PEACE: P̲arallel E̲nvironment for A̲ssembly and C̲lustering of Gene E̲xpression

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

ABSTRACT

We present PEACE, a standalone tool for the high-throughput *ab initio* clustering of transcript fragment sequences produced by Next Generation or Sanger Sequencing technologies. Installed and run through an easily downloaded GUI, PEACE can process large data sets and group the fragments by gene association, achieving a greater sensitivity (without sacrificing runtime) than the competing clustering tool WCD. Through the GUI the user can collect cluster statistics and examine specific clusters for more comprehensive study. PEACE, used as a "pre-assembly" tool, enables any assembly tool to work with individual small clusters of closely related sequences, resulting in a signficant: increases in sensitivity, robustness to sequenceing error, and reduction in overall runtime.

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 dictated by the genome (1). But to access to those transcripts, we must deal with the fragmented data produced by both Next Generation and traditional Sanger sequencing technology.

Until recently, 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 technology (NGS) is rapidly replacing Sanger Sequencing, allowing for much more comprehensive coverage of the transcriptome. For example, ESTs in 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 did 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, requiring fast computational techniques if we are to process them.

Given a set of transcript fragments sampled from across the genome, a necessary first step in the 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 significant among these: performing this step will allow the application of the assembly tool to individual clusters – saving significant amounts of time due to the reduced input size (4). But clustering is a computationally challenging problem. Even with the smaller number of ESTs produced using Sanger sequencing, the runtime and memory requirements to cluster based on pairwise sequence alignments make such an approach infeasible. The much larger data-set size produced by NGS technologies exacerbates this problem.

Here we present PEACE, a tool for the *ab initio* clustering of transcript fragments by gene, 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 computational tool, as well as enabling transparent parallel processing and providing various tools for result analysis. Using a *minimum spanning tree* (MST) based clustering strategy (5, 6) and the d^2 sequence distance function (7), PEACE generates the MSTs using a tailored version of Prim's algorithm (8). The result is an easy-to-use and clustering tool that , without any sacrifice in runtime, is considerably more sensitive and robust to sequencing errors than other clusters tools in the literature (4, 9, 10, 11, 12, 13, 14, 15, 16).

PEACE: INSTALLATION AND USE

The PEACE GUI, available from the PEACE website (www.peace-tools.org), handles the details of both installation and job management and provides tools for the initial analysis of the results. By downloading and executing the `peace.jar` GUI Java program on a Linux, Windows or OS-X machine running Java 1.6+, the user is presented with the

---

*to whom correspondence should be addressed

PEACE management tool. Through this the user can install the computational tool and perform a clustering of a data file in the FASTA format, view an initial analysis of the clusters and produce files containing subsets of the clusters for assembly by a tool such as CAP3 (9). A typical (first) use of PEACE would generally require the the following steps.

Tool Installation: Installing the GUI requires only that the Java file be downloaded from the website. To install the PEACE tool onto a local or remote machine, the user will select from within the GUI the appropriate menu tab (Figure 1(a)), which then starts the install wizard that will prompt for the appropriate information. Figure 1(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 can start a new job and follow the wizard menus. Figure 1(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 1(d)).

- **Classified Summary Graph:** The user may view a distribution of cluster sizes. Further, the user may set up a *classifier*, associating certain patterns with specific colors. These patterns will be 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 a these classifiers, and the resulting histogram, is illustrated in Figure 1(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 (5, 6). We use the d^2 distance measure (7) to represent edge weights. Prim's algorithm (8) then allows us 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 (7). 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.

We can model the fragment input 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 of the graph. An edge with a large weight connects fragments which are likely unrelated; once such edges are removed, the components define a series of overlaps. Hence 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. Such a task is infeasible both in terms of runtime and memory usage for the data set sizes we expect to process.

We approach 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 we 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 apply the full d^2 algorithm (see Sections A and B of the Supplementary Materials for more details).

RESULTS

We have tested PEACE on both simulated and real data, comparing results against those produced by the WCD clustering tool (4) and the CAP3 assembly tool (9) (which calculates a cluster as an implicit, intermediate step). For our simulation tests we used the **ESTSim** tool (17) to generate simulated transcript of varying length fragments 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); other measurements are also discussed in the supplementary materials. In Figure 2(left) we see a significant improvement of PEACE (blue) over WCD (green) and CAP3 (black) in its sensitivity to sequencing errors, while in Figure 2(right) we see a comparable improvement in Type 1 error. In Figure 3 we look at the number of singleton clusters (fragments not joined to any cluster), which will 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 dataset 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 , for more details).

CONCLUSIONS

Here we have presented PEACE, a stand-alone tool for the high-throughput clustering of transcript fragments, 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 preformed 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, naturally biological effects such as the trans-splicing of transcripts, or effects from poorly cleaned transcript data (e.g. the failure to remove sequencing adapters), post-transcriptional poly(A)/(T) tails, and 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 while) results in a significant reduction in overall assembly time.

ACKNOWLEDGEMENTS

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. A Jain, M Murty, and P Flynn. Data clustering: a review. *Computing Surveys (CSUR)*, 31(3), Sep 1999.
6. 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].
7. 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].
8. R Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, Jan 1957.
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 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].
11. G. Slater. *Algorithms for analysis of exptressed sequence tags*. PhD thesis, University of Cambridge, Cambridge, 2000.
12. J Parkinson, D Guiliano, and M Blaxter. Making sense of est sequences by clobbing them. *BMC Bioinformatics*, Jan 2002. [PubMed:12398795] [PubMed Central:PMC137596].
13. 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].
14. Ketil Malde, Eivind Coward, and Inge Jonassen. Fast sequence clustering using a suffix array algorithm. *Bioinformatics*, 19(10):1221–6, Jul 2003. [PubMed:12835265].
15. 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].
16. E Picardi, F Mignone, and G Pesole. Easycluster: a fast and efficient gene-oriented clustering tool for large-scale transcriptome …. *BMC Bioinformatics*, Jan 2009.
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].
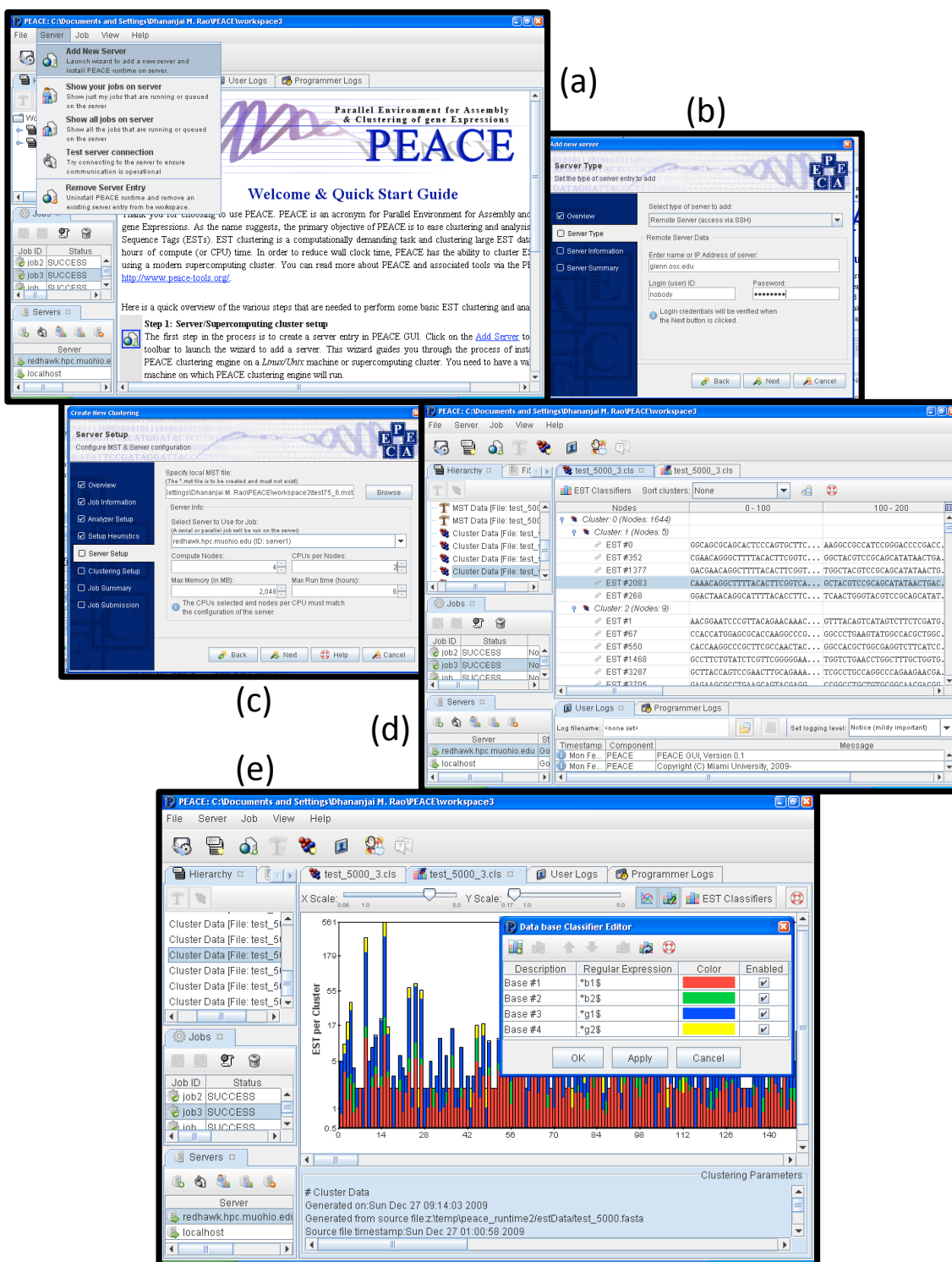
Figure 1. Screenshots of the PEACE GUI during execution, including (a) GUI Welcome and server installation menu; (b) setup wizard for installing the computational tool on a remote server; (c) execution wizard for starting a selected job to be executed in parallel mode; (d) basic cluster output; and (e) histogram view of cluster results and classifier editor for setting up differential expression profiles.
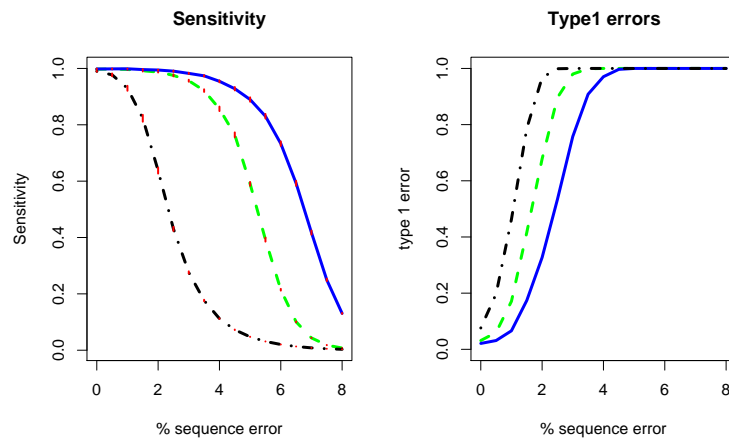
Figure 2. Comparisons of Sensitivity and Type 1 error, based on the average over 30 simulated fragment sets derived from 100 zebra fish genes (see Supplementary Materials, Section , for more details). Blue/Solid = PEACE, Green/Dash = WCD, Black/Dot-Dash = CAP3; vertical tics = 95% confidence intervals on estimates. Intervals are not presented for Type 1 error due to the extremely small variance.
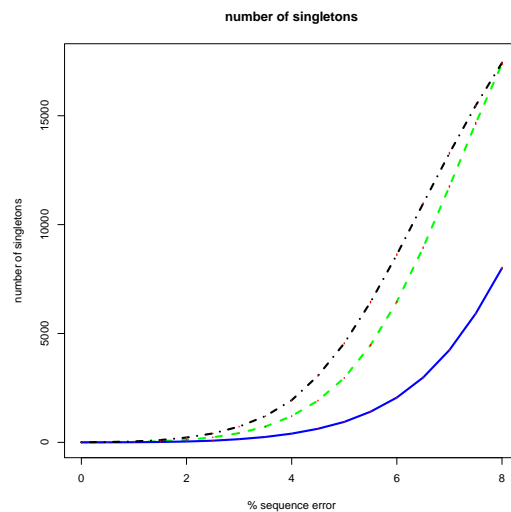

Figure 3. Average number of fragments flagged as singletons by each tool when run on the simulated sequences; correct answer in all cases is zero. Blue/Solid = PEACE, Green/Dash = WCD, Black/Dot-Dash = CAP3.