A Study of Performance of Longest Common Subsequence Identification with Sequence Identity of Biosequences

Conference Paper · July 2011

DOI: 10.1049/cp.2011.0446

2 authors:

Sumathy Eswaran
Dr. M.G.R. University
31 PUBLICATIONS **59** CITATIONS

Rajagopalan S P
Dr.MGR Educational and Research Institute
182 PUBLICATIONS **939** CITATIONS

IET Chennai and Dr.MGR University Second International Conference on Sustainable Energy and Intelligent System (SEISCON 2011), Dr. M.G.R. University, Maduravoyal, Chennai, Tamil Nadu, India. July. 20-22, 2011.

A Study of Performance of Longest Common Subsequence Identification with Sequence Identity of Biosequences

Sumathy Eswaran* and S.P.Rajagopalan†

Keywords: SRLCS, LCS, Heuristic, Homology, Identity.

Abstract

Searching for clue to the result with biosequences is an important area of research for computational scientists in bioinformatics. The sequences are longer and demand more and more computational power in order that the result yields benefits to the society. More often the computational results are used in obtaining quick clue to the expected results of lengthy laboratory process. The identity and similarity between sequences provide the basic clue and guidance as to how to progress with work. This paper analyses SRLCS algorithm with the tools like CLUSTAL-W, and MUSCLE in identifying Longest Common Subsequence (LCS) with reference to identity between the bio sequences.

1 Introduction

The availability of computational power on account of technological advances has benefited many fields including Computational Biology. Computational biologists analyse biosequences of protein, DNA, Gene etc to know their relevance in another organism of interest like evolutionary, functional or structural relationship. Sequence similarity is the basis for many interesting findings for computational biologists like providing information about conserved region, identifying the presence of foreign genome in an organism, identifying the structural and functional relationship between two sequences or knowing about evolutionary and homologous relationships.

Two sequences are said to be similar if the order of sequence characters is recognizably the same in the sequences and is usually found by showing that they can be aligned. The first step to finding the sequence similarity is identifying Longest Common Subsequence (LCS). LCS problem determines the longest ordered sequence(s) found between the given sequences. LCS is computationally complex problem when the sequences are longer. Some Biosequences of Gene can run into Mega basepair order. Identification of LCS between more than 2 sequences is said to be an MLCS or Multiple Sequence Alignment (MSA) problem. MLCS problem is NP-hard.

The computational complexity of LCS problem is directly proportional to (i) dissimilarity between the sequences, (ii) size of ∑ where ∑ is the alphabets the sequence is made up of and (iii) the size of the sequences themselves. The complexity is further more when the problem is dealt as Multiple Longest Sequence problem (MLCS).

Sometimes finding an optimal MLCS is often computationally not feasible. Many algorithms have been derived towards reducing the resource requirement. A close to optimal solution or clue towards worthiness or necessity to investigate further may be of great lead to biologists. Therefore a heuristic approach to identifying LCS by SRLCS[22] is studied with other known familiar MSA tools like CLUSTAL-W[1] and MUSCLE[9].

2 Related Works

Dynamic programming is the mother of all in solving alignment problems. Smith–Waterman[20] for Local alignment and Needleman-Wunsch[16] for global alignment. Dynamic programming solution complexity is O(nm) for both time and space for m sequences of length n. Decision tree model by Aho and et al.[2] gave lower bound of O(mn). Hirschberg[12] solution reduces the space complexity to O(m+n).

Lot of work has been done and many algorithms have been developed towards reducing the complexity. Parallel algorithms can divide the problem and hence can handle computational complexity to a large extent. The parallel algorithms like FastLCS [25], EFPLCS [21] and parMLCS [17] gave near linear speed up for large number of sequences. FastLCS complexity is O(|LCS(X,Y)|) for time complexity and max{4*(n+1)+4*(m+1), L} for space complexity. EFP LCS is 70% more efficient than FASTLCS in resource utilization of both memory and CPU.

However as said earlier there is a need for trade off between accurate and suboptimal acceptable solution , while dealing with large sequences. Heuristics algorithms take this place by identifying LCS within reasonable resource requirement. The heuristic parameter determines the solution quality. Solution quality can be set to the acceptable limit by the user with reference to the problem in hand. Heuristic algorithms reduce the search space. Time Horizon Specialised Branching Heuristic (THSB)[23], Ant Colony Optimization (ASO)[19], Beam Search[4] are all heuristic algorithms while MLCS APP[18], SRLCS[22] are heuristic parallel algorithms.

SRLCS algorithm accepts bounding reference (h) set by the user according to the solution quality expectation. This when applied to the unexpended length of the shorter sequence determines the candidates for pruning and hence reducing the search space. This is represented by the equation (1) given below:

$$f(p) = g(p) + h(p) \quad (1)$$

where g(p) = f(Probable candidate for LCS contribution) and h(p) = purpose function from the user.

3. Tools for LCS identification

3.1 Clustal –W

CLUSTAL–W[1]is a popular general purpose Multiple Sequence Alignment (MSA) program for DNA or Protein sequences. CLUSTAL -W calculates the best match for the selected sequences and lines them up for display so that identities, similarities and differences can be seen. CLUSTAL-W uses progressive alignment method. CLUSTAL-W 2.0.12 multiple sequence alignment program windows version was downloaded from the European Bioinformatics Institute (EBI)[10]

3.2 Muscle

MUSCLE stands for MUltiple Sequence Comparison by Log- Expectation. MUSCLE[9] is claimed to achieve both better average accuracy and better speed than ClustalW2 or T-Coffee[26], depending on the chosen options. MUSCLE attempts to do alignment using progressive and iterative method from the k-tuple subsequences of the sequences. MUSCLE v3.8.31 by Robert C Edgar from public domain was downloaded and used.

3.3 SRLCS

SRLCS identifies LCS by creating Successor Table for each of the sequences which will have successor entries for each element in ∑ where ∑ is the set of elements in the sequence. Then starting from Initial Identical Pair, Successor pairs are generated. Based on dominant successor, the surely less dominant ones are pruned at each stage. When no more successors, Pair table is back tracked to collect LCS. |LCS| = Maximum level in Pair Table. Heuristic pruning is applied based on h function to discard those successors which are unimportant for target solution quality.

4 Experiment and Results

Pair wise LCS identification was done on CLUSTAL–W, MUSCLE and SRLCS on Protein Sequences of about length 200. Since a Desktop Intel Pentium system with 2GB memory was used, pairwise comparison was done. On a powerful configuration, MLCS can be identified.

Eight sequences each from 3 different families of Pfamseq database [14] were taken for testing. In each family one sequence was used as Query string and compared with other 7 strings. In all, 24 sets of data having similarity from 28% to 88% were used. The results were observed for optimal LCS and performance of these algorithms with reference to identity between sequences.

4.1 Result 1

Sequences from PF03678 family of pfamseq database were taken. Both MUSCLE and SRLCS are able to produce LCS between sequences of varying length. The sequences size is limited to about 230 as the system on which the experiment was done had only 2GB RAM. The results are tabled in Table.1. It is observed that although SRLCS requires more memory than MUSCLE when the pair wise identity is less than 80%, it brings out the optimal LCS. However when the pair wise identity between the sequences is above 80%, SRLCS requires less memory than MUSCLE. Hence SRLCS could be used with more efficiency when the target user's purpose is to identify the possibility of presence of subsequence or near subsequence, a case fit to be a homology. It is important to note that SRLCS can bring out the all the LCS possible as seen in column(5) of table.1. This could be a useful feature when one is working on evoluting the sequences to identify distant homology.

Sequence X Length	Sequence Y Length	Identity % between two sequences	LCS by SRLCS	No of LCS by SRLCS	Memory used by SRLCS in MB	Muscle LCS Length	Memory Used by Muscle in MB
204	215	24	85	432	913	57	3
175	172	44	94	288	726	83	3
204	207	52	123	6	439	109	3
175	175	65	125	4	70	117	3
227	228	75	176	4	410	173	4
227	228	78	182	1	45	179	4
175	177	84	150	1	2	150	3
227	227	88	200	1	2	199	4
227	227	88	200	9	3	200	4

4.2 Result 2

Sequences from families PF10786, PF03678, PF10108.2 were used. LCS identification within the family was done. One each from each family was a query sequence while 8 others were used as reference sequences. These had pair wise identity percentage ranging from 24 to 88 and length from 169 to 228. The graph in Figure.1.shows the behavior of the three methods i.e. SRLCS, CLUSTAL-W and MUSCLE in obtaining the optimal LCS when the identity between the sequences differs. While all the three perform correct on higher identity between the sequences, SRLCS still performs better on lower identity by providing optimal LCS. The numeric comparison is enumerated in table.2 for PF03678 family experiment.

Similar experiment was done on other two families as mentioned earlier.

The average length of sequences taken for test is 207 in PF10786 sequences with average identity of 52%. The LCS yield by CLUSTAL-W, SRLCS and MUSCLE respectively were 108, 122 and 109.

With PF10108.2; Exon_PolB family of sequences the average identity was 68% with average length 175 and the LCS yield CLUSTAL, SRLCS and MUSCLE respectively were122, 126 and 122. From the observations it is inferred that the SRLCS is consistent to give optimum LCS irrespective of the length of the sequences or the length of LCS.

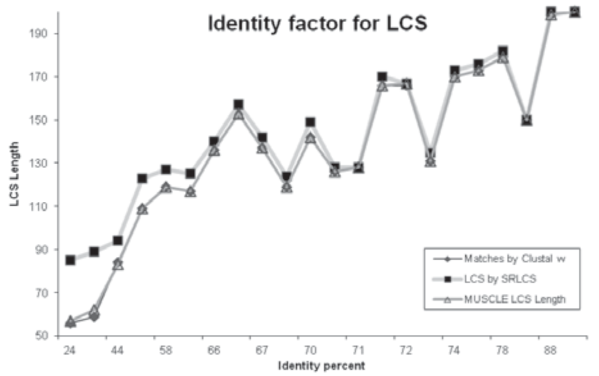

Figure 1. Pairwise Identity vs. LCS identification of SRLCS, CLUSTAL-W and MUSCLE

Table 2. Clustal - SRLCS- MUSCLE comparison on LCS identification with PF03678.7Adeno_hexon_C

Sequence name	Average Length of seq	Average Identity in %	LCS by Clustal w	LCS by SRLCS	LCS by MUS-CLE
Q76I40_9ADEN/10-236 Ref string length 227					
HEX_ADEM 1/592-819	228	66	153	157	153
O39793_ADE E1/596-823	228	72	166	170	166
O40957_ADE E2/586-812	227	72	166	167	167
Q9IF30_ADE-BA/597-824	228	74	170	173	170
B3VQN1_AD EC2/588-815	228	75	173	176	173
Q8B661_ADE T1/594-821	228	78	179	182	179
HEX_ADE05 /636-862	227	88	199	200	199
B2ZX08_ADE 40/607-833	227	88	200	200	200
Average	228	77	176	178	176

4.3 Result 3

With regard to providing optimal LCS, the precision was measured.

$$\text{Precision} = \frac{\text{The length of the common subsequence computed by the algorithm}}{\text{The length of the longest common subsequence in correct match}}$$

It is also observed that SRLCS precision is maintained at 100% while CLUSTAL and MUSCLE achieve precision only when the identity between sequences is above 80%. Figure.2.shows the graph of the results obtained in this regard on the same set of pfam sequences.

5 Conlcusion

Heuristic algorithms cannot be directly compared with one another as the performance depends on the heuristic function. From the above results, Performance of SRLCS with regard to detection

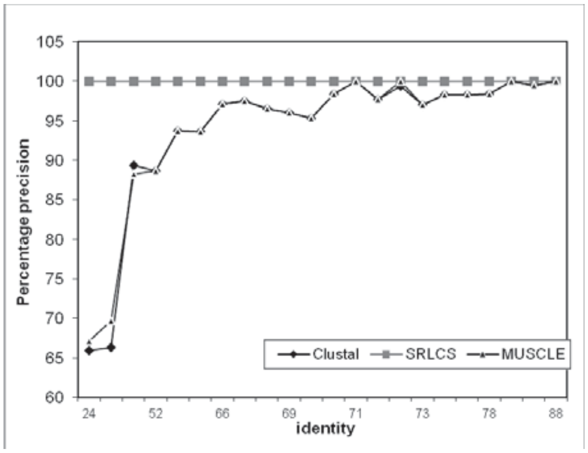

Figure 2. Precision to identify LCS by SRLCS , CLUSTAL-W and MUSCLE

of Homology and subsequence is satisfactory. Further SRLCS can be implemented using threads or as parallel implementation [22]. It is also scalable for MSA [22]. Most importantly SRLCS can enumerate all the possible LCS and not just one.

References

[1] "Clustal W and Clustal X version 2.0", Larkin M., et al. Bioinformatics 2007 23(21):2947-2948

[2] A.Aho, D.Hirschberg and Jullman, 1976, Bounds on the Complexity of the Longest Common Subsequence Problem, J.Assoc.Comput.Mach., Vol. 23, No.1,1976

[3] Anoop Kumar and Lenore Cowen , Augmented Training of Hidden Markov Models to recognize remote homologs via simulated evolution, bioinformatics/btp265, vol25, no.13 2009

[4] Blum, C.; Blesa, M. J.; and L´opez-Ib´a´nez, M.. Beam search for the longest common subsequence problem. Comput. Oper. Res. 36(12):2009, 3178–3186.

[5] BobGross, Multiple Sequences alignments , Bio68

[6] Bryan Bergeron,M.D., Bioinformatics computing, Pearson Education publication

[7] Dan E.Krane, Michael L.Raymer, Fundamental Concepts of BioInformatics, Pearson Education

[8] David W Mount , Bioinformatics Sequence and Genome Analysis, CBS Publishers

[9] Edgar, R.C. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Research, Vol. 32, No. 5, , 2004, 1792-1797

[10] ftp://ftp.ebi.ac.uk/pub/software/clustalw2/2.0.12/

[11] Hakata, K., and Imai, H. Algorithms for the longest common subsequence problem for multiple strings based on geometric maxima. Optimization Methods and Software 10:233–260, 1998.

[12] Hirschberg, D.S. Algorithms for the longest common subsequence problem. J. ACM 24(4):664–675., 1977.

[13] http://pfam.janelia.org/

[14] http://pfam.sanger.ac.uk/

[15] L.Bergroth, H.Hakonen and T.Raita, A survey of longest common subsequence algorithms, Seventh International

Symposium on string Processing information Retieval, 2000.

[16] Needleman SB, Wunsch CD: A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol, 48(3):443-453, 1970.

[17] Qingguo Wang, Dmitry Korkin and Yi Shang , Efficient Dominant Point Algorithms for the Multiple Longest Common Subsequence(MLCS) problem , IJCAI 2009/ 1494–1500.

[18] Qingguo Wang, Mian Pan, Yi Shang and Dmitry Korkin, A Fast Heuristic Search Algorithm for Finding the Longest Common Subsequence of Multiple Strings, Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10), , 2010.

[19] Shyu, S. J., and Tsai, C.-Y. Finding the longest common subsequence for multiple biological sequences by ant colony optimization. Comput. Oper. Res. 36(1):73–91., 2009.

[20] Smith t.F., Waterman M.S,Identification of common molecular subsequence, Journal of Molecular Biology, Vol.215, 1990

[21] Sumathy Eswaran, S.P.Rajagopalan, An Efficient Fast Pruned Algorithm for finding Longest Common Sequences in Bio Sequences, Annals.Computer Science Series, 8th Tome, 1st Fasc, page 137 – 150, 2010.

[22] Sumathy Eswaran, S.P.Rajagopalan, “Heuristic SRLCS Algorithm to determine the proper alignment strategy for Biosequences, International Journal of Research and Reviews in Information Technology (IJRRIT), Vol. 1, No. 2, 1-7, June 2011, ISSN: 2046-6501

[23] Todd Easton, Abhilash Singireddy, 2008, A large neighborhood search heuristic for the longest common subsequence problem, J Heuristics 14:271-283, 2008.

[24] V.Freschi and A.Bogliolo, Longest Common Subsequences between runlength encode strings: a new algorithm with improved parallelism, Information Processing Letters, 2004.

[25] Wei Liu, Lin Chen, A Fast Longest Common Subsequence Algorithm for Biosequences Alignment, IFIP vol 258, 2008.

[26] Cédric Notredame et al, T-coffee: a novel method for fast and accurate multiple sequence alignment, Journal of Molecular Biology, Volume 302, Issue 1, 8, Pages 205-217 , 2000.

Biographies

Sumathy Eswaran A.M.I.E., M.Tech., (Ph.d) is a Research Scholar at Vels university and working as Associate Professor at Dr.MGR Educational and Research Institute University , Chennai, India. The author brings in 15 years of experience from the computer industry at M/S.ECIL and M/s.OMC Computers Limited. And now adds 7 years of Teaching Experience at UG and PG level Engineering Courses .

Dr. S. P. RajaGopalan former Dean College Development Council, Madras University, instrumental for the MCA curriculum in the country, now Professor Emeritus at Dr. MGR Educational and Research Institute, Chennai. He has more than 150 papers published in international journals and has supervised 16 Ph.ds. Dr.S.P.Rajagopalan is a registered Research supervisor at many of the well known universities. He has written many books in mathematics and computer science fields.