

Related works:

(Assignment2)

- For locating LCS, **Tripathi and Pandey**(2016) discussed two similarity algorithms: maximum common sub stream (MCS) and Rabin-Karp (R-K). Rabin-Karp is better than (MCS). International Conference System Modeling & Advancement in Research Trends (SMART), Moradabad, pp.334–338.
- **Dheenadayalan** (2013) Using hash-based common substring with suffix tree (HCSST) and HCS with separate chaining (HCSSC) calculations, suggested memory-efficient solutions for discovering standard substrings in various arrangements. International Conference on Technology, Informatics, Management, Engineering and Environment, Bandung, pp.140–145
- **Yang** (2013) Pro-MLCS with prevailing point method to deal with locate the fundamental substrings was expected. Genius MLCS can quickly find an expected arrangement and then continue to deliver better results. until getting the ideal one. ‘a new progressive algorithm for a multiple longest common subsequences problem and its efficient parallelization’, IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 5, pp.862–870.
- **Rubi and Arockiam** (2012) suggested Positional LCS as a way to reduce time complexity. This an algorithm based on position to find (LCS) in a sequence database (SDB). And in (2016) proposed a method called Decode-HMM-MLCS. Journal of Chemical and Pharmaceutical Sciences (JCPS), Vol. 9, No. 1, pp.59–64. IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, pp.1–4
- **Alsmadi and Nuser** (2012) proposed the two algorithms called longest common substring and longest common subsequences (LCS, LCSS) to compare the DNA sequences. International Journal of Advanced Science and Technology, Vol. 47, pp.13–32.
- **Wang** (2011) suggested another calculation for finding a LCS of any number of strings. This algorithm is focused a concept called dominant point approach. Its calculation depends on divide and conquer technique. IEEE Trans. Knowledge and Data Eng., Vol. 23, No. 3, pp.321–334.
- **Shukla and Agarwal** (2010) proposed a simple and time-efficient parallel calculation. That computes the relative places of characters, which is used to determine the LCS of DNA, RNA, protein. International Conference on Computer and Communication Technology (ICCCCT), Allahabad, Uttar Pradesh, pp.496–502.
- **Rizvi and Agarwal** (2007) suggested a calculation for finding LCS from two given groupings of DNA and Proteins. and in (2006) suggested a technique which analyses the database grouping of DNA representing genome of some living. (2007) IEEE 33rd Annual Northeast Bioengineering Conference, Long Island, NY, pp.302–306. (2016) Third International Conference on Information Technology: New Generations (ITNG’06), Las Vegas, NV, pp.560–561.
- **Beal** (2015) proposed a compression method for solving LCS problem in genome resequencing data. IEEE International Conference on Bioinformatics and Biomedicine 2015, doi: 10.1186/s12864-016-2793-0.
- **Ozkan and Turksen** (2015), suggested a fuzzy C-means (FCM) calculation for coordinating the LCS. That computes in two steps. <https://www.researchgate.net/publication/281084370> (accessed 29 May 2018).
- **Lavanya and Murugan** (2013), suggested two techniques for finding (MLCS) and (MSCS). In first technique uses support vector. In second technique uses Positional Weight Matrix. International Journal of Engineering and Technology(IJET), Apr–May, Vol. 5, No. 2, pp.1153–1161.
- **Peng and Wang** (2017) to recognize MLCS, proposed a graph-based model called levelled-DAG. Frontiers in Genetics, Vol. 8, Article 104, pp.1–13.

- **Chen (2017)** proposed an algorithm for automatic international disease classification based on similarities using LCS. PLoS ONE, Vol. 12, No. 3, e0173410 [online]
<http://sci-hub.tw/https://doi.org/10.1371/journal.pone.0173410>.