Published online X *Nucleic Acids Research, X, Vol. X, No. X* **1–3**
doi:10.1093/nar/gkn000

PEACE: Parallel EST Analysis and Clustering Engine

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 tool for the high-throughput clustering of transcript fragment sequences produced Next Generation or Sanger sequence technologies. Installed and easily run through an easily downloaded GUI, PEACE can process large data sets, group the fragments by gene association, faster and with greater sensitivity than competing clustering tools such as WCD, and with considerably better sensitivity then the implicit clustering preformed by assembly tools such as CAP3. Through the GUI the user can collect cluster statistics and easily examine specific clusters for more comprehensive study. And by using PEACE as a "pre-assembly" tool to allow the application of an assembly tool to the individual clusters, the user gains both the significantly increase sensitivity of the clustering method and a large decrease in overall runtime for the clustering and assembly as compared to the application of an assembly tool to the data as a whole.

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 organization roll of the genome in the life cycle of any organism. Studying the transcriptome has led to gene discovery, provided information of protein isoforms, and generally help shed new light on the biological processes both controlling and dictated by the genome (Nagaraj *et al.*, 2007). But in order to get access those transcripts, we must first deal with the fragmented data produced by both Next Generation and traditional Sanger sequencing technology.

Given a set of transcript fragments sampled from multiple transcripts across the genome, a necessary first step is that of clustering: separating the fragments according to the transcript from which they were derived. While clustering is frequently preformes implicity by assembly tools such as CAP3 (Huang and Madan, 1999), clustering the data as a "pre-assembly" step has a number of advantages. Most significant among these: preforming this step will allow the application of the assembly tool to individucal clusters – saving significant amounts of time due to the reduced input size (Hazelhurst *et al.*, 2008). But clustering is a computationally challenging problem, especially when done *ab initio* instead of relying on access to a sequenced genome. Even with the number of fragments produced using Sanger sequencing, the runtime and memory requirements to cluster based on pair-wise sequence alignments make such an approach infeasible. The much larger data-set sizes produced by the short-read technologies exacerbates this problem.

Here we present PEACE, a tool for the *ab initio* clustering of transcript fragments by gene, applicable to both short-read 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 providing various tools for result analysis. PEACE can be run in single or multi-processor mode, and requires no Internet connectivity past the initial download. Using a *minimum spanning tree* (MST) based clustering strategy (Jain *et al.*, 1999, Wan *et al.*, 2008) and the d^2 sequence distance function (Hide *et al.*, 1994), PEACE generates the MSTs using a tailored version of Prim's algorithm (Prim, 1957). The result is an easy-to-use clustering tool that is considerably more sensitive to sequencing errors than the competing WCD tool (Hazelhurst *et al.*, 2008) and others in the literature without any sacrifice of performance (CITE) (e.g. (Burke *et al.*, 1999), (Slater, 2000), (Huang and Madan, 1999), (Parkinson *et al.*, 2002), (Kalyanaraman *et al.*, 2003), (Malde *et al.*, 2003), (Ptitsyn and Hide, 2005), (Picardi *et al.*, 2009)).

PEACE: INSTILLATION AND USE

The PEACE GUI, available from the PEACE website (www.peace-tools.org), handles the details of both instillation 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 PEACE management tool. Through this the user can install the computational tool and preform a clustering of a data file, view an initial analysis of the clusters and produce files containing subsets of the clusters for assembly by a tool such as CAP3 (Huang and Madan, 1999). A typical (first) use of the tool would generally require the the following steps.

Tool Instillation: 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 `Server > Add New Server` tab from the GUI main menu (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; in this example, 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.

*to whom correspondence should be addressed

Job Processing: After importing the target file into the GUI (`File > New Data Set`), the user can start a new job with the `Job > Job to Complete Clusters` tab 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 instillation). 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 and (if necessary) copied back, the user has several options for analysis:

- **Export:** The user can export the contents of one or more clusters into a fasta file, obtaining a subset of the original file containing the sequences corresponding to the selected clusters ready for processing by an assembly tool (e.g. CAP3 (Huang and Madan, 1999)).
- **View Clustering**: The user may view a list of all clusters, expanding selected clusters to a list of all fragments (illustrated in Figure 1(d)).
- **Classified Summary Graph:** The user may view a distribution of cluster sizes in a linear or log scale. 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 distribution does clusters containing matching elements in a specified colors (e.g. if the user wished to restrict their view to clusters containing transcript fragments from a specific source). 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 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 (Jain *et al.*, 1999, Wan *et al.*, 2008). We use the d^2 distance measure (Hide *et al.*, 1994) to represent edge weights. Prim's algorithm (Prim, 1957) 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 (Forêt *et al.*, 2009, Hide *et al.*, 1994). 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. Our method of calculating d^2, the *two-pass d^2 heuristic*, is a refinement of that used in the WCD clustering tool (Hazelhurst *et al.*, 2008). We augment the computation with a preliminary pass that narrows down the two regions of maximum similarity, and then explore those local regions in more detail.

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 (Hazelhurst *et al.*, 2008). 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 (Hazelhurst *et al.*, 2008) and the CAP3 assembly tool (Huang and Madan, 1999) (which calculates a cluster as an intermediate step). For our simulation tests we used the **ESTSim** tool (Hazelhurst and Bergheim, 2003) to generate simulated transcript fragments under varying models of error (Supplementary Materials, Section C1), generating the fragments from the list of 100 zebra fish genes used in the WCD testing (Hazelhurst *et al.*, 2008). 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) (Hazelhurst *et al.*, 2008, Wang *et al.*, 2004); 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 fragments derived from the *Chlamydomonas reinhardtii* genome. To compute sensitivity we used the *gmap* tool (Wu and Watanabe, 2005) 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 out preforming CAP3). Using the mouse fragment set used in Hazelhurst *et al.* (Hazelhurst *et al.*, 2008) PEACE again shows slight improvements in sensitivity and Type 1 error rates, with an 18% speed up for PEACE (see Supplementary Materials, Section D, for more details).

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, Jens Mueller and David Scoville at Miami University for their help with this project.

REFERENCES

Burke, J., Davison, D., and Hide, W. (1999). d2_cluster: a validated method for clustering est and full-length cdnasequences. *Genome Res*, **9**(11), 1135–42.

Forêt, S., Wilson, S. R., and Burden, C. J. (2009). Characterizing the d2 statistic: word matches in biological sequences. *Statistical applications in genetics and molecular biology*, **8**(1), Article 43.

Hazelhurst, S. and Bergheim, A. (2003). Estsim: A tool for creating benchmarks for est clustering algorithms. *Dept. of Computer Science, Univ. of Witwatersrand (South Africa), Tech. Rep. CS-2003-1*.

Hazelhurst, S., Hide, W., Lipták, Z., Nogueira, R., and Starfield, R. (2008). An overview of the wcd est clustering tool. *Bioinformatics*, **24**(13), 1542–6.

Hide, W., Burke, J., and Davison, D. B. (1994). Biological evaluation of d2, an algorithm for high-performance sequence comparison. *Journal of Computational Biology*, **1**(3), 199–215.

Huang, X. and Madan, A. (1999). Cap3: A dna sequence assembly program. *Genome Res*, **9**(9), 868–877.

Jain, A., Murty, M., and Flynn, P. (1999). Data clustering: a review. *Computing Surveys (CSUR*, **31**(3).

Kalyanaraman, A., Aluru, S., Kothari, S., and Brendel, V. (2003). Efficient clustering of large est data sets on parallel computers. *Nucleic Acids Res*, **31**(11), 2963–74.

Malde, K., Coward, E., and Jonassen, I. (2003). Fast sequence clustering using a suffix array algorithm. *Bioinformatics*, **19**(10), 1221–6.

Nagaraj, S., Gasser, R., and Ranganathan, S. (2007). A hitchhiker's guide to expressed sequence tag (est) analysis. *Brief Bioinformatics*.

Parkinson, J., Guiliano, D., and Blaxter, M. (2002). Making sense of est sequences by clobbing them. *BMC Bioinformatics*.

Picardi, E., Mignone, F., and Pesole, G. (2009). Easycluster: a fast and efficient gene-oriented clustering tool for large-scale transcriptome .... *BMC Bioinformatics*.

Prim, R. (1957). Shortest connection networks and some generalizations. *Bell System Technical Journal*.

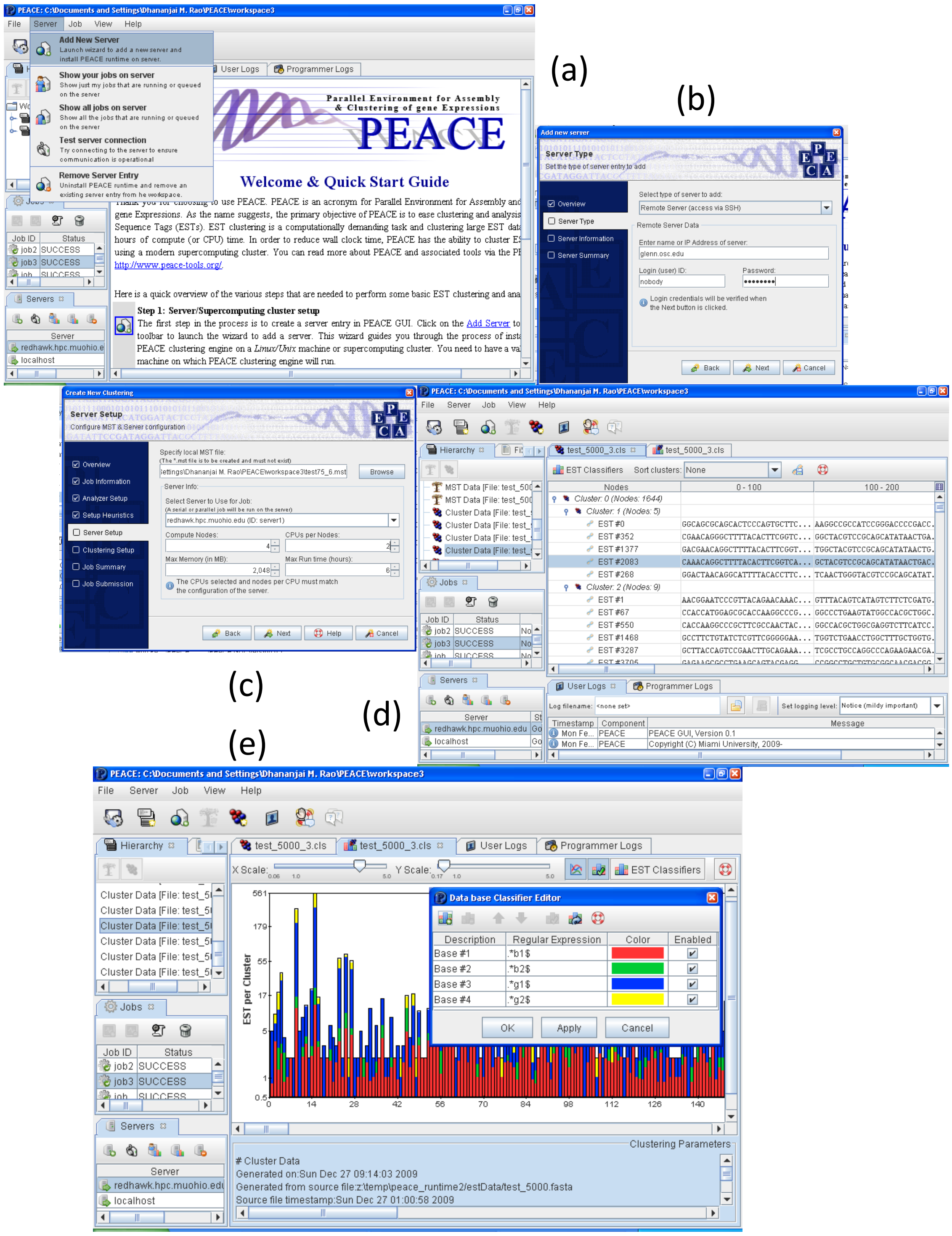
Ptitsyn, A. and Hide, W. (2005). Clu: a new algorithm for est clustering. *BMC Bioinformatics*, **6 Suppl 2**, S3.

Slater, G. (2000). *Algorithms for analysis of exptressed sequence tags*. Ph.D. thesis, University of Cambridge, Cambridge.

Wan, X.-F., Ozden, M., and Lin, G. (2008). Ubiquitous reassortments in influenza a viruses. *Journal of bioinformatics and computational biology*, **6**(5), 981–99.

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**(17), 2973–84.

Wu, T. D. and Watanabe, C. K. (2005). Gmap: a genomic mapping and alignment program for mrna and est sequences. *Bioinformatics*, **21**(9), 1859–75.


(a)
(b)
(c)
(d)
(e)
PEACE: C:\Documents and Settings\Dhananjai M. Rao\PEACE\workspace3
File
Server
Job
View
Help
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.
User Logs
Programmer Logs
Parallel Environment for Assembly & Clustering of gene Expressions
PEACE
Hierarchy
Jobs
Servers
Add new server
Server Type
Set the type of server entry to add
Overview
Server Type
Server Information
Server Summary
Back
Next
Cancel
Create New Clustering
Server Setup
Configure MST & Server configuration
Overview
Job Information
Analyzer Setup
Setup Heuristics
Server Setup
Clustering Setup
Job Summary
Job Submission
Browse
Help
test_5000_3.cls
EST Classifiers
X Scale:
Y Scale:
EST per Cluster
Data base Classifier Editor
OK
Apply
Cancel
Clustering Parameters

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 PE 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

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.

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.

MST Data [File: test_500
MST Data [File: test_500
Cluster Data [File: test_
Cluster Data [File: test_
Cluster Data [File: test_
Cluster Data [File: test_

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.

Job ID	Status	
job2	SUCCESS	No
job3	SUCCESS	No
job	SUCCESS	No

Server	St
redhawk.hpc.muohio.edu	Go
localhost	Go

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-

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 1. Screenshots of the PEACE GUI during execution, including (a) GUI Welcome and server instillation 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.

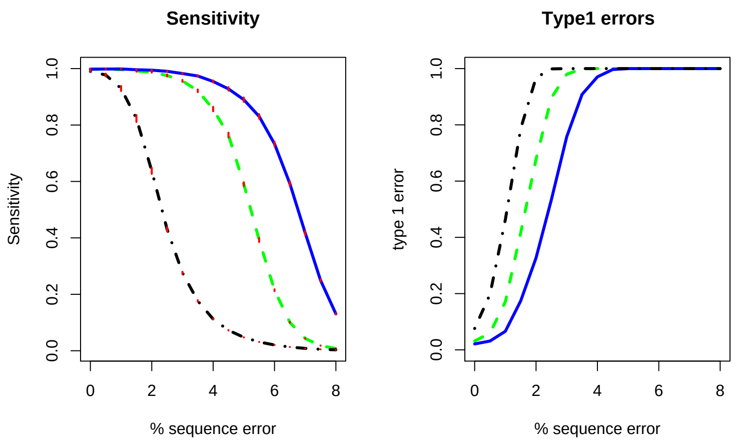


Figure 2. Comparisons of Sensitivity and Type 1 error, based on the average over 30 Simulated fragment sets derived from 100 zebra fish genes (see Supplementary Materials, Section C, for more details). Blue/Solid = PEACE, Green/Dash = WCD , Black/Dot-Dash = CAP3 ; vertical tics = 95% confidence intervals on estimates. Intervals are not presented for Type 1 error due to the extremely small variance.

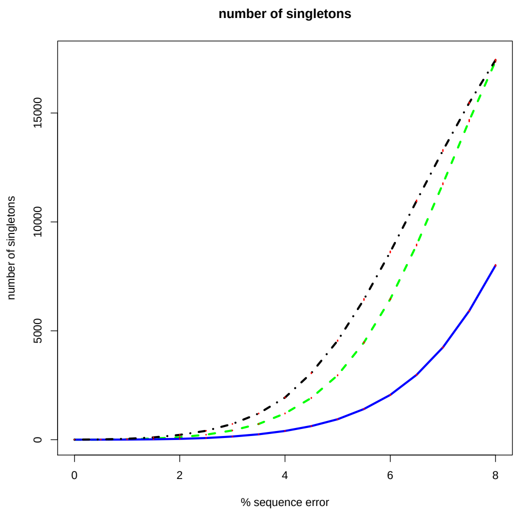


Figure 3. Average number of fragments flagged as singletons by each tool when run on the simulated sequences; correct answer in all cases is zero. Blue/Solid = PEACE, Green/Dash = WCD , Black/Dot-Dash = CAP3 .