Evaluating EST Clustering with Simulated Data

Ke Ye and Bernard M.E. Moret
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131
Email: {keye,moret}@cs.unm.edu

Abstract— **Expressed sequence tags (ESTs) provide a rapid and inexpensive approach to gene sequencing, gene discovery, and the characterization of gene regulation and alternate splicings. The first step in using ESTs is to cluster them, that is, to group them according to the gene that produced them. In these applications, clustering performance and quality are the most critical issues. Most efforts in EST clustering have focused on biological data, but such data make assessment of clustering quality and robustness difficult.**

In this paper, we use the EST simulator of Hazelhurst and Bergheim, `ESTSim`, to study experimentally the accuracy and robustness of four different EST clustering tools, `ESTate`, `N2tool`, `BlastClust`, and simple hierarchical agglomerative clustering based on Smith-Waterman similarities, using the Rand index as a measure of clustering quality.

Our results indicate that `N2tool` consistently dominates `BlastClust` and that, in turn, `ESTate` and clustering based on Smith-Waterman similarities greatly outperform `N2tool`, with the much slower Smith-Waterman-based clustering slightly outperforming `ESTate`. We also find that, of the various parameters affecting EST generation that can be set in the `ESTSim` simulator (EST length, indels and substitutions, polymerase decay, stutter, and ligation), polymerase decay has by far the largest effect on the quality of clustering—a finding that could lead to improved EST and assembly quality through the control of bench procedures.

I. Introduction

Expressed sequence tags (ESTs) provide a rapid and inexpensive approach to gene discovery [1], [2], the characterization of gene expression and regulation [3], the identification of gene function [4], and the study of alternative splicing [5]. However, because they are the product of a single read, EST sequences are incomplete and error-prone; they must be processed through a chain of bioinformatics tools in order to produce useable genetic information. A crucial step in this chain is the clustering of ESTs, that is, the partitioning of the collection of EST sequences into groups that correspond to the genes (or, more accurately, the transcripts) that produced them.

Evaluating the accuracy of EST clustering is thus of real significance, as is assessing its robustness in the face of the various sources of error that affect the quality of ESTs. Such an evaluation is difficult to conduct with actual biological data, because we do not know the correct answer. Thus we use simulation as our basic experimental tool; more specifically, we use real cDNA data to create simulated ESTs (under a model that includes a number of error mechanisms), feed them to the clustering algorithms, and compare the results to the known correct answers. Hazelhurst and Bergheim [6] developed a program called `ESTSim` to generate artificial ESTs under a highly parameterized error model. We use their program to generate a variety of datasets with which to test four different EST clustering tools: `ESTate`, designed specifically to cluster ESTs, `N2tool`, designed to cluster rapidly collections of DNA sequences, `BlastClust`, an even faster tool based on `BLAST` scores, and a simple agglomerative hierarchical clustering using Smith-Waterman subsequence alignment scores. We then assess the quality of the resulting clusters by computing the Rand index [7] to evaluate the results. None of these tools is used in large assembly projects, where the EST clustering is generally a part of the assembler, as in the TIGR assembler (see, e.g., [8]) or `CAP3` [9], or where only the results of the clustering are made available, as in NCBI's Unigene (see, e.g., [10]). However, it is difficult to isolate the clustering computations in these large programs, and the four approaches we chose are representative of the main approaches to EST clustering, including those used in the assemblers.

We find that `ESTate` and the clustering based on Smith-Waterman similarity scores are the most accurate, by a very significant margin, and that both are relatively robust against variation in EST length and most sources of error. However, we also find that all four clustering methods suffer most strongly from errors caused by polymerase decay—a finding that could lead to improved EST and sequence assembly quality through the control of bench procedures.

II. ESTs and EST Clustering

An expressed sequence tag (EST) is a short sequence from an expressed gene, typically from 300bp to 500bp. They are commonly produced through a multistep process in which full-length mRNAs are extracted from cells, purified, then reverse transcribed into complementary DNAs (cDNAs), which are used as templates to produce double-stranded cDNAs. The double-stranded cDNAs are then inserted into vectors, cloned, and sequenced. ESTs are the random partial single reads from either end of these cDNA clones.

Since a single read is obtained for each EST, the EST sequence is of relatively low quality [11]—the error rate is as high as a few percent. ESTs suffer from the following problems: (i) compression and base-calling errors resulting from frameshifts; (ii) clone orientation, associated clone ID chimeras and missing 3' or 5' ends; (iii) contaminating sequences including genomic, vector, ribosomal DNA, and

cDNAs from unrelated species; and (iv) lack of annotations. The main approach to reducing errors is to cluster the ESTs, then assemble them (using overlap) with as high a coverage as possible, attempting to produce the longest possible contigs (contiguous sequences in the assembly).

An EST cluster is an index class that consolidates all ESTs originating from the same transcript—or, in the simpler cases, from the same gene. (Through alternative splicing, several forms of transcripts can originate from the same locus on a gene, thereby complicating the picture. Attempts are being made to collect data on alternatively spliced ESTs [12].) EST clustering is the process that produces such clusters from a set of EST sequences. Since each cDNA has multiple copies and the copies from the same gene product are mostly identical or quite similar, they overlap. These overlaps, if of sufficient size to reach significance, provide the main tool for EST clustering.

The clustering process involves five phases: preprocessing, initial clustering, assembly, alignment processing, and cluster joining. Preprocessing includes screening out low-quality regions, contaminations, vector sequences, and repeat sequences, in order to minimize the probability of clustering unrelated sequences. Initial clustering, the phase of interest to us, partitions the collection of sequences into clusters. Assembly takes the sequences placed into the same cluster and attempts to align them with each other on the basis of local overlaps to produce long contigs. Alignment processing then checks the resulting alignments for errors or alternative forms and generates consensus sequences. Finally, cluster joining joins clusters according to the cDNA clone information such as clone ID and the 5' and 3' reads information. The last three steps are highly specialized for the genes under consideration, the reason why EST clustering is usually integrated within an assembler. Our target in this paper is the second phase, which has not been well studied in terms of the respective attributes of competing strategies nor in terms of their sensitivity to the various sources of error that escape the preprocessing phase.

All published EST clustering algorithms use some form of similarity computation; most usually work in two steps—they compute a pairwise similarity matrix for the ESTs and then use this matrix to produce clusters [2]. The first step is often critical and always application-dependent; it determines much of the quality of the clustering results. We can therefore distinguish three main classes of EST clustering algorithms according to the choice of similarity measure.

- *Clustering based on alignment scores.* The clustering tools used in assemblers all fall in this category, using `BLAST` or similar techniques to obtain alignment scores very quickly. Our last three approaches (`N2tool`, `BlastClust`, and hierarchical clustering using Smith-Waterman similarity scores) fall in this category as well. Of these various tools, some use general-purpose clustering algorithms, others attempt to tailor the clustering to the characteristics of ESTs.
- *Clustering not using alignment scores.* These algorithms rely on pattern matching (single words, multiple words, etc.) in order to establish similarity; they generally incorporate much domain knowledge in the choice of these patterns. Perhaps the first of these tools was D2-cluster [13]; `ESTate` falls within this category.

III. Experimental Design

A. Simulating ESTs

Real EST sequences are of low quality, containing many errors; they also display the characteristics nucleotide and codon-usage biases of the transcripts from which they originate. Thus a good simulator needs to be able to generate ESTs from real transcripts or cDNA sequences and needs to incorporate models for various types of errors. Such a program was developed by Hazelhurst and Bergheim at Witwatersrand University; their code, `ESTSim` [6], generates simulated, but very realistic ESTs from a collection of given cDNA sequences. The program includes models for single-base errors such as insertion, deletion, and substitution; two kinds of polymerase decay; stutter; and ligation.

In our study, we started from cDNA sequences from the human cDNA library in the mammalian collection at the National Cancer Institute (`mgc.nci.nih.gov`). We generated 20 datasets, each with around 120 EST sequences. The length of each EST sequence ranges from 300bp to 500bp. We tested various error models in an initial screening phase, then focused on single base-pair errors, polymerase decay, and a modest amount of stuttering.

B. Representative Clustering Algorithms

As discussed earlier, we selected an agglomerative hierarchical clustering scheme based on Smith-Waterman subsequence similarity scores, `BlastClust`, `N2tool`, and `ESTate` as our test algorithms. The first three are based on alignment scoring—one using the slow dynamic programming computation to obtain accurate similarity scores, the other two using less accurate, but faster similarity computations. The last is based on pattern matching. We downloaded the code for the three named programs and implemented our own version of Smith-Waterman and of agglomerative hierarchical clustering.

The classic Smith-Waterman algorithm [14] uses dynamic programming to compute "optimal" alignments between sequences, optimal, that is, in terms of indels and substitutions under the chosen scoring function. We used the standard scoring (1, -1, and -2) and kept independent gap penalties, the latter because we wanted to reflect the emphasis on contigs and the fact that gaps, in this setup, are entirely the product of errors; independent gap penalties also allow the dynamic program to run in quadratic time. Agglomerative hierarchical clustering begins by placing each item in its own clusters, then proceeds by merging a pair of clusters, adjusting similarity values between clusters to reflect the merger, and repeating until a single cluster is obtained, forming a rooted binary tree. The final clusters are determined by using similarity gaps between the various merging steps in the algorithm (or, if that is known in advance, when the target number of clusters has been reached).

`BlastClust` [4] is a clustering tool designed especially to cluster protein or DNA sequences based on pairwise matches returned by the BLAST algorithm. It uses BLAST scores to assign statistical significance, matches pairs that reach that level of significance, then constructs clusters using a simple, greedy, single-linkage clustering method.

`N2tool`, a part of the `ICAtools` suite [15], runs pairwise comparisons of all input sequences to identify which share a region (which can be quite small) of similarity. Sequences that share such a region are then placed into a cluster—with the result that the same sequence may appear in many clusters. Other tools in the `ICAtools` suite can then disambiguate the results. One of the original purposes of `N2tool` was the discovery of unknown contaminants in the sequences, but the tool has also been used directly for EST clustering.

`ESTate`, developed by Slater for EST analysis (`fe.hgmp.mrc.ac.uk/gslater/estate.tar.gz`), offers clustering and database support. It clusters sequences in two stages: first, `precluster` uses finite-state machines and fast word-matching to compute the number of matching words of a specified length between all pairs of sequences in subquadratic time; then `estcluster` builds the clusters based on the scores thus generated.

C. Evaluating Clusters

Cluster validation refers to the procedures that evaluate the results of cluster analysis in a quantitative and objective way [16]. A large variety of such procedures have been defined. Moreover, as we discussed, EST clusters are often assessed on the basis of the contigs that could be formed from them. In the case of a simulation study, however, the situation is different, in that we know what the "correct" clustering is and can compare it to the clusterings produced by the various procedures. For that specific purpose (comparing two clusterings), we chose the Rand index [7], which has no particular bias with respect to clustering errors; the value of the index ranges from 1 (perfect match) down to 0.

D. The Experiments

We ran three successive experiments. A preliminary experiment, with reduced data, was used to gather data in order to set parameters and thresholds for the ensuing, "real" experiments. We then used these parameter and threshold settings to run two experiments. In the first experiment, we used relatively clean EST data (low-level suttering and small numbers of single base-pair errors) in order to evaluate how well each algorithm could perform and to compare their strengths. In the second experiment, we generated noisier data according to several data patterns in order to test our parameter and threshold settings and to assess the effect of error parameters on the quality of answers produced by each algorithm.

IV. Results and Discussion

A. Preliminary Experiment

We ran a preliminary experiment with just 10 datasets in order to adjust each program's clustering parameters and threshold values to optimize results. Table I shows the preliminary results obtained with the best parameter settings—results that indicate a clear linear ordering in terms of performance, with `BlastClust` at the bottom (much worse than all others), followed by `N2tool`, then `ESTate`, and finally the agglomerative clustering based on Smith-Waterman similarity scores.

TABLE I
SAMPLE MEANS AND STANDARD DEVIATIONS OF THE RAND INDEX.

	Smith-Waterman	`ESTate`	`N2tool`	`BlastClust`
Sample mean	0.965	0.926	0.828	0.255
Std deviation	0.029	0.025	0.025	0.052

B. First Experiment

In this experiment, only low-level stuttering and single-base error models were used to generate high-quality EST sequences. Table II shows the parameter settings used with `ESTSim` to generate 20 EST datasets.

TABLE II
PARAMETER SETTINGS FOR `ESTSIM`.

α	β	γ	ζ	ξ	K	λ	μ	ν	η	θ
0.005	16	0.02	1	1	10	10	10	0	20	0

Figure 1 shows the Rand index values for the various algorithms on all 20 datasets.

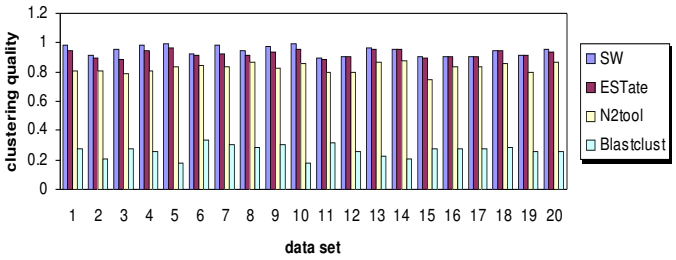

Fig. 1. The clustering quality of the four EST clustering algorithms on our 20 datasets.

Table III reports the mean and standard deviation of the Rand index for the results of each algorithm.

TABLE III
SAMPLE MEANS AND STANDARD DEVIATIONS OF THE RAND INDEX.

	Smith-Waterman	`ESTate`	`N2tool`	`BlastClust`
Sample mean	0.944	0.923	0.828	0.257
Std deviation	0.034	0.026	0.034	0.043

We used the Friedman rank sum test [17] at a 95% confidence interval to test the significance of the score differences between the four algorithms—against a null hypothesis positing that the differences are simply due to chance. First, we tested the scores of all four algorithms, obtaining a p-value (as computed by the R package) of $5 \cdot 10^{-12}$—and

thus firmly rejecting the null hypothesis. Since, however, it is quite obvious that BlastClust is much worse than the other three, we decided to run the same test on just the other three algorithms, now obtaining a p-value of $0.02 \cdot 10^{-6}$—another clear rejection of the null hypothesis. Finally, we ran the test only on the results of Smith-Waterman and ESTate, obtaining a p-value of $0.16 \cdot 10^{-3}$. Thus all distinctions are statistically significant at high levels of confidence.

Table III, Figure 1, and the results of the rank tests lead to some clear conclusions:

- While there is some variability from dataset to dataset, the performance of each clustering algorithm is remarkably consistent across all datasets—a fairly predictable finding in view of the high quality of the ESTs used.
- The ranking of the four algorithms is always the same: Smith-Waterman is the best, followed very closely by ESTate and more distantly by N2tool, while BlastClust is the worst in all respects.

In order to see the differences between Smith-Waterman and ESTate more clearly, Figure 2 presents a comparison of just these two algorithms.

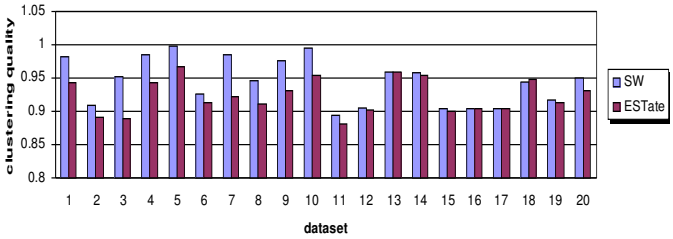

Fig. 2. Comparing the clustering quality for Smith-Waterman and ESTate on our 20 datasets.

Smith-Waterman dominates ESTate, but the differences are usually minor—less than 1% in most cases, with only 2 of the 20 test cases yielding differences above 5%. The standard deviation in scores for Smith-Waterman is 0.034 while that of ESTate is 0.026, indicating that Smith-Waterman is perhaps slightly less robust than ESTate. However, as discussed earlier, the Smith-Waterman dynamic program takes time quadratic in each of the number of ESTs and their lengths, while ESTate runs in subquadratic time. For sizable datasets, ESTate is thus preferable.

C. Second Experiment

This experiment is made of three parts, all aimed at understanding the effect of EST errors on the quality of clustering and the behavior of the algorithms. In these three parts, we no longer look at BlastClust, since its performance is so much worse than that of our other three approaches. In the first part of this experiment, we compared the three algorithms on datasets of different quality, using our previous threshold values and parameters. In the second part, we reran Smith-Waterman and ESTate with clustering parameters and threshold values adjusted for the dataset of poorer quality. Finally, we focused on polymerase decay, which (from informal experiments) appeared to be the parameter with the most damaging effect on the quality of clustering.

1) Comparing Smith-Waterman, ESTate, and N2tool with different data patterns: Table IV shows the parameters used to generate the two data patterns in this experiment—data in pattern 1 are of much higher quality than in pattern 2.

TABLE IV
PARAMETER SETTINGS FOR ESTSIM FOR TWO DATA PATTERNS.

Pattern	α	β	γ	ζ	ξ	K	λ	μ	ν	η	θ
#1	0.005	30	0.04	1	1	10	10	10	0	20	0
#2	0.010	60	0.08	2	1	10	10	10	0	0	0

The threshold values we used for the three programs are shown in Table V.

TABLE V
PARAMETER AND THRESHOLD SETTINGS FOR CLUSTERING.

	Parameters	Thresholds
Smith-Waterman	f(match)=1, f(mismatch)=-1, f(gap)=-2	118
ESTate	word_length=9, min_word_count=24	600
N2tool	screen_width=80	20

Table VI summarizes the mean values, standard deviation, and paired p-values (to test for the significance of the difference between the two patterns) for the three programs based on the two different data patterns.

TABLE VI
MEAN, STANDARD DEVIATION, AND P-VALUES FOR TWO DATA PATTERNS.

	Data Pattern #1 mean (std dev)	Data Pattern #2 mean (std dev)	p-value (paired)
Smith-Waterman	0.939 (0.031)	0.924 (0.027)	0.113
ESTate	0.913 (0.033)	0.892 (0.026)	0.036
N2tool	0.809 (0.035)	0.821 (0.024)	0.222

According to the p-values in Table VI, only ESTate reaches statistical significance (at the 95% level) in terms of its behavior on the two data patterns. All three algorithms in fact exhibited remarkable robustness in the face of a significantly worse data quality. (Curiously, N2tool even improved as the data worsened, although not to the point of rivalling the other two methods.)

2) Evaluating the effect of parameter values and thresholds: We then investigated the effect of lowering the clustering threshold—an action taken against the expectation that noisy datasets would yield lower similarity scores. The new parameters and thresholds are listed in Table VII.

We reran Smith-Waterman and ESTate on all 20 datasets with data pattern 2. Table VIII shows that both algorithms benefit from adjustment in parameters in almost all cases. The problem, of course, is to adjust such parameters automatically and at reasonable computing cost.

TABLE VII

ADJUSTED PARAMETER AND THRESHOLD VALUES IN SW AND ESTATE.

	Parameters	Thresholds
Smith-Waterman	f(match)=1, f(mismatch)=-1, f(gap)=-2	98
ESTate	word_length=9, min_word_count=24	500

TABLE VIII

RELATIVE PERFORMANCE OF SW AND ESTATE UNDER DATA PATTERN 2, COMPARED TO UNADJUSTED VERSIONS OF THEMSELVES.

	better	same	worse
Smith-Waterman	16	3	1
ESTate	19	0	1

3) The effect of polymerase decay.: During the course of experimentation, we noticed that the polymerase decay parameter ξ has a significant effect on clustering quality. For values of ξ larger than 1, clustering remained very poor no matter how we adjusted the clustering parameters and thresholds. In order to understand the effect, we used two data patterns with identical parameters, except for parameter ξ, which increases from 1 in the first pattern to 2 in the second. We then ran the Smith-Waterman algorithm on 10 datasets each with data pattern 1 and with data pattern 2, with good clustering in the first pattern and very poor clustering in the second pattern. Figure 3 shows the distributions of alignment scores for each data pattern—the figure is a histogram with the x-axis representing the score values. One can easily see that good alignment scores are much less common with the second data pattern than with the first.

When mRNA is reverse transcribed into cDNA, the DNA polymerase decays [18], with a resultant increase in the rate of single base errors. This decay has two phases: one is the gentle decay for the bulk of the read ($> 95\%$), which is modeled by parameter ξ in ESTSim, while the other is a very sharp decay that occurs at the end of the read and causes the error rate to shoot up, a decay modeled by parameter ζ in ESTSim. According to our experiments, the gentle polymerase decay has much greater effect on the EST sequence quality than the sharp final decay, presumably because, unlike the final decay, the gentle decay covers most of the read, so that even a very small change in the rate of decay can cause a significant change in the overall quality of the EST.

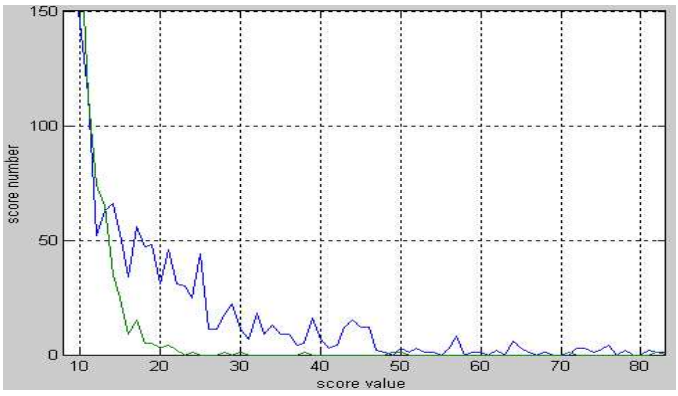

Fig. 3. The distributions of SW alignment scores under data patterns 1 and 2.

V. CONCLUSIONS

We evaluated four different EST clustering algorithms (based on different similarity measures and clustering methods) under a variety of conditions using simulated data. Some valuable points are summarized from these two experiments. First, clustering quality is statistically different not only with different similarity measures (BLAST vs. Smith-Waterman, for instance), but also with clustering methods. /bin/bash: a: command not found clustering quality depends on many factors including data quality as well as clustering threshold values and parameters. In some cases, improving data quality can help clustering quality. Sometimes, setting or adjusting clustering threshold values and parameters is even more critical than choosing clustering algorithms. Finally, investigating the error parameters may provide biologists heuristics to improve EST data quality during the process of EST generation. For example, we found that gentle polymerase decay has a significant effect on EST quality, yet that is a parameter that can be controlled (at some cost) on the bench.

Much work remains to be done. Most importantly, a continued investigation of the most important error parameters in EST generation is in order: we did not investigate ligation effects and contamination effects, for instance. EST clustering tools for now remain rather *ad hoc*: the parameters and thresholds are mostly set by hand—and the choice of method (if any) is left up to the user. Yet some characteristics of the data must help one select an approach and establish a good set of parameter values. Finally, the many heuristics embodied in the assembly tools need more rigorous evaluation—investigating at least some of these heuristics within a pure clustering context should indicate how the clustering phase of these assemblers could be improved.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation under grants IIS 01-21377, EF 03-31654, and DEB 01-20709 (the latter under a subcontract to the University of Texas at Austin), by the National Institutes of Health under grant 2R01GM056120-05A1 (under a subcontract to the University of Arizona), and by the IBM Corporation under a contract from the Defense Advanced Research Projects Agency.

REFERENCES

[1] M. Adams, M. Dubnick, A. Kerlavage, R. Moreno, J. Kelley, T. Utterback, J. Nagle, C. Fields, and J. Venter, "Sequence identification of 2,375 human brain genes," *Nature*, vol. 355, pp. 632–634, 1992.

[2] L. Hillier, G. Lennon, M. Becker, M. Bonaldo, B. Chiapelli, S. Chissoe, N. Dietrich, T. DuBuque, A. Favello, W. Gish, M. Hawkins, M. Hultman, T. Kucaba, M. Lacy, M. Le, N. Le, E. Mardis, B. Moore, M. Morris, J. Parsons, C. Prange, L. Rifkin, T. Rohlfing, K. Schellenberg, and M. Marra, "Generation and analysis of 280,000 human expressed sequence tags," *Genome Research*, vol. 6, no. 9, pp. 807–828, 1996.

[3] G. Vasmatzis, M. Essand, U. Brinkmann, B. Lee, and I. Pastan, "Discovery of three genes specifically expressed in human prostate by expressed sequence tag database analysis," *Proc. Nat'l Acad. Sci., USA*, vol. 95, no. 1, pp. 300–304, 1998.

[4] T. Wolfsberg, J. McEntyre, and G. Schuler, "Guide to the draft human genome," *Nature*, vol. 409, pp. 824–826, 2001.

[5] A. Mironov, J. Fickett, and M. Gelfand, "Frequent alternative splicing of human genes," *Genome Research*, vol. 9, no. 12, pp. 1288–1293, 1999.

[6] 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.

[7] W. Rand, "Objective criteria for evaluation of clustering methods," *J. American Statistical Assoc.*, vol. 66, no. 336, pp. 846–850, 1971.

[8] M. Pop and D. Kosack, "Using the TIGR assembler in shotgun sequencing projects," *Methods Mol. Biol.*, pp. 279–294, 2004.

[9] X. Huang and A. Madan, "Contig assembly program version 3 (CAP3): A DNA sequence assembly program," *Genome Research*, vol. 9, pp. 868–877, 1999.

[10] G. Schuler, "Pieces of the puzzle: expressed sequence tags and the catalog of human genes," *J. Mol. Med.*, vol. 75, pp. 694–698, 1997.

[11] J. Aaronson, B. Eckman, R. Blevins, J. Borkowski, J. Myerson, S. Imran, and K. Elliston, "Toward the development of a gene index to the human genome: An assessment of the nature of high-throughout EST sequence data," *Genome Research*, vol. 6, no. 9, pp. 829–845, 1996.

[12] H. Pospisil, A. Herrmann, R. Bortfeldt, and J. Reich, "EASED: Extended alternatively spliced est database," *Nucl. Acids Res.*, vol. 32, pp. D70–D74, 2004.

[13] J. Burke, D. Davison, and W. Hide, "D2_cluster: A validated method for clustering EST and full-length cDNA sequences," *Genome Research*, vol. 9, no. 11, pp. 1135–1142, 1999.

[14] T. Smith and M. Waterman, "Identification of common molecular subsequences," *J. Mol. Biol.*, vol. 147, no. 1, pp. 195–197, 1981.

[15] J. Parsons, "Improved tools for DNA comparison and clustering," *Bioinformatics*, vol. 11, pp. 603–613, 1995.

[16] A. Jain and R. Dubes, *Algorithms for Clustering Data*. Prentice-Hall, Englewood Cliffs, NK, 1988.

[17] S. Siegel and N. Castellan, *Nonparametric Statistics for the Behavioral Sciences (2nd ed.)*. McGraw-Hill, NY, 1988.

[18] B. Maier, D. Bensimon, and V. Croquette, "Replication by a single DNA polymerase of a stretched single-stranded DNA," *Proc. Nat'l Acad. Sci., USA*, vol. 97, no. 22, pp. 12 002–12 007, 2000.