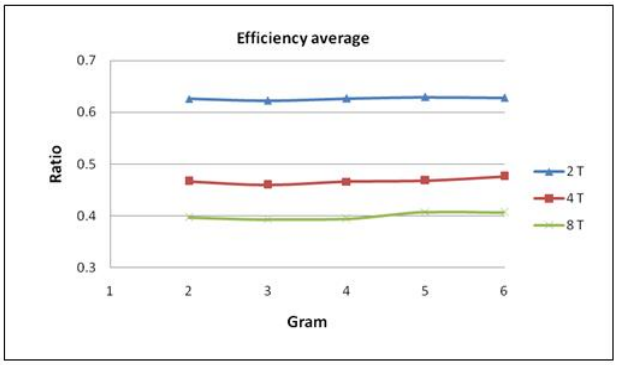

Fig. 7 Efficiency Average

V. CONCLUSION

This research makes one principal contribution that is the integration of N-gram transformation running time for building index table using N-gram algorithm and the improving of the sequential algorithm by using parallel methods. This idea appeared after the protein database became very huge, and at the same time the processors became more powerful even though with low price. This leads us to make use of the advantages of using parallel programming. This system is designed for multiple instruction multiple data (MIMD) and implemented on Khawarizmi multi-core Cluster, which is a dual quad core machine, using Intel C++ compiler, C++ language and OpenMP. In our design, we have no communication between the tasks because we used the multi-core machine with the shared memory. Our research results indicate that the use of parallel programming with large dataset has promising result which can be improved further.

Our research proves that the use of N-gram transformation algorithm is an economical solution. It saves time and space too, and the result shows that the size of the index is smaller than the size of the dataset when the N-gram sizes are 5 and 6.

ACKNOWLEDGMENT

The author would like to thank School of Computer Sciences, University Sains Malaysia (USM) for the research grant.

REFERENCES

[1] R. Bader, "OpenMP - Parallel programming on shared memory systems ": Leibniz-rechenzentrum. Retrieved September 14,2008 from: http://www.lrz-muenchen.de/services/software/parallel/openmp/, 2008.

[2] B. Barney, "Introduction to Parallel Computing." vol. 2008: Livermore Computing,National Laboratory. Retrieved July 4, 2008 from: https://computing.llnl.gov/tutorials/parallel_comp, 2007.
[3] O. Beng Chin, P. Hwee Hwa, W. Hao, W. Limsoon, and Y. Cui, "Fast filter-and-refine algorithms for subsequence selection," in Database Engineering and Applications Symposium, 2002. Proceedings. International, 2002, pp. 243-254.
[4] A. Califano and I. Rigoutsos, "FLASH: a fast look-up algorithm for string homology," in Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on, New York, NY, USA, 1993, pp. 353-359.
[5] G. Cooper, M. Raymer, T. Doom, D. Krane, and N. Futamura, "Indexing genomic databases," in Bioinformatics and Bioengineering, 2004. BIBE 2004. Proceedings. Fourth IEEE Symposium on, 2004, pp. 587-591.
[6] Q. Cory, "Introduction to programming shared-memory and distributed-memory parallel computers," Crossroads, vol. 8, pp. 16-22, 2002.
[7] H. Ela, P. A. Malcolm, and W. I. Robert, "A Database Index to Large Biological Sequences," in Proceedings of the 27th International Conference on Very Large Data Bases: Morgan Kaufmann Publishers Inc., 2001.
[8] H. Ela, P. A. Malcolm, and W. I. Robert, "Database indexing for large DNA and protein sequence collections," The VLDB Journal, vol. 11, pp. 256-271, 2002.
[9] Z. Elberrichi and B. Aljohar, "N-grams in Texts Categorization," Scientific Journal of King Faisal University (Basic and Applied Sciences), vol. Vol.8 No.2, pp. 25-38, 2007.
[10] C. Fondrat and P. Dessen, "A rapid access motif database (RAMdb) with a search algorithm for the retrieval patterns in nucleic acids or protein databanks," Comput. Appl. Biosci., vol. 11, pp. 273-279, June 1, 1995 1995.
[11] I. Foster, Designing and Building Parallel Programs. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc, 1995.
[12] A. Grama, A. Gupta, G. Karypis, and V. Kumar, Introduction to Parallel Computing, Second Edition ed.: Addison Wesley, 2003.
[13] U. Hobohm and C. Sander, "A Sequence Property Approach to Searching Protein Databases," Journal of Molecular Biology, vol. 251, pp. 390-399, 1995.
[14] L. Hsiao Ping, T. Yin Te, S. Ching Hua, S. Tzu Fang, and T. Chuan Yi, "An IDC-based algorithm for efficient homology filtration with guaranteed seriate coverage," in Bioinformatics and Bioengineering, 2004. BIBE 2004. Proceedings. Fourth IEEE Symposium on, 2004, pp. 395-402.
[15] M. Huerta, F. Haseltine, and Y. Liu. vol. 2008: National Institute of Mental Health (NIH) , Biomedical information science and technology initiative (BISTI), Retrieved February 6, 2008, from: http://www.bisti.nih.gov/CompuBioDef.pdf., 2000.
[16] M. N. Hwang and J. Kim, "Protein Sequence Search based on N-gram Indexing," Bioinformatics and Biosystems, vol. Vol. 1, pp. 53-57, 2006.
[17] X. Jiang, P. Zhang, X. Liu, and S. S. T. Yau, "Survey on index based homology search algorithms," Springer Science and Business /media, pp. 40:185-212, 23 March 2007 2007.
[18] T. Kahveci and A. K. Singh, "An Efficient Index Structure for String Databases," in Proceedings of the37th VLDB conference, Roma, Italy, 2001, pp. 351--360.
[19] T. Kahveci and A. K. Singh, "MAP: searching large genome databases," in Pacific Symposium on Biocomputing, Hawaii, 2003, pp. 303-314.
[20] W. J. Kent, "BLAT---The BLAST-Like Alignment Tool," Genome Res., pp. GR-2292R, March 20, 2002.
[21] M. S. Kim, K. Y. Whang, J. G. Lee, and M. J. Lee, "n-Gram/2L: A Space and Time Efficient Two-Level n-Gram Inverted Index Structure," in VLDB, Trondheim, Norway, 2005, pp. 325-336.
[22] C. D. Manning, P. Raghavan, and H. Schutze, Introduction to Information Retrieval: Cambridge University Press, 2008.
[23] S. Microsystems, "Multithreaded programming guide," Sun Microsystems, Inc Business. Retrieved September 14,2008 from: http://docsun.cites.uiuc.edu/sun_docs/C/solaris_9/SUNWdev/MTP/toc.html 2002.
[24] Z. B. Miled, N. Li, M. Mahoui, and O. Bukhres, "Information Retrieval in Biomedical Research," Wiley encyclopedia of biomedical engineering, 2006.
[25] Z. Ning, A. J. Cox, and J. C. Mullikin, "SSAHA: A Fast Search Method for Large DNA Databases," Genome Res., vol. 11, pp. 1725-1729, October 1, 2001 2001.
[26] T. H. Ong, K. L. Tan, and H. Wang, "Indexing Genomic Databases for Fast Homology Searching," in Proceedings of the 13th International Conference on Database and Expert Systems Applications, 2002.
[27] B. Parhami, Introduction to Parallel Processing Algorithms and Architectures New York: Kluwer Academic Publishers, 2002.
[28] C. Weimin and K. Aberer, "Efficient querying on genomic databases by using metric space indexing techniques," in Proceedings of the 8th International Workshop on Database and Expert Systems Applications: IEEE Computer Society, 1997.
[29] H. Williams and J. Zobel, "Indexing nucleotide databases for fast query evaluation," in Advances in Database Technology — EDBT '96, 1996, pp. 275-288.
[30] H. E. WILLIAMS, "Effective query filtering for fast homology searching," Pacific Symposium on Biocomputing, pp. 214 - 225 1999.
[31] H. E. Williams and J. Zobel, "Indexing and retrieval for genomic databases," Knowledge and Data Engineering, IEEE Transactions on, vol. 14, pp. 63-78, 2002.
[32] C. Xia, L. Shuai Cheng, O. Beng Chin, and K. H. T. Anthony, "Piers: an efficient model for similarity search in DNA sequence databases," SIGMOD Rec., vol. 33, pp. 39-44, 2004.
[33] L. Yip Chi and B. Kao, "A study on n-gram indexing of musical features," in International Conference on Multimedia and Expo, 2000 IEEE New York, NY, USA, 2000, pp. 869-872 vol.2.

Jehad A. H. Hammad is currently at the end of his postgraduate student tenure and researcher in parallel and distributed computing research group in School of Computer Sciences, University Sains Malaysia (USM). His research interests include Parallel and Distributed Computing Architecture, Advance Network and Data Communication. Jehad was a supervisor and member of inclusive Education team at the Directorate of Education, Bethlehem, Palestine. The Author is looking forward to further research and innovate in the field of Bioinformatics and related studies by putting his vast industrial experience.

Nur'Aini Abdul Rashid received a BSc from Mississippi State University, USA, and her MSc and PhD from University Sains Malaysia, Malaysia, all in Computer Science. Her Ph.D research involved analyzing and managing protein sequence data. Currently, she is a senior lecturer at the School of Computer Sciences at the University Sains Malaysia. Nur'Aini research interests include parallel algorithms, information retrieval methods and clustering algorithms.