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

ABSTRACT

We present PEACE, a stand alone tool for the high-throughput *ab initio* clustering of transcript fragment sequences produced by Next Generation or Sanger Sequencing technologies, freely available from www.peace-tools.org. Installed and run through an easily downloaded GUI, PEACE can process large data sets of transcript fragments of length 50 bases or greater, grouping the fragments by gene association and achieving a greater sensitivity (without sacrificing runtime) than the competing clustering tools. Once clustered, the user can employ the GUI to collect statistics and single out specific clusters for more comprehensive study. PEACE, when used as a "pre-assembly" tool, enables any assembly tool to work with individual small clusters of closely related sequences. For a 5% sequencing error rate, we see a 52% improvement in sensitivity over the leading cluster tool, WCD, and 184% improvement in sensitivity over the assembly tool Cap3. When applied to a 361 Mb set of Arabidopsis transcriptome fragments, we see a 40% speedup over WCD.

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, allowing for more comprehensive coverage of the transcriptome. 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 rate of coverage than Sanger Sequencing (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 faster, more efficient analysis algorithms.

Given a set of transcript fragments sampled from across the genome, a necessary first step of the set's analysis is that of 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).

However, clustering is a computationally challenging problem. Even with the smaller number of ESTs produced using Sanger Sequencing, the runtime and memory requirements to cluster on the basis of pair-wise sequence alignments make such an approach infeasible in practice. The much larger data-set size produced by NGS technologies exacerbates this problem. 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 WCD, the leading clustering tool in the literature (4), as well as other tools designed for the same purpose (4, 7, 8, 9, 10, 11, 12, 13, 14), PEACE proves to be both more sensitive and more robust to sequencing error without sacrificing runtime. Nor are any of these tools designed for the ease of installation and use that 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 PEACE GUI allows the user to both easily install (locally or remotely) and run the clustering engine, as well as enabling

*to whom correspondence should be addressed

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

transparent parallel processing and providing various tools for result analysis.

PEACE: INSTALLATION AND USE

The PEACE GUI can be launched by downloading and executing the Jar file available on the PEACE website (www.peace-tools.org) to 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 (15, 16). 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 to

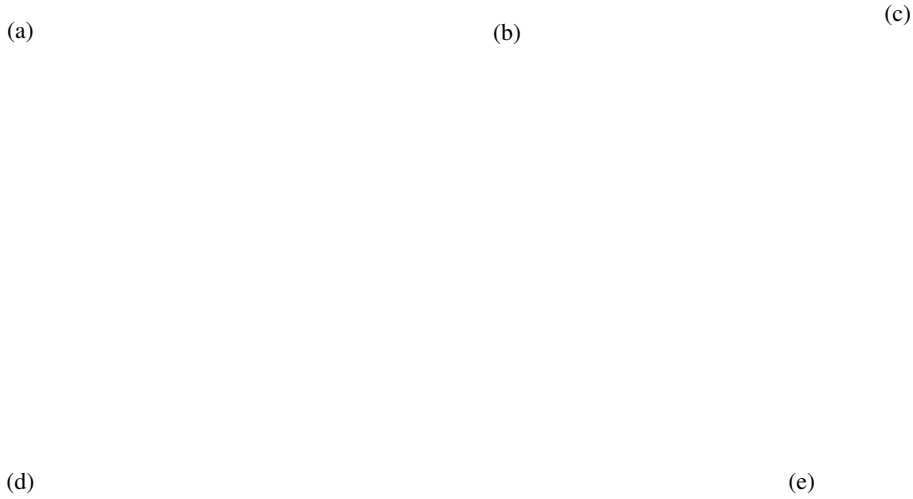

Figure 2. Screenshots of the PEACE GUI during execution, including (a) GUI Welcome and server installation menu; (b) setup wizard for installing the computational tool on a remote server; (c) execution wizard for starting a selected job to be executed in parallel mode; (d) basic cluster output; and (e)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

apply the full d^2 algorithm (see Sections A and B 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 (17) to generate simulated transcript fragments of varying length under different models of error (Supplementary Materials, Section C.1), generating the fragments 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). Our primary methods of quality assessment were *sensitivity* (the fraction of fragment pairs from the same gene that were correctly clustered together) and *Type 1 error* (the fraction of genes that were divided between clusters) (4, 18). Specificity was not an issue in simulated tests, as both PEACE and WCD were completely successful in separating unrelated ESTs (save when dealing with recently duplicated genes). Other measurements are also discussed in the supplementary materials. In Figure 3 (left), reflecting tests done on simulated Sanger EST sets, we see a significant improvement of PEACE (blue) over WCD (green) and Cap3 (black) in its sensitivity to sequencing errors, while in Figure 3 (right) we see a comparable improvement in Type 1 error. In Figure 4 we look at the number of singleton clusters (fragments not joined to any cluster), which should not occur in our simulated sets. We again see significant improvement in PEACE. The sequential runtime of PEACE is slightly improved over that of WCD (see Supplementary Materials, Figure S4).

In applying the tools to real data, we started with a set of approximately $190{,}000$ clean Sanger ESTs derived from the *Chlamydomonas reinhardtii* genome (19). To compute sensitivity we used the *gmap* tool (20) to map the fragment set to the genome, taking this as our reference clustering. We see slight improvements in PEACE over WCD in both sensitivity and Type 1 error (both significantly outperforming Cap3). Using the mouse data set used in Hazelhurst *et al.* (4) PEACE again shows slight improvements in sensitivity and Type 1 error rates, with an 18% speedup for PEACE (see Supplementary Materials, Section D, for more details).

We were unable to run WCD on short-read sequences, as the distributed implementation is unable to handle such data without an adjustment to the code. Hence, while PEACE was run on both simulated and real NGS fragment sets, there was no cluster tool to which the results could be

Figure 3. Comparisons of Sensitivity 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 C, for more details). Blue/Solid = PEACE, Green/Dash = WCD, Black/Dot-Dash = Cap3; vertical tics = 95% confidence intervals on estimates. Intervals are not presented for Type 1 error due to the extremely small variance.

Figure 4. Average number of fragments flagged as singletons by each tool when run on the simulated sequences; correct answer in all cases is zero. Blue/Solid = PEACE, Green/Dash = WCD, Black/Dot-Dash = Cap3.

compared. PEACE did well, achieving a sensitivity of 0.75 when applied to a simulated short-read set subjected to a sequencing error rate of 2% (with comparable *Jaccard Index* values reflecting both sensitivity *and* specificity (4, 11)). Runtime was approximately 17.6 minutes to process the 7.9 Mb of data when run in sequential mode on a 3.2 GHz Intel Xeon CPU. When applied to a 64 Mb set of 165,490 *Chlamydomonas reinhardtii* transcript fragments sequenced by 454 pyrosequencing technology (ranging in size from 50 bases to several hundred), PEACE completed the job in 49 minutes using 30 of the Xeon processors.

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, available at www.peace-tools.org, 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 algorithm for the clustering of the fragments by gene association, PEACE shows significant improvement in sensitivity, without sacrificing runtime, over the competing WCD tool (4), 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. The problems can be handled through the application of the assembler, and the ability to apply any assembler to small cluster (as opposed to the data set as a whole) results in a significant reduction in overall assembly time.

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. A Jain, M Murty, and P Flynn. Data clustering: a review. *Computing Surveys (CSUR*, 31(3), Sep 1999.
16. 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].
17. 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.
18. 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].
19. Chun Liang, Yuansheng Liu, Lin Liu, Adam C Davis, Yingjia Shen, and Qingshun Quinn Li. Expressed sequence tags with cdna termini: previously overlooked resources for gene annotation and transcriptome exploration in chlamydomonas reinhardtii. *Genetics*, 179(1):83–93, May 2008. [PubMed:18493042] [PubMed Central:PMC2390652] [doi:10.1534/genetics.107.085605].
20. Thomas D Wu and Colin K Watanabe. Gmap: a genomic mapping and alignment program for mrna and est sequences. *Bioinformatics*, 21(9):1859–75, May 2005. [PubMed:15728110] [doi:10.1093/bioinformatics/bti310].