Published online 3 June 2010 *Nucleic Acids Research, 2010, Vol. 38, Web Server issue* ***W737–W742***
doi:10.1093/nar/gkq470

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 and
[3]Department of Microbiology, Miami University, Oxford, Ohio 45056, USA

Received February 14, 2010; Revised May 11, 2010; Accepted May 12, 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 downloadable user-friendly graphical user interface (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 comparable 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 leading tools in the literature, 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 to the products of Next Generation Sequencing Technology and significantly improved results over Cap3 in both cases. In short, PEACE provides an intuitive GUI 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 equencing. 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 speed (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 run time and memory requirements to cluster on the basis of pairwise 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 (MST) (6) to quickly and accurately find clusters of ESTs expressed from the same gene without reference to a sequenced genome. Compared against clustering tools in the literature (4,7–15), PEACE produces results of quality competitive with the WCD and TGICL tools (4,15), and more sensitive and robust than

*To whom correspondence should be addressed. Tel: 513 529 0357; Fax: 513 529 0333; Email: karroje@muohio.edu
Correspondence may also be addressed to C. Liang. Tel: 513 529 2336; Fax: 513 529 4243; Email: liangc@muohio.edu

other tools. From the user perspective, no tool in the literature can match PEACE for ease of installation or use, or the post-clustering analysis tools PEACE provides.

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 transparent parallel processing and providing various tools for result analysis.

PEACE: INSTALLATION AND USE

The PEACE GUI, available as a JAR file from the PEACE website (www.peace-tools.org), can be run on any machine supporting the standard Java Virtual Machine (JVM). The user can employ the GUI to install the clustering engine and perform a clustering of a data file in FASTA format, view an 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 (Figure 1).

Tool installation

To install the PEACE clustering engine onto a local or remote machine, the user selects from within the GUI the appropriate menu tab (Figure 2a), which then starts an install wizard that will prompt for the appropriate information. Figure 2b 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 2c illustrates the process of specifying the number of processors available (if running on a machine supporting the MPI protocol—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.

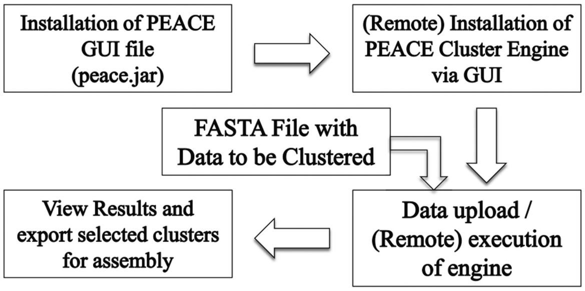

Installation of PEACE GUI file (peace.jar)
(Remote) Installation of PEACE Cluster Engine via GUI
FASTA File with Data to be Clustered
Data upload / (Remote) execution of engine
View Results and export selected clusters for assembly

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

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 2d).
- *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 were matched against the fragment header information from the original FASTA file, allowing the overlay of 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 for a given cluster. The method of setting up these classifiers, and the resulting histogram, is illustrated in Figure 2e.

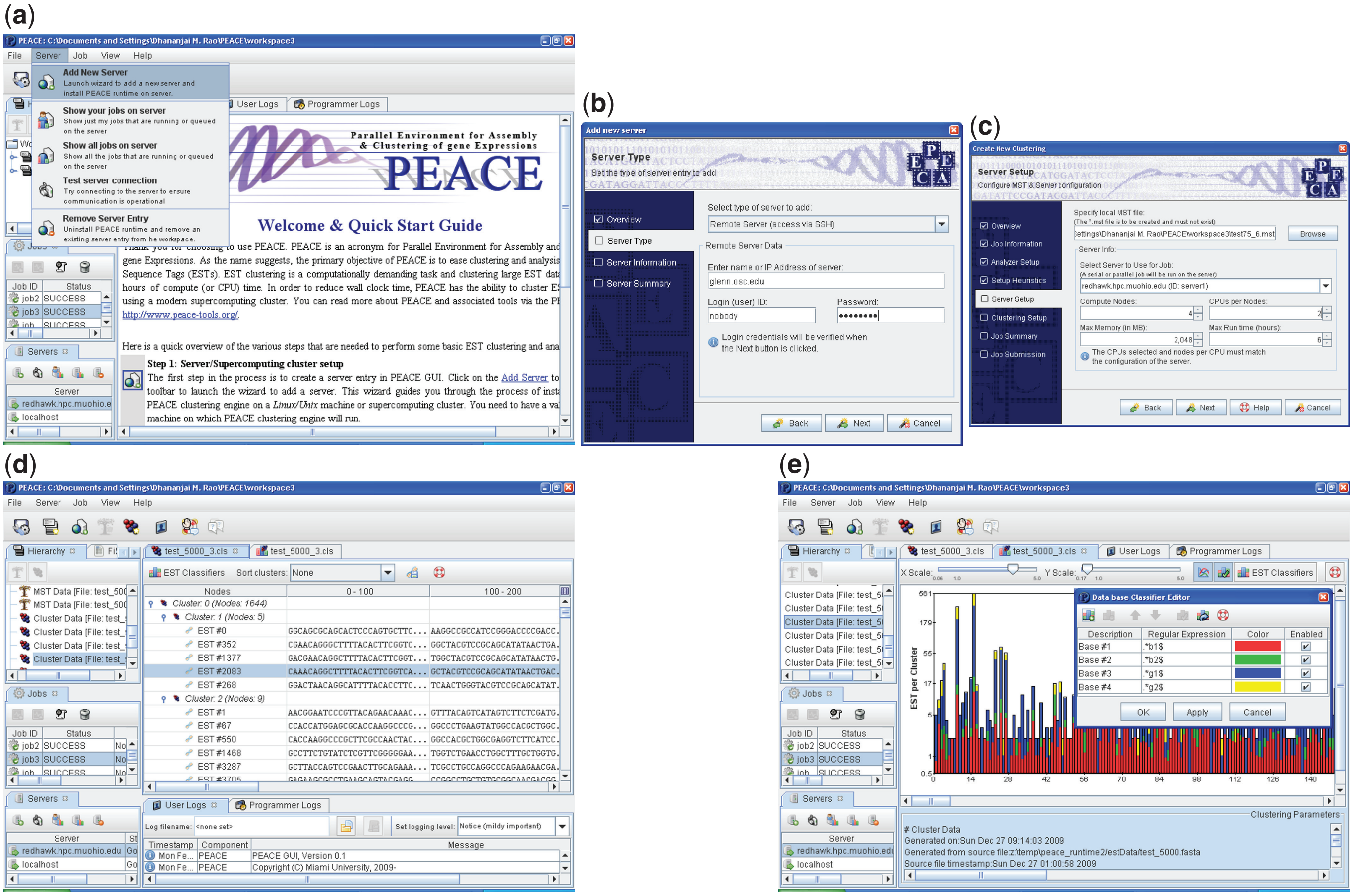
Extensive documentation for the tool has been posted on the PEACE website, as well as links to several tutorial videos demonstrating PEACE use and capabilities.

METHODS

The clustering performed by PEACE is based on the use of 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). The d^2 algorithm 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 Data 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



PEACE: C:\Documents and Settings\Dhananjai M. Rao\PEACE\workspace3

Add New Server
Launch wizard to add a new server and install PEACE runtime on server.

Show your jobs on server
Show just my jobs that are running or queued on the server

Show all jobs on server
Show all the jobs that are running or queued on the server

Test server connection
Try connecting to the server to ensure communication is operational

Remove Server Entry
Uninstall PEACE runtime and remove an existing server entry from he workspace.

Parallel Environment for Assembly & Clustering of gene Expressions
PEACE

Welcome & Quick Start Guide

Thank you for choosing to use PEACE. PEACE is an acronym for Parallel Environment for Assembly and gene Expressions. As the name suggests, the primary objective of PEACE is to ease clustering and analysis Sequence Tags (ESTs). EST clustering is a computationally demanding task and clustering large EST data hours of compute (or CPU) time. In order to reduce wall clock time, PEACE has the ability to cluster E using a modern supercomputing cluster. You can read more about PEACE and associated tools via the PI http://www.peace-tools.org/.

Here is a quick overview of the various steps that are needed to perform some basic EST clustering and ana

Step 1: Server/Supercomputing cluster setup
The first step in the process is to create a server entry in PEACE GUI. Click on the Add Server to toolbar to launch the wizard to add a server. This wizard guides you through the process of insta PEACE clustering engine on a *Linux/Unix* machine or supercomputing cluster. You need to have a va machine on which PEACE clustering engine will run.

Job ID	Status
job2	SUCCESS
job3	SUCCESS
job	SUCCESS

Server
redhawk.hpc.muohio.e
localhost

Add new server

Server Type
Set the type of server entry to add

☑ Overview
☐ Server Type
☐ Server Information
☐ Server Summary

Select type of server to add: Remote Server (access via SSH)
Remote Server Data
Enter name or IP Address of server: glenn.osc.edu
Login (user) ID: nobody
Password: ●●●●●●●●●
Login credentials will be verified when the Next button is clicked.

Create New Clustering

Server Setup
Configure MST & Server configuration

☑ Overview
☑ Job Information
☑ Analyzer Setup
☑ Setup Heuristics
☐ Server Setup
☐ Clustering Setup
☐ Job Summary
☐ Job Submission

Specify local MST file:
(The *.mst file is to be created and must not exist)
ettings\Dhananjai M. Rao\PEACE\workspace3\test75_6.mst
Server Info:
Select Server to Use for Job:
(A serial or parallel job will be run on the server)
redhawk.hpc.muohio.edu (ID: server1)
Compute Nodes: 4
CPUs per Nodes: 2
Max Memory (in MB): 2,048
Max Run time (hours): 6
The CPUs selected and nodes per CPU must match the configuration of the server.

test_5000_3.cls
EST Classifiers Sort clusters: None

Nodes	0 - 100	100 - 200
Cluster: 0 (Nodes: 1644)		
Cluster: 1 (Nodes: 5)		
EST #0	GGCAGCGCAGCACTCCCAGTGCTTC...	AAGGCCGCCATCCGGGACCCCGACC.
EST #352	CGAACAGGGCTTTTACACTTCGGTC...	GGCTACGTCCGCAGCATATAACTGA.
EST #1377	GACGAACAGGCTTTTACACTTCGGT...	TGGCTACGTCCGCAGCATATAACTG.
EST #2083	CAAACAGGCTTTTACACTTCGGTCA...	GCTACGTCCGCAGCATATAACTGAC.
EST #268	GGACTAACAGGCATTTTACACCTTC...	TCAACTGGGTACGTCCGCAGCATAT.
Cluster: 2 (Nodes: 9)		
EST #1	AACGGAATCCCGTTACAGAACAAAC...	GTTTACAGTCATAGTCTTCTCGATG.
EST #67	CCACCATGGAGCGCACCAAGGCCCG...	GGCCCTGAAGTATGGCCACGCTGGC.
EST #550	CACCAAGGCCCGCTTCGCCAACTAC...	GGCCACGCTGGCGAGGTCTTCATCC.
EST #1468	GCCTTCTGTATCTCGTTCGGGGGAA...	TGGTCTGAACCTGGCTTTGCTGGTG.
EST #3287	GCTTACCAGTCCGAACTTGCAGAAA...	TCGCCTGCCAGGCCCAGAAGAACGA.
EST #3705	[illegible]	[illegible]

Log filename: <none set> Set logging level: Notice (mildy important)

Timestamp	Component	Message
Mon Fe...	PEACE	PEACE GUI, Version 0.1
Mon Fe...	PEACE	Copyright (C) Miami University, 2009-

Data base Classifier Editor

Description	Regular Expression	Color	Enabled
Base #1	.*b1$	🟥	☑
Base #2	.*b2$	🟩	☑
Base #3	.*g1$	🟦	☑
Base #4	.*g2$	🟨	☑

```
# Cluster Data
Generated on:Sun Dec 27 09:14:03 2009
Generated from source file:z:\temp\peace_runtime2/estData/test_5000.fasta
Source file timestamp:Sun Dec 27 01:00:58 2009
```

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.

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 that 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 run time and memory usage for the data set sizes we expect to process.

PEACE approaches the problem by generating an MST 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 apply the full d^2 algorithm (see Sections A.3 of the Supplementary Data 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 Data, Section D.1, for details on the sequence size and error models.) Fragments were generated from the list of 100 zebrafish 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 Data).

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 run time for PEACE and WCD, again observing almost identical results when run

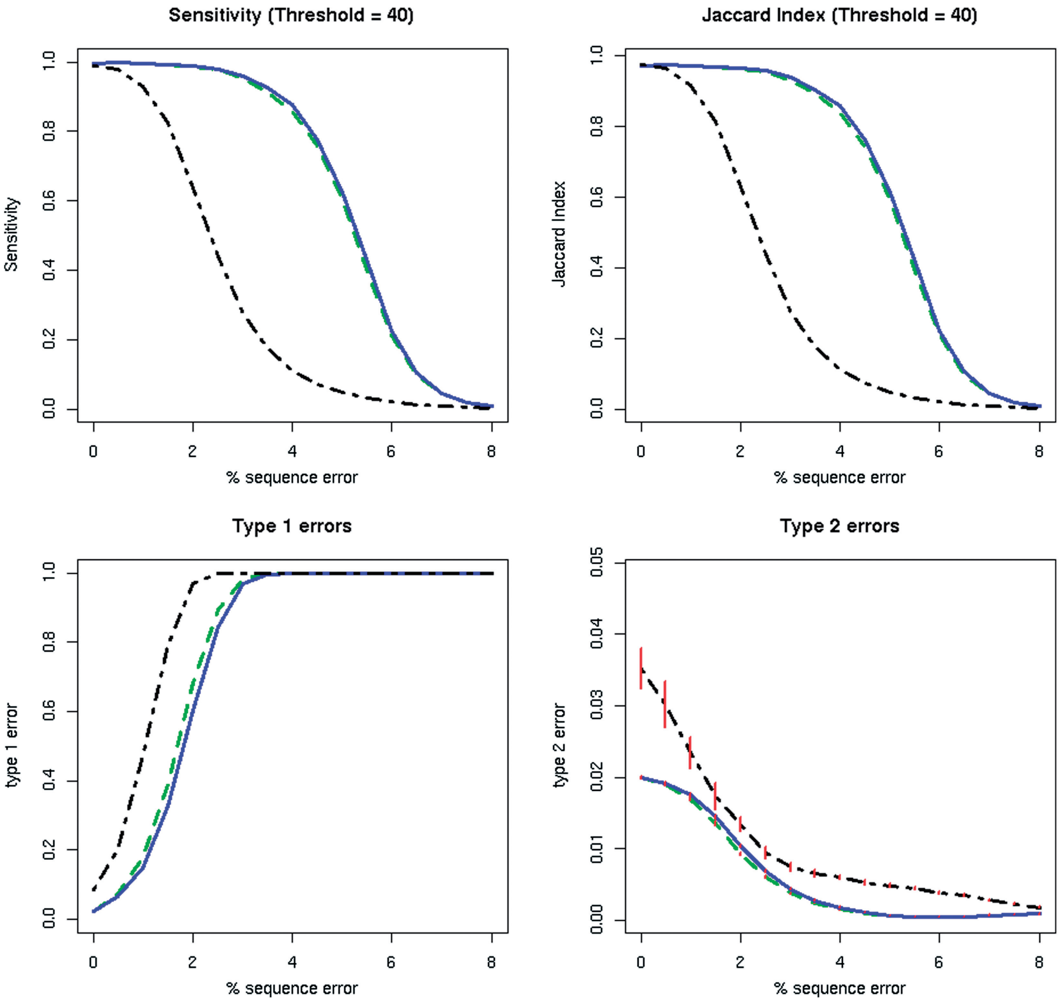


Figure 3. Comparisons of sensitivity, Jaccard Index, Type 1 error and Type 2 error, based on the average over 30 simulated Sanger sequence ESTs sets derived from 100 zebrafish genes (see Supplementary Data, Section D, for more details). Blue/solid, PEACE; Green/dash, WCD (version 0.5.1); Black/dot-dash, Cap3; vertical tics, 95% confidence intervals on estimates.

sequentially—but significantly faster run time for PEACE on multiple processors 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. We find that, as with WCD, the memory used for execution is linear in the size of the data set, requiring slightly more memory than WCD and considerably less than Cap3. (See Section D.4 and Supplementary Figure S3 of the Supplementary Data.)

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 Data, 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 Data, Section E.2), where we observe a significant increase in the incorrect merging of clusters. While this might be acceptable to a user planning to employ an assembly tool capable of breaking up the clusters (but unable to remerge a split cluster), for the purposes of clustering assessment it makes sense to base our analysis (and set our default values) at the lower threshold.

Comparisons against TGICL (15) are more complicated. We find that while TGICL is more sensitive at lower error rates, PEACE is more robust to error, and in all cases PEACE has an improved Type 2 error rate and is better able to distinguish duplicated genes. (See Supplementary Data, Section D.6.)

In applying the tools to real 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) (Table 1). We notice mixed results, with PEACE, WCD and TGICL showing comparable similarity, PEACE and WCD showing some superiority to TGICL in Type 1 error rate, but with a poorer showing in Type 2 error rate, and with all three tools coming considerably closer to the correct number of clusters than does Cap3. In run time, we see some inconsistency, with PEACE showing a 60% run time improvement over WCD in the first data set and a time comparable to TGICL, but requiring 20% more time than WCD in the *Arabidopsis*, data set and losing to TGICL by a large margin. In

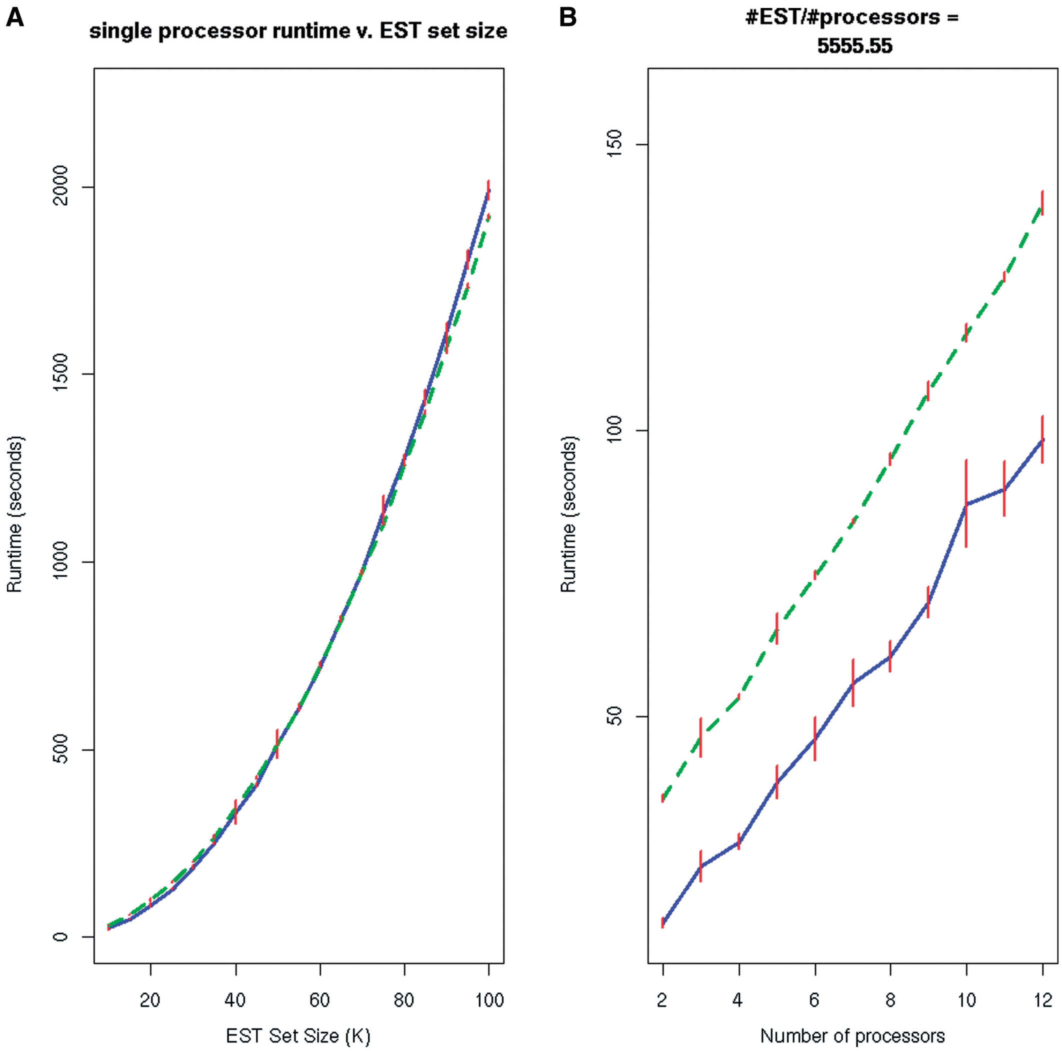


Figure 4. Comparisons of run time: (**A**) a comparison of the sequential run time of PEACE (blue) and WCD (green) on simulated sets ranging in size from 10K sequences to 100K sequences. (Cap3 run time is not reported, as the time spent on clustering cannot be differentiated from the time spent on assembly.) (**B**) 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 5555. 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 E5520 (2005).

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

		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
	TGICL	0.998	0.949	0.568	0.018	221	86	278
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
	TGICL	0.939	0.209	0.401	0.020	20248	1065	425

Supplementary Table S1, we present run times of several more sets for the distance-based tools, observing that while PEACE appears to be significantly faster on the smaller sets, WCD does overtake it for larger sets.

We tested the 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 200 bp to 370 bp with a mean of 250 bp and a standard deviation of 17 bp) and the Illumina short-read technology (producing reads of exactly 62 bp). In the 454 experiments, PEACE achieved a sensitivity of 0.871, an 89.7% improvement in sensitivity over WCD and a 326% improvement in sensitivity over TGICL, with comparable values for the Jaccard Index. WCD was unable to handle the Illumina data, while PEACE sensitivity, at a value of 0.384, was three times as good as TGICL (for more details, see Supplementary Table S3). In short, PEACE performs quite well for 454 reads and provides some useful information for Illumina reads, with considerably better results than WCD or TGICL.

CONCLUSIONS

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 parallel mode. Using a novel MST-based algorithm for the clustering of fragments by gene association, PEACE shows significant improvement in sensitivity over the competing WCD tool and TGICL tool when applied to NGS reads (4,15), 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 limitations. PEACE cannot handle duplicate genes; like WCD, it is unable to separate clusters corresponding to genes with a >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, while the ability to apply any assembler to small clusters 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 applications.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

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.

FUNDING

Funding for open access charge: Miami University, Departments of Computer Science and Botany.

Conflict of interest statement. None declared.

REFERENCES

1. Nagaraj,S., Gasser,R. and Ranganathan,S. (2007) A hitchhiker's guide to expressed sequence tag (EST) analysis. *Brief. Bioinform.*, **8**, 6–21.
2. Cheung,F., Haas,B.J., Goldberg,S.M.D., May,G.D., Xiao,Y. and Town,C.D. (2006) Sequencing medicago truncatula expressed sequenced tags using 454 life sciences technology. *BMC Genomics*, **7**, 272.
3. Emrich,S.J., Barbazuk,W.B., Li,L. and Schnable,P.S. (2007) Gene discovery and annotation using LCM-454 transcriptome sequencing. *Genome Res.*, **17**, 69–73.
4. Hazelhurst,S., Hide,W., Lipták,Z., Nogueira,R. and Starfield,R. (2008) An overview of the wcd EST clustering tool. *Bioinformatics*, **24**, 1542–1546.
5. Hide,W., Burke,J. and Davison,D.B. (1994) Biological evaluation of d2, an algorithm for high-performance sequence comparison. *J. Comput. Biol.*, **1**, 199–215.
6. Prim,R. (1957) Shortest connection networks and some generalizations. *Bell Syst. Tech. J.*, **36**, 1389–1401.
7. Burke,J., Davison,D. and Hide,W. (1999) d2_cluster: a validated method for clustering EST and full-length cDNAsequences. *Genome Res.*, **9**, 1135–1142.
8. Slater,G. (2000) Algorithms for analysis of exptressed sequence tags. *Ph.D. Thesis*. University of Cambridge, Cambridge.
9. Huang,X. and Madan,A. (1999) Cap3: a DNA sequence assembly program. *Genome Res.*, **9**, 868–877.
10. Parkinson,J., Guiliano,D. and Blaxter,M. (2002) Making sense of EST sequences by CLOBBing them. *BMC Bioinformatics*, **3**, 31.
11. Kalyanaraman,A., Aluru,S., Kothari,S. and Brendel,V. (2003) Efficient clustering of large EST data sets on parallel computers. *Nucleic Acids Res.*, **31**, 2963–2974.
12. Malde,K., Coward,E. and Jonassen,I. (2003) Fast sequence clustering using a suffix array algorithm. *Bioinformatics*, **19**, 1221–1226.
13. Pertea,G., Huang,X., Liang,F., Antonescu,V., Sultana,R., Karamycheva,S., Lee,Y., White,J., Cheung,F., Parvizi,B. *et al.* (2003) TIGR Gene Indices clustering tools (TGICL): a software system for fast clustering of large EST datasets. *Bioinformatics*, **19**, 651–652.
14. Ptitsyn,A. and Hide,W. (2005) CLU: a new algorithm for EST clustering. *BMC Bioinformatics*, **6(Suppl. 2)**, S3.
15. Picardi,E., Mignone,F. and Pesole,G. (2009) EasyCluster: a fast and efficient gene-oriented clustering tool for large-scale transcriptome. *BMC Bioinformatics*, **10(Suppl. 6)**, S10.
16. Jain,A., Murty,M. and Flynn,P. (1999) Data clustering: a review. *Comput. Surv.*, **31**, 264–323.
17. Wan,X.-F., Ozden,M. and Lin,G. (2008) Ubiquitous reassortments in influenza A viruses. *J. Bioinform. Comput. Biol.*, **6**, 981–999.
18. Hazelhurst,S. and Bergheim,A. (2003) ESTSim: a tool for creating benchmarks for EST clustering algorithms. *Technical Report CS-2003-1*, Department of Computer Science, University of Witwatersrand.
19. Richter,D.C., Ott,F., Auch,A.F., Schmid,R. and Huson,D.H. (2008) MetaSim: a sequencing simulator for genomics and metagenomics. *PLoS ONE*, **3**, e3373.
20. Wang,J-.P.Z., Lindsay,B.G., Leebens-Mack,J., Cui,L., Wall,K., Miller,W.C. and dePamphilis,C.W. (2004) EST clustering error evaluation and correction. *Bioinformatics*, **20**, 2973–2984.