Published online

Nucleic Acids Research, , Vol. , No. ***1–5***
doi:10.1093/nar/gkn000

PEACE: Parallel Environment for Assembly and Clustering of Gene Expression

D.M. Rao [1], J.C. Moler [1], M. Ozden [1], Y. Zhang [1], C. Liang[1,2*] and J.E. Karro [1,3*]

[1] Department of Computer Science and Software Engineering,
[2] Department of Botany,
[3] and Department of Microbiology, Miami University, Oxford, Ohio, USA

Received Feb. 13, 2010
Resubmitted April 21, 2010

ABSTRACT

We present PEACE, a stand alone tool for high-throughput *ab initio* clustering of transcript fragment sequences produced by Next Generation or Sanger Sequencing technologies. It is freely available from www.peace-tools.org. Installed and managed through a download-able user-friendly GUI, PEACE can process large data sets of transcript fragments of length 50 bases or greater, grouping the fragments by gene associations with a sensitivity compatible to leading clustering tools. Once clustered, the user can employ the GUI's analysis functions, facilitating the easy collection of statistics and allowing them to single out specific clusters for more comprehensive study or assembly. Using a novel *minimum spanning tree* based clustering method, PEACE is the equal of other leading tools in the literature in quality with an interface making it accessible to any user. It produces results of quality virtually identical to those of the WCD tool when applied to Sanger sequences, significantly improved results over WCD and TGICL when applied the products of Next Generation Sequencing Technology, and significantly improved results over Cap3 in both cases. In short, PEACE provides an intuitive graphical user interface and a feature-rich, parallel clustering engine that proves to be a valuable addition to the leading cDNA clustering tools.

INTRODUCTION

Understanding an organism's transcriptome, the set of (spliced) transcripts expressed by genes of the organism, is a vital step in understanding the full functional and organizational role of the genome in the life cycle of any eukaryote. Studying the transcriptome has led to gene discovery, provided information on splice variants, and helped shed light on the biological processes both controlling and controlled by the genome (1). However, to access those transcripts, we must deal with the fragmented data produced by both Next Generation and traditional Sanger sequencing technology.

In the past, access to a transcriptome sequence was primarily through the use of Expressed Sequence Tags (ESTs), single-pass cDNA sequences derived from transcribed mRNAs and sequenced by Sanger Sequencing technology. More recently, Next Generation Sequencing (NGS) technology has begun to rapidly replace Sanger Sequencing. For example, ESTs now being added to the GenBank dbEST are increasingly the product of NGS technologies such as 454 pyrosequencing, which enables the sequencing of novel and rare transcripts at a considerably higher (2, 3). From a computational perspective, this is a mixed blessing: while NGS provides immense quantities of new information, it also provides immensely larger data sets – and thus a need for fast, efficient analysis algorithms.

Given a set of transcript fragments sampled from across the genome, a necessary first step of the set's analysis is clustering: separating the fragments according to the transcript from which they were derived. Frequently performed implicitly by assembly tools, clustering the data as a "pre-assembly" step has a number of advantages. Most significantly: performing this step will allow the application of the assembly tool to individual clusters – saving significant amounts of time (4).

Clustering is a computationally challenging problem; the runtime and memory requirements to cluster on the basis of pair-wise sequence alignments make such an approach infeasible in practice. To deal with this, PEACE combines our own version of the d^2 alignment-free sequence distance function (5) and the concept of a *minimum spanning tree* (6) to quickly and accurately find clusters of ESTs expressed from the same gene without reference to a sequenced genome. Compared against clustering tool in the literature (4, 7, 8, 9, 10, 11, 12, 13, 14, 15), PEACE proves to be of equivalent quality to the WCD and TGICLtools (4, 15), and more sensitive and robust than other tools. From the user perspective, o tool in the literature can match PEACEfor ease of instillation, use, or the post-clustering analysis tools PEACEprovides.

In short, PEACE is a computational tool for the *ab initio* clustering of transcript fragments by gene association, applicable to both NGS and traditional Sanger Sequencing technologies. Available through the www.peace-tools.org website, the

*to whom correspondence should be addressed

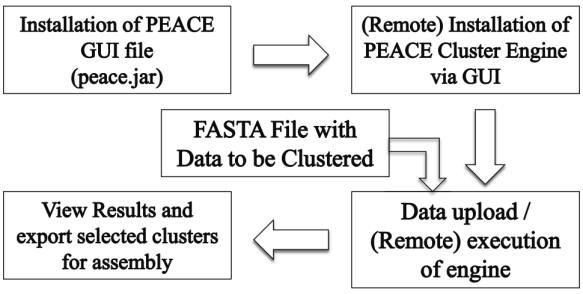

Figure 1. Overview of the procedure for clustering and analysis using PEACE.

PEACE GUI allows the user to both easily install (locally or remotely) and run the clustering engine, as well as enabling transparent parallel processing and providing various tools for result analysis.

PEACE: INSTALLATION AND USE

The PEACE GUI can be launched by through the Jar file available on the PEACE website (www.peace-tools.org) on any machine running the standard Java Virtual Machine (JVM). Once running, the user can employ the GUI to install the clustering engine and perform a clustering of a data file in FASTA format, view an initial analysis of the clusters, and produce files containing subsets of the clusters as input to assembly tools such as Cap3 (9). A typical (first) use of PEACE must be performed in the following manner (see Figure 1):

Tool Installation: (First use only.) To install the PEACE clustering engine onto a local or remote machine, the user selects from within the GUI the appropriate menu tab (Figure 2(a)), which then starts an install wizard that will prompt for the appropriate information. Figure 2(b) illustrates the request for server information; the user has chosen to install the PEACE computational tool on a remote machine and is providing the necessary connection information. Server information is persistent between GUI sessions, giving the user access to PEACE on the target machine as needed.

Job Processing: After importing the target sequence file into the GUI, the user starts a new job by following the wizard menus. Figure 2(c) illustrates the process of specifying the number of processors available (if running on a machine supporting the OpenMPI protocol – which will be determined during job installation). Once executed, the GUI will manage the job thread, alert the user when the job is completed (or when the user next runs the GUI after completion), and copy the final results back to the local machine if necessary.

Result Analysis: Once the resulting clusters have been computed, the user has several options for analysis:

- **Export:** The user can export the contents of one or more clusters into a FASTA format file, obtaining a subset of the original target file containing the sequences corresponding to the selected clusters ready for processing by an assembly tool (e.g. Cap3 (9)).
- **View Clustering**: The user may view a list of clusters, expanding selected clusters to a list of all individual sequences (illustrated in Figure 2(d)).
- **Classified Summary Graph:** The user may view a distribution of cluster sizes. Further, the user can set up a *classifier*, associating certain patterns with specific colors. These patterns ware matched against the fragment header information from the original FASTA file, allowing the overlay of a colored cluster size distributions. For example, if the sequence names contain unique string patterns denoting different cDNA libraries, the classifier can help the user to determine and visualize the differential expression profiles of different libraries for a given cluster. The method of setting up these classifiers, and the resulting histogram, is illustrated in Figure 2(e).

Extensive documentation for the tool has been posted on the PEACE website, as well as links to several tutorial videos demonstrating PEACE use and capabilities.

METHODS

The clustering performed by PEACE is based on the use of minimum spanning trees (MSTs), known to be an effective approach for narrow band single linkage clustering (16, 17). Using a graph structure to model the fragment relationships and the d^2 distance measure to assign edge weights (5), we can employ Prim's algorithm (6) to efficiently calculate an MST from which we can infer a high-quality clustering solution.

The d^2 distance measure used to assign edge weights is an alignment-free measurement of sequence distance that can be calculated significantly faster than a Smith-Waterman alignment (5). d^2 works by comparing the frequency of words (strings of a fixed length) appearing in a limited region of each string. Fragments overlapping by a sufficient length will share neighborhoods of enough similarity to ensure a small distance even in the presence of a moderate number of base errors. In practice we employ our own variation of d^2, the *two-pass* d^2 *algorithm*, which heuristically searches for a neighborhood of maximum similarity and then finds the d^2 score based on that neighborhood (see Supplementary Materials for details).

Fragment input is modeled as a weighted, undirected graph: the fragments are represented as nodes, with d^2 sequence distances assigned to the connecting edges as weights. Conceptually, we want to remove each edge exceeding a threshold score from the complete graph and define our partitions by the remaining connected components. An edge with a large weight connects fragments which are likely unrelated; once such edges are removed the components define a series of overlaps. Those fragments that can still be connected by some path correspond to the same gene. However, such an approach requires the calculation of all edge weights. That task is infeasible both in terms of runtime and memory usage for the data set sizes we expect to process.

PEACE approaches the problem by generating a minimum spanning tree of the described graph, then removing edges exceeding our threshold. By using Prim's algorithm we are able to calculate edge weights on-the-fly (reducing memory requirements) and can skip the calculation of a majority of edge distances using the u/v and t/v filtering heuristics employed in WCD (4). These heuristics allow us to quickly dismiss many of the edges as too large without the need

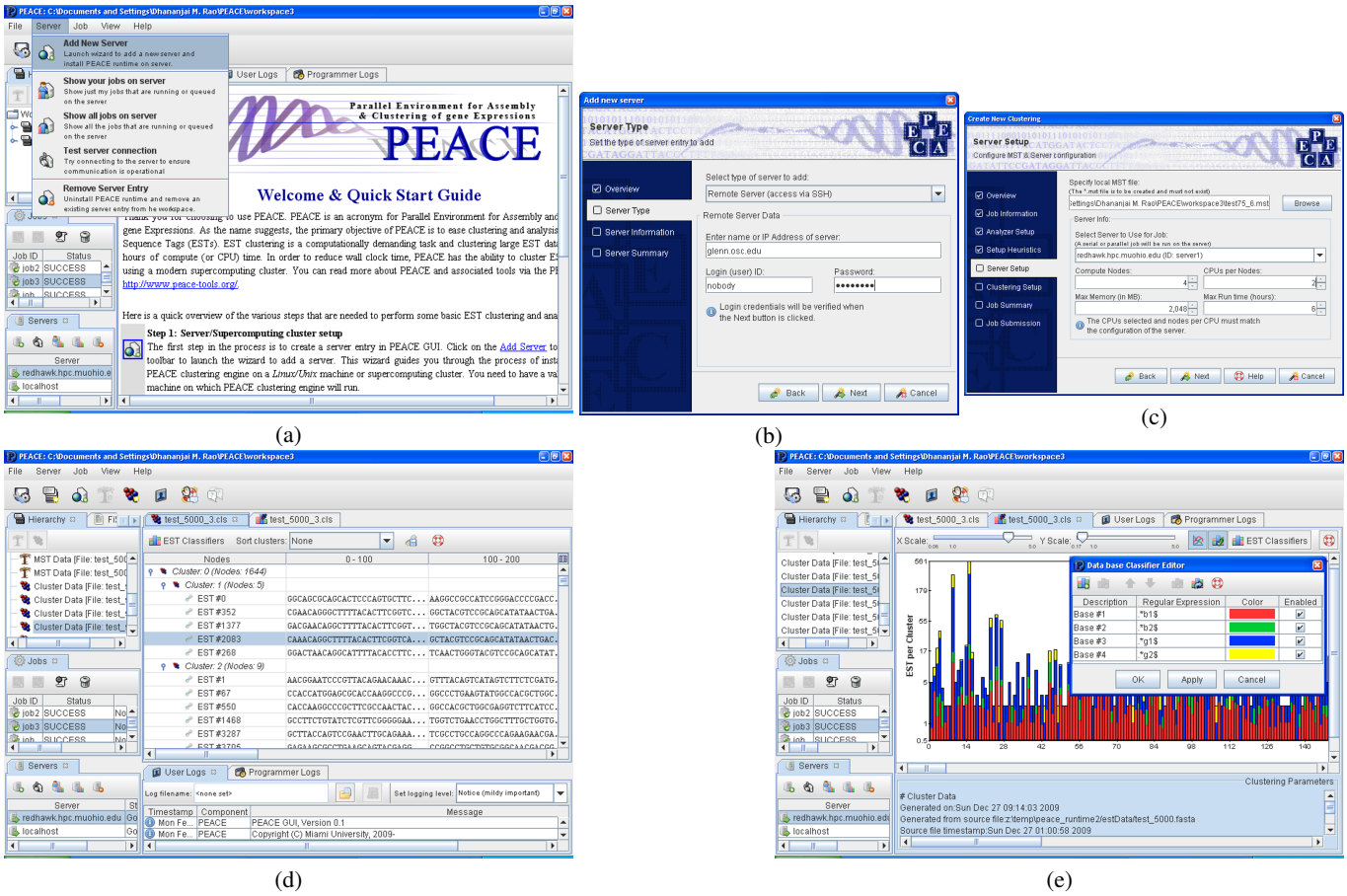

Figure 2. Screenshots of the PEACE GUI during execution, including (a) GUI Welcome and server installation menu; (b) setup wizard for installing the computational tool on a remote server; (c) execution wizard for starting a selected job to be executed in parallel mode; (d) basic cluster output; and (e) histogram view of cluster results and classifier editor for setting up differential expression profiles.

to apply the full d^2 algorithm (see Sections A.3 of the Supplementary Materials for more details).

RESULTS

PEACE has been tested on both simulated and real data from NGS and Sanger Sequencing technologies, comparing results against those produced by the WCD clustering tool (4) and the Cap3 assembly tool (9) (the latter of which implicitly calculates a clustering in the process of assembly). For our simulation tests we used the **ESTSim** tool to generate simulated Sanger sequenced transcript fragments, and the MetaSim tool to generate simulated short read sequences from 454 and Illumina technologies (18, 19). (See Supplementary Materials, Section D.1, for details on the sequence size and error models.) Fragments were generated from the list of 100 zebra fish genes used in the WCD testing (4). Tool parameters were taken to match, as closely as possible, those used in the WCD study (see Supplementary Materials).

The most important method of quality assessment is *sensitivity* (the fraction of fragment pairs from the same gene that were correctly clustered together). We also look at the *Jaccard Index* (which balances sensitivity with the number of false positives), *Type 1 error* (the fraction of genes that were divided between clusters), and *Type 2 error* (the fraction of clusters containing two or more genes) (4, 20). In Figure 3 we plot these four tests as a function of error rate, and observe the almost identical results between PEACE and WCD. In Figure 4 we plot the runtime for PEACE and WCD, again observing almost identical results when run sequentially – but significantly faster runtime for PEACE on multiple processor when holding the EST/processor ratio constant (ranging from a 65% improvement for two processors to a 17% improvement for 12 processors).

In the course of our simulations we also investigated the memory footprint – of particular concern in the large data sets commonly encountered. In short, we find that, as with WCD, the required memory for an execution is linear in the size of the data set – requiring slightly more memory than WCD and considerably less than Cap3. We present relevant results in Section D.4 and Figure S3 of the Supplementary Materials.

We note, as a point of interest, that we can significantly improve the quality of PEACE simulation results through the increase of the threshold value (see Supplementary Materials, Section D.5) – achieving a significant improvement in PEACE sensitivity without an adverse effect on the Jaccard Index. However, the improvement does not carry over to the application of real data (Supplementary Materials, Section E.2), where we observe a significant increase in the incorrect merging of clusters. While this might be acceptable to a user

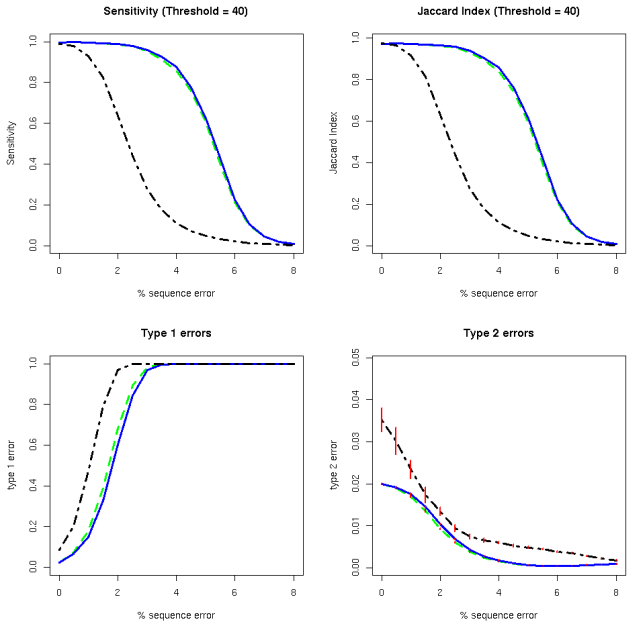

Figure 3. Comparisons of Sensitivity, Jaccard Index, Type 1 error and Type 1 error, based on the average over 30 simulated Sanger Sequence ESTs sets derived from 100 zebra fish genes (see Supplementary Materials, Section D, for more details). Blue/Solid = PEACE, Green/Dash = WCD (version 0.5.1), Black/Dot-Dash = Cap3; vertical tics = 95% confidence intervals on estimates. Intervals are not presented for Type 1 error due to the effective lack of variance.

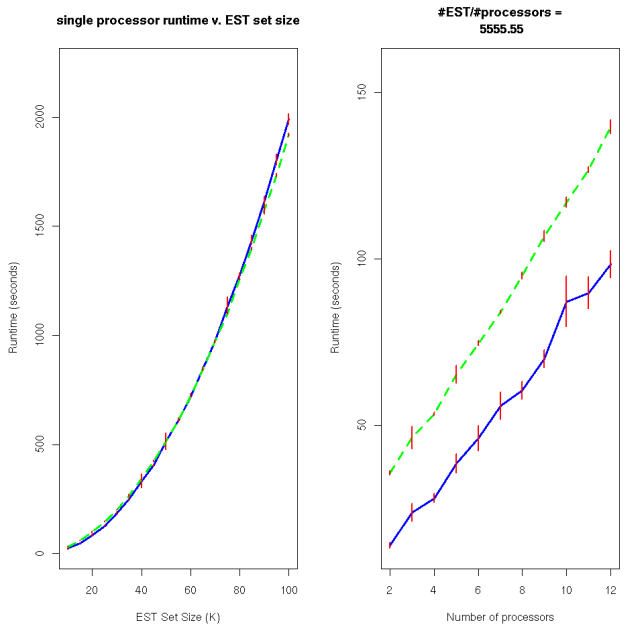

Figure 4. Comparisons of Runtime: On left, we compare the sequential runtime of PEACE (blue) and WCD (green) on simulate sets ranging in size from 10K sequences to 100K sequences. (Cap3 runtime is not reported, as the time spent on clustering cannot be differentiated from the time spent on assembly.) On right, a comparison when run in parallel, ranging the number of processors from 2 to 12 while holding the EST/Processor ratio steady at the constant 8888. All values represent the average of 30 runs; vertical tics = 95% confidence intervals on estimates. All runs were done on a 3.0 GHz Intel Xeon EM64T CPU with a 2 MB cache and a 800 MHz front side bus, model number Xeon LV 3.0 (2005).

planning to employ an assembly tool capable of breaking up the clusters (but unable to re-merge a split cluster), for the purposes of clustering assessment it makes sense base our analysis (and set out default values) at the lower threshold.

While PEACE clearly matches WCD on long-read sequence, comparisons against TGICL (15) are more complicated. In short, we find that when parameters are picked appropriately, TGICL is more sensitive at lower error rate – but PEACE is more robust to error, in all cases has an improved Type 2 error rates and is better able to distinguish duplicated genes. (See the Supplementary Materials for a more detailed analysis.)

In applying the tools to real data Sanger data, we used the Human Benchmark Data Set used to test EasyCluster (14), and the A076941 *Arabidopsis thaliana* data set used to test wcd (4, 21) (see Table 1). We notice essentially identical results for PEACE and wcd in quality, both significantly better than Cap3 in Sensitivity and Type 1 error rate, while slightly worse in the Jaccard Index and Type 2 error. In runtime we see some inconsistency, with PEACE showing a 60% runtime improvement over WCD in the first data set, but requiring 20% more time than WCD in the Arabidopsis data set. (We cannot make a reasonable runtime comparison to Cap3, as the runtime of that tool reflects both clustering and assembly.) In Table S1 of the Supplementary Materials we present runtimes of several more sets, observing that while PEACE appears to be significantly faster on the smaller sets, WCD does overtake it for larger sets.

We tested the three tools on short-read data using the MetaSim tool of *Richter et al.* (19). Encoded into MetaSim are sequence generation and error models corresponding to several technologies, including the 454 short-read technology (producing reads ranging in size from 200bp to 370bp with a mean of 250bp and a standard deviation of 17bp), and the Illumina short-read technology (producing reads of exactly 62bp). Experiments were run using two sets of WCD parameters (the default parameters, and setting window size to 75 and threshold to 75 – thus matching the equivalent PEACE parameters), and using default values of Cap3. Only with PEACE were we able to get reasonable results, seeing a sensitivity of 0.871 for the 454 reads and 0.583 or the shorter Illumina reads, a Jaccard Index of 0.809 and 0.372 respectively, and other comparably figures. (See Table S3 in the Supplementary Materials.) In short, PEACEperforms quite well for 454 reads and provides some useful information for Illumina reads, while WCD and Cap3 are unable to produce working results.

CONCLUSIONS

Here we have presented PEACE, a stand alone tool for the high-throughput clustering of transcript fragments capable of dealing with sequences as short as 50 bases. PEACE is open-source and managed through a user friendly GUI that enables both local and remote installation and execution in sequential or parallel mode. Based on a novel minimum spanning tree based algorithm for the clustering of the fragments by gene association, PEACE shows significant improvement in sensitivity over the competing WCD tool (4) when applied to

		Sensitivity	Jaccard	Type 1 error	Type 2 error	Number of Clusters	Number of Singletons	Single processor runtime (s)
EasyCluster Human Benchmark (111 Genes)	PEACE	0.998	0.672	0.153	0.042	118	21	293
	WCD	0.998	0.672	0.144	0.044	113	16	804
	Cap3	0.657	0.643	1.000	0.001	2269	1827	NA
WCD A076941 Benchmark (13240 genes)	PEACE	0.932	0.475	0.351	0.027	18825	8951	1166
	WCD	0.933	0.476	0.350	0.027	18787	8553	966
	Cap3	0.826	0.802	0.486	0.014	25042	14916	NA

Table 1. Comparisons of runs on the EasyCluster human Benchmark Data Set and the WCD A076941 Arabidopsis thaliana data set using the standard quality measurements.

NGS reads, matches WCD when applied to Sanger sequencing output, and shows an order of magnitude in improvement over the clustering performed in the course of assembly by the Cap3 tool (9).

As a clustering tool based on sequence distance, PEACE faces certain inherent limitations. For example, PEACE cannot handle duplicate genes; like WCD, it is unable to separate clusters corresponding to genes with a greater than 88% similarity. Similarly, other natural biological effects (e.g. the trans-splicing of transcripts), effects from poorly cleaned transcript data (e.g. the failure to remove sequencing adapters or post-transcriptional poly(A)/(T) tails), and the presence of low-complexity repeats can cause similar effects in these clustering tools. These problems can be handled through the application of the assembler, and the ability to apply any assembler to small clusters (as opposed to the data set as a whole) results in a significant reduction in overall assembly time.

Peace can be downloaded from www.peace-tools.org, where we are committed to keep maintaining and improving the tool in the future. Meanwhile, we are developing our own MST-based assembly tool that can seamlessly integrate with PEACE. The underlying modular design of PEACE offers users many possibilities to expand and incorporate the MST algorithm for other bioinformatics application.

ACKNOWLEDGEMENTS

Dr. Karro was funded under a PhRMA Foundation Informatics Research Starters Grant while conducting this research. We would also like to acknowledge Iddo Friedberg, David Woods, Elizabeth Bikun, Jens Mueller and David Scoville at Miami University for their help with this project.

REFERENCES

1. S Nagaraj, R Gasser, and S Ranganathan. A hitchhiker's guide to expressed sequence tag (est) analysis. *Brief Bioinformatics*, Jan 2007.
2. Foo Cheung, Brian J Haas, Susanne M D Goldberg, Gregory D May, Yongli Xiao, and Christopher D Town. Sequencing medicago truncatula expressed sequenced tags using 454 life sciences technology. *BMC Genomics*, 7:272, Jan 2006. [PubMed:17062153] [PubMed Central:PMC1635983] [doi:10.1186/1471-2164-7-272].
3. Scott J Emrich, W Brad Barbazuk, Li Li, and Patrick S Schnable. Gene discovery and annotation using lcm-454 transcriptome sequencing. *Genome Res*, 17(1):69–73, Jan 2007. [PubMed:17095711] [PubMed Central:PMC1716268] [doi:10.1101/gr.5145806].
4. Scott Hazelhurst, Winston Hide, Zsuzsanna Lipták, Ramon Nogueira, and Richard Starfield. An overview of the wcd est clustering tool. *Bioinformatics*, 24(13):1542–6, Jul 2008. [PubMed:18480101] [PubMed Central:PMC2718666] [doi:10.1093/bioinformatics/btn203].
5. Winston Hide, John Burke, and Daniel B Davison. Biological evaluation of d2, an algorithm for high-performance sequence comparison. *Journal of Computational Biology*, 1(3):199–215, Aug 1994. [PubMed:8790465].
6. R Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, Jan 1957.
7. J Burke, D Davison, and W Hide. d2_cluster: a validated method for clustering est and full-length cdnasequences. *Genome Res*, 9(11):1135–42, Nov 1999. [PubMed:10568753] [PubMed Central:PMC310833].
8. G. Slater. *Algorithms for analysis of exptressed sequence tags*. PhD thesis, University of Cambridge, Cambridge, 2000.
9. X Huang and A Madan. Cap3: A dna sequence assembly program. *Genome Res*, 9(9):868–877, 1999. [PubMed:10508846] [PubMed Central:PMC310812].
10. J Parkinson, D Guiliano, and M Blaxter. Making sense of est sequences by clobbing them. *BMC Bioinformatics*, Jan 2002. [PubMed:12398795] [PubMed Central:PMC137596].
11. Anantharaman Kalyanaraman, Srinivas Aluru, Suresh Kothari, and Volker Brendel. Efficient clustering of large est data sets on parallel computers. *Nucleic Acids Res*, 31(11):2963–74, Jun 2003. [PubMed:12771222] [PubMed Central:PMC156714].
12. Ketil Malde, Eivind Coward, and Inge Jonassen. Fast sequence clustering using a suffix array algorithm. *Bioinformatics*, 19(10):1221–6, Jul 2003. [PubMed:12835265].
13. Andrey Ptitsyn and Winston Hide. Clu: a new algorithm for est clustering. *BMC Bioinformatics*, 6 Suppl 2:S3, Jul 2005. [PubMed:16026600] [PubMed Central:PMC1637039] [doi:10.1186/1471-2105-6-S2-S3].
14. E Picardi, F Mignone, and G Pesole. Easycluster: a fast and efficient gene-oriented clustering tool for large-scale transcriptome .... *BMC Bioinformatics*, Jan 2009.
15. G. Pertea, X. Huang, F. Liang, V. Antonescu, R. Sultana, S. Karamycheva, Y. Lee, J. White, F. Cheung, B. Parvizi, J. Tsai, and J. Quackenbush. Tigr gene indices clustering tools (tgicl): a software system for fast clustering of large est datasets. *Bioinformatics (Oxford, England)*, 19(5):651–652, Mar 22 2003. [PubMed:12651724].
16. A Jain, M Murty, and P Flynn. Data clustering: a review. *Computing Surveys (CSUR*, 31(3), Sep 1999.
17. Xiu-Feng Wan, Mufit Ozden, and Guohui Lin. Ubiquitous reassortments in influenza a viruses. *Journal of bioinformatics and computational biology*, 6(5):981–99, Oct 2008. [PubMed:18942162].
18. S Hazelhurst and A Bergheim. Estsim: A tool for creating benchmarks for est clustering algorithms. *Dept. of Computer Science, Univ. of Witwatersrand (South Africa), Tech. Rep. CS-2003-1*, 2003.
19. Daniel C Richter, Felix Ott, Alexander F Auch, Ramona Schmid, and Daniel H Huson. Metasim: a sequencing simulator for genomics and metagenomics. *PLoS ONE*, 3(10):e3373, Jan 2008. [PubMed:18493042] [PubMed Central:PMC2556396].
20. Ji-Ping Z Wang, Bruce G Lindsay, James Leebens-Mack, Liying Cui, Kerr Wall, Webb C Miller, and Claude W dePamphilis. Est clustering error evaluation and correction. *Bioinformatics*, 20(17):2973–84, Nov 2004. [PubMed:15189818] [doi:10.1093/bioinformatics/bth342].
21. Scott Hazelhurst. Algorithms for clustering expressed sequence tags: the wcd tool. *South African Computer Journal*, 40:51–62, May 2008.