PEACE: Parallel Environment for Assembly and Clustering of Gene Expression

A Distance metrics, heuristics and filters

A.1 The d^2 distance metric

d^2, as outlined by *Hide et al.* [Hide *et al.* (1994)], is an *alignment free* distance pseudo-metric which quantifies local similarity between sequences based on a simple word count. Let $c_x(w)$ denote the number of times word w occurs in string x. We search for similarity between strings x and y by looking at the difference between $c_x(w)$ and $c_y(w)$ for different words w. Specifically, for all words of a fixed length k, we calculate:

$$d_k^2(x,y) = \sum_{|w|=k} (c_x(w) - c_y(w))^2$$

However, applying such a definition to two sequences as a whole leads to a measure of global similarity, while we want to measure local similarity (thus, for example, assigning two sequences with sufficiently long overlapping ends to be at a distance of zero – or a small distance if errors are present). For this, instead of comparing the entire two strings, we compare sliding windows from each string of a fixed size r. Formally, for sequences x and y ($|x| \geq r$, $|y| \geq r$), we define:

$$d^2(x,y) = \min\left\{d^2(u,v) \ : \ u \sqsubseteq x, v \sqsubseteq y, |u| = |v| = r\right\} \tag{S1}$$

(where $u \sqsubseteq x$ denotes that u is a substring of x). Defined as such, d^2 is, in a mathematical sense, a *pseudo-metric*: $d^2(x,y) = 0$ does not imply $x = y$.

The PEACE implementation of d^2 was initially based on the description from *Hazelhurst* [Hazelhurst (2004)]. For parameters we adapted those used by the WCD clustering tool [Hazelhurst *et al.* (2008)]: a word size $k = 6$ and a window size $r = 100$.

A.2 Two-pass d^2

Our "two-pass d^2" algorithm works by sampling a subset of window pairs evenly distributed across the sequences, narrowing down a smaller region in which to search for the

best scoring window-pair. In the first pass, the algorithm look at every length r window on one sequence, but on the other sequence we skip the window by s bases between every sampling. It then take the best such window pair, extend each of these two windows by s bases in each direction, and apply (S1) to this limited region.

In PEACE, we use $k = 6$, $r = 100$, and $s = 50$.

A.3 Filtering heuristics

PEACE uses the u/v and t/v heuristics, roughly as described by *Hazelhurst et al.* [Hazelhurst *et al.* (2008)] as filters that allows us to avoid over 99% of the potential d^2 calculations. Each heuristic takes two sequences and estimates whether it is worth proceeding to the d^2 computation by sampling and comparing word frequency across the two sequences. The u/v heuristic looks at every h-th word of size v on one sequence, and rejects if it does not find at least u occurrences of these words on the other sequence. The t/v heuristic demands that there be at least t size v words on one sequence that occur within a l base range within the second sequence. If a sequence pair meets the requirements of both these filters, then we compute the d^2 distance.

In PEACE, we set $h = 16$, $v = 8$, $u = 4$, $t = 65$, and $l = 100$.

B MST-Based Calculations

While d^2 serves to quantify sequence distance, the basis for the clustering algorithm is the minimum spanning tree (MST). Viewing the ESTs as nodes and the data set as an d^2-weighted graph, the derivation of a minimum spanning tree results in the placement of nodes of a given tree into a restricted neighborhood of the graph. By then removing larger edges, we are left with connected components corresponding to the gene-based clusters. The MST-based clustering method has been used effectively in other applications, but to our knowledge this is the first such use for the EST clustering problem [Jain *et al.* (1999); Wan *et al.* (2008)].

B.1 Calculation of the MST

To calculate this MST we use Prim's algorithm [Prim (1957)]. For a graph of n nodes and e edges, Prim's can be implemented such that the algorithm has an $O(e + n \log n)$ worst-case runtime. However, the *narrow-band* nature of the model allows us to reduce this bound. If we were to model only the edges connecting adjacent nodes, we would find the node degrees to be very small relative to n: any EST overlaps only a few others, and has no connection to a vast majority of the data set members. Since we can quickly eliminate most of these excess edges with the u/v and t/v heuristics (removing more than 99% of all edges before applying Prim's), we find in practice that e is a very small fraction of n^2. In the runtime results (see Section C.4 and Figure C.4), we find that when holding

the EST size distribution constant, the tool as a whole has a runtime of $O(n^2)$ – appearing to dominated by the time required to apply the filtering heuristics to every EST pair.

B.2 Removing edges

Once the MST is has been calculated, our last step is to remove all edges exceeding a threshold weight T, taking the resulting components as our clusters. WCD, when faced with a similar challenge, sets the threshold at $T = 40$, hypothesizing that EST pairs with a d^2 distance of greater than 40 are unlikely to overlap. We have set out threshold to $T = 130$.

C Simulated Test Results

In *Hazelhurst et al.* [Hazelhurst *et al.* (2008)] the authors conduct an investigation of WCD against a number of clustering tools. As they make a convincing argument that WCD is returning better results then the tools against which they compare, we limit our analysis to a comparison of PEACE against WCD and the CAP3 assembly tool [Huang and Madan (1999)] – a tool which implicitly clusters while performing assembly. We do so by applying all three tools to a number of data sets, both simulated and real.

C.1 Simulation Tool and Parameters

For simulated data sets we use the **ESTsim** tool to generate simulated EST data sets [Hazelhurst and Bergheim (2003)], using the collection of zebra fish genes that served as a basis for the WCD simulations [Hazelhurst *et al.* (2008)]. In generating the ESTs, ESTsim models three types of error (general base read errors, errors due to polymerase decay, and primer interference), and allows those errors to take the form of substitutions, deletions, and insertions (of bases and Ns). See the paper for a discussion of the probability distributions and default parameters. For generating our simulated data, we use the default values for all parameters that are not explicitly being subjected to variation in our experiments – paralleling the testing of the WCD tool.

C.2 Methodology

Each estimate given in the main paper, or in the following, is averaged over 30 trials. Each trial consists of the application of all three tools to a simulated data set, the set having been derived from the application of the ESTsim tool to a set of 100 zebra fish gene sequences [Hazelhurst and Bergheim (2003)]. All confidence intervals are calculated at a 95% level of significance.

C.3 Result Quality

While our primary measurement of tool result is sensitivity (Figure 1), a number of alternative measurements were investigated as well. The *Jaccard* index measures both sensitivity and specificity, calculated as the ratio of true positives to the sum of true positives, true negatives, and false positives ($tp/(tp+fn+fp)$) [Hazelhurst *et al.* (2008)]. Notably, when we removed duplicated genes before generating the simulated EST sets, we found all three tools to have 100% specificity: no false positives. In such a case the Jaccard Index is identical to the Sensitivity Index, hence PEACE exhibits the same improvement on both indexes relative to WCD and CAP3. Applying the tools to the original genes (in which 21% of the genes were highly similar to genes in the remaining 79%), we find only a slight difference in the Jaccard Index, and the same relative performance of the three tools (Figure C.3). The sensitivity of each tool to duplications is discussed below.

Type 1 and Type 2 errors, as defined in *Wang et al.* [Wang *et al.* (2004)], measure quality at the level of the genes from which the ESTs were derived. For Type 1 error, we look at the fraction of genes that were broken into two or more partitions, while in Type 2 error we look at the fraction of clusters that contain two or more genes. In Figure C.3 we compare the Type 1 and Type 2 error rates of the three tools for varying levels of simulated base read error. For Type 1 error we find that while all tools do equally poorly when the EST is subjected to more than a 5% error rate, PEACE does significantly better than WCD in the lower ranges (while WCD does correspondingly better than CAP3). When looking at Type 2 error rate, we find that PEACE does a generally better job than CAP3 in separating genes into different clusters. We note, however, that Type 1 error is the more important measurement: genes incorrectly split between clusters cannot be later recovered (without re-examining the cluster as a whole), but there is the potential for an assembly program to later identify incorrectly merged clusters based only on local cluster information.

One of the difficulties in clustering EST data is dealing with highly similar genes. Genes with a high degree of similarity will produce ESTs that reflect that similarity, hence appear to overlap – resulting in the incorrect clustering of ESTs from separate genes. Assembly tools such as CAP3 may have more ability to discriminate between clusters given their more intensive investigation of overlaps, but highly similar sequences are going to cause a problem for any tool.

In Figure C.3 we look at the ability of each tool to separate duplicates as a function of the % divergence between the duplications. Unsurprisingly, CAP3 (the assembler) does the best here, able to effectively separate duplicates at 92% similarity or less. PEACE and WCD are roughly comparable, both clearly able to separate duplicates out at a similarity level of 83% – but completely unable to distinguish sharing a similarity of 88% or more.

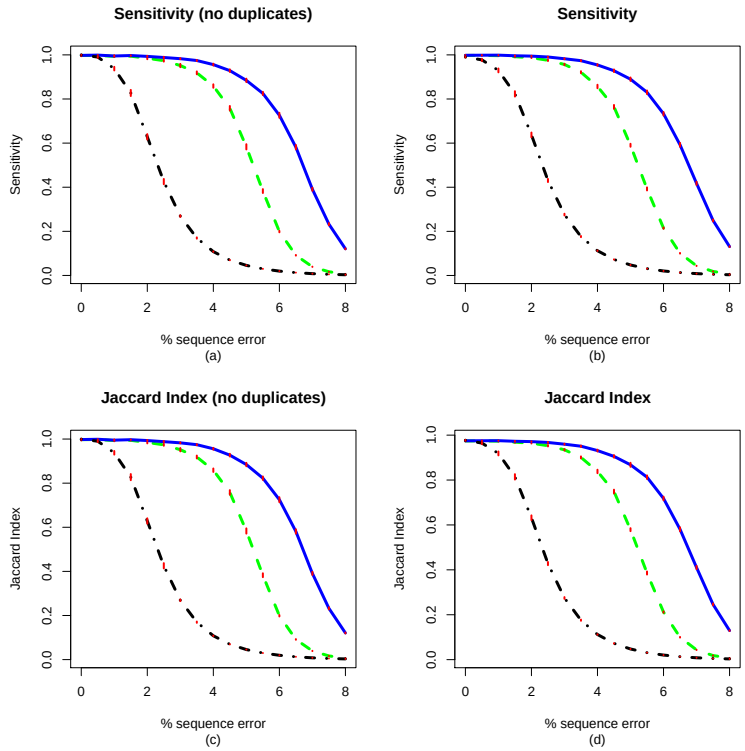

Figure S1: Comparisons of Sensitivity and the Jaccard Index as a function of base read error rate, based on 30 simulated EST sets derived from a set of 79 genes with no significant similarity (left column), and an additional 21 genes showing significant similarity to the base set (right column). Blue/Solid = PEACE, Green/Dash = WCD, Black/Dot-Dash = CAP3; vertical tics = 95% confidence intervals on estimates.

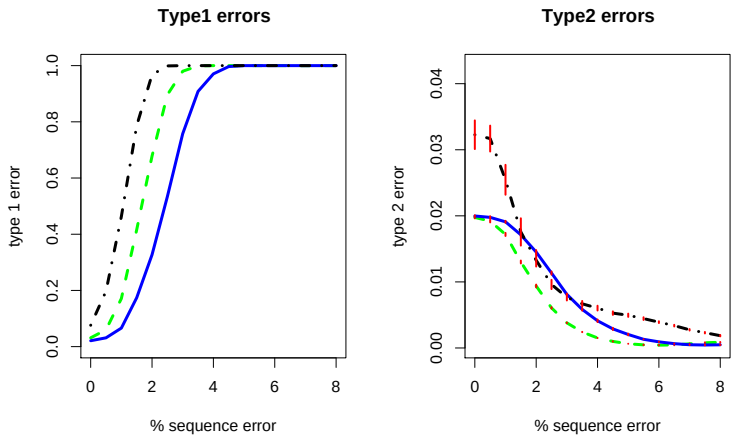

Figure S2: Type 1 error rate (fraction of genes partitioned) and Type 2 error rate (fraction of clusters containing multiple genes) as a function of base read error rate. Estimates averaged over application to 30 simulated ESTs sets derived from the full 100 zebra fish gene set. Blue/Solid = PEACE, Green/Dash = WCD, Black/Dot-Dash = CAP3; vertical tics = 95% confidence intervals on estimates (not shown for Type 1 error due to the extremely small variance in estimates).

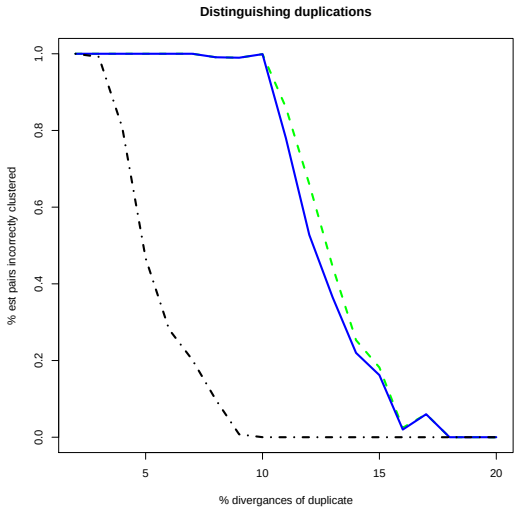

Figure S3: Ability to distinguish duplicates as a function of divergence. Estimates averaged over 30 trials; one trial consists of taking a random gene, copying it and stochastically changing bases at the specified rate, then using the two genes as the bases for generating a simulated set. Blue/Solid = PEACE, Green/Dashed = WCD, Black/Dot-Dashed = CAP3; variance was too small for visible plotting of confidence intervals.

C.4 Runtime

In Figure C.4 we see give a runtime comparison of PEACE and WCD, seeing a slight improvement in PEACE. (As CAP3 performs both clustering and assembly, runtime comparisons are not meaningful.) As stated, we see an almost perfectly quadratic curve ($r > 0.99$), indicating that the runtime is dominated by the time required to look at every EST pair (either filtering or computing d^2). As both PEACE and WCD can be run in a parallel mode, we also examined runtime as a function of the ratio of EST set sizes to number of processors used. In Figure C.4 we show two such ratios, again seeing an improvement in PEACE over WCD.

D Applications to Real Data Sets

In Table D we lay out more comprehensive results for the application of our tool to an EST set collected from the *Chlamydomonas reinhartdii* genome, while in Table S2 we see the results for the mouse set used for the WCD analysis [Hazelhurst *et al.* (2008)]. We note that for the Chlamydomonas genome the correct clustering is not known; our estimates of sensitivity are based on as assignment of ESTs to the sequenced genomes by the Gmap tool [Wu and Watanabe (2005)], leaving some potential for error in the reference. Further, we found a number of ligations in the data (ESTs derived by incorrectly combining multiple transcripts), which were removed before testing.

Interestingly, where in the simulated results we saw significant improvement in PEACE over WCD in terms of result quality and only a light improvement in runtime, here we see the reverse. The two tools are roughly comparable in terms of sensitivity, Type 1 error and Type 2 error, while PEACE shows a 10% and 20% improvement in runtime (both showing significant improvements over CAP3). Predictably , the full assembly tool does somewhat better in terms of specificity (the Jaccard Index and Type 2 error), though scores here are worse for all three tools due to the increased number of homologous gene pairs as compared to our simulated sets.

	Sensitivity	Jaccard	Type 1 error	Type 2 error	Number of Clusters	Singletons	Single processor runtime (s)
PEACE	0.958	0.386	0.623	0.026	19649	10493	8799
WCD	0.94	0.500	0.654	0.021	22433	13000	9913
CAP3	0.77	0.740	0.737	0.015	33771	22169	

Table S1: **Chlamydomonas reinhartdii**: 189975 ESTs, average length 553 bp, estimated 9886 actual genes represented.

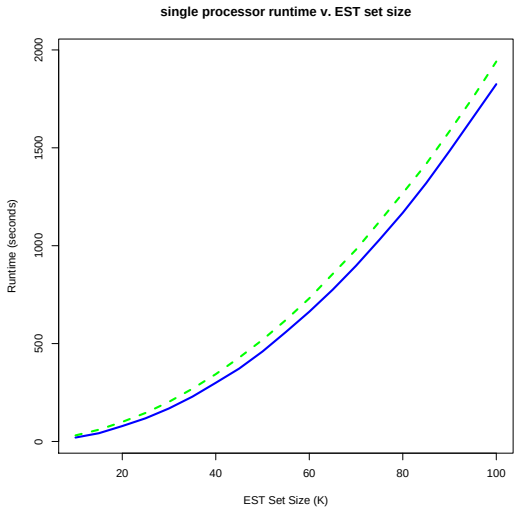

Figure S4: Single Processor Runtime as a function of input size, averaged over 50 simulated EST sets; Blue/Solid = PEACE, Green/Dashed = WCD; vertical tics represent= 95% confidence intervals on estimates.

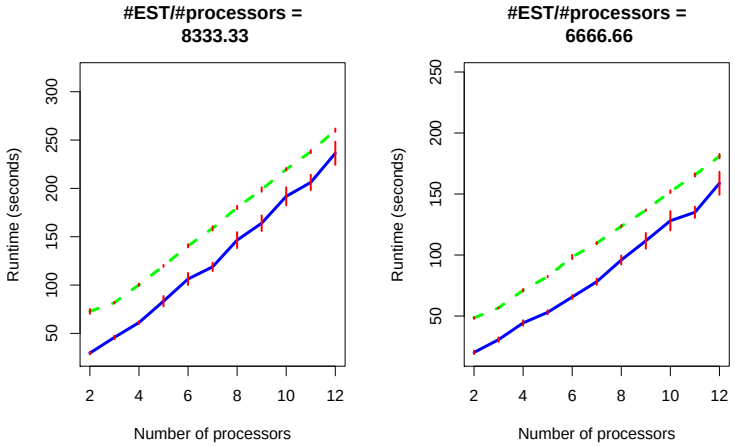

Figure S5: Runtime of a function of input size when holding constant the ratio of set size to number of processors used; averaged over 30 simulated EST sets. Blue/Solid = PEACE, Green/Dashed = WCD; vertical tics represent= 95% confidence intervals on estimates.

	Sensitivity	Jaccard	Type 1 error	Type 2 error	Number of Clusters	Singletons	Single processor runtime (s)
PEACE	0.936	0.348	0.340	0.034	18338	8206	820
WCD	0.932	0.476	0.350	0.0275	18787	8554	1000
CAP3	0.830	0.805	0.486	0.0135	25040	14925	

Table S2: **A076951 (Mouse EST set)**: 76914 ESTs, average length 427, 13240 actual genes represented.

References

Hazelhurst, S. (2004). An efficient implementation of the d2 distance function for est clutering: preliminary investigations. *Proceedings of the SAICSAT*, pages 1–5.

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).

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

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.