Fast sequence clustering using a suffix array algorithm(2003)

Ketil Malde, Eivind Coward and Inge Jonassen

The paper presents a fast algorithm for clustering expressed sequence tags (EST), based on suffix arrays. The algorithm avoids all-against-all comparison by calculating longest common prefix for two strings in lexicographically ordered array of all suffixes of all ESTs.

Suffix array is built, using a leftwise radix sort algorithm. Theoretically, the time bound, which is $O(n \log n)$, can be improved with linear-time Karkkainen-Saunders algorithm [**?**]. But in practice this algorithm doesn't lead to a significant improvment due to random memory access and higher memory consumption.

Recursive sorting algorithm also leads to memory consumption, so authors propose perform this algorithm on subsets of data. For all words of selected length l they extract from the data all suffixed that have this word as a prefix and constuct suffix array for this subset of data. Then they identify cliques, generate and retain pairs. After that space can be reclaimed.

Proposed measure of similarity for a pair of sequences depends on the sum of lengths of blocks longer than parameter k, which represent the number of exact matches between sequences. So two truly similar ESTs with a lot of SNPs may not be clustered together. On the contrary, algorithms, where similarity is defined as stringency (not exact matching) in a window of fixed length are more specific.

Some steps of the proposed approach require clarification, for instance, there's no description of dynamic programming algorithm, used for construction the largest consistent set from the matching blocks already identified. Also authors don't explain how to merge clusters, even though all other steps of cluster construction are described.

Algorithm performance was compare to established clustering tools. Provided optimal parameters of block size and threshold the algorithm shows good scalabilty. However, a good measure of clustering quality is hard to come up with. The measure used in the paper is the number of *exactly* mathcing clusters between different clustering tools. The only tool which outperforms the one presented in the paper is UniGene. But it does not contain all of the benchmark sequences and contains additional sequences, that can impact clustering, so results of UniGene are only indicative.

In conclusion authors mention some possible improvements. Implementing the performance-critical parts of the algorithm in a lower level language and parallel implementation should increase scalability. And to provide better specificity authors plan to explore using gap penalties.

References

[1] Juha Kärkkäinen and Peter Sanders. Simple Linear Work Suffix Array Construction. Linear work suffix array construction. *J. ACM*, 01/2006; 53:918-936.

[2] John Burke, Dan Davison, Winston Hide. d2_cluster: A Validated Method for Clustering EST and Full-Length cDNA Sequences. *Genome Res*, 1999 November; 9(11): 1135–1142.