A Method for Evaluating the Quality of String Dissimilarity Measures and Clustering Algorithms for EST Clustering

Judith Zimmermann* Zsuzsanna Lipták† Scott Hazelhurst‡

January 15, 2004

Abstract

We present a method for evaluating the suitability of different string dissimilarity measures and clustering algorithms for EST clustering, one of the main techniques used in transcriptome projects. Our method consists of first generating simulated ESTs according to user-specified parameters, and then evaluating the quality of clusterings produced when different dissimilarity measures and different clustering algorithms are used. We have implemented two tools for this purpose: *(i)* ESTSim (*EST Simulator*), which generates simulated EST sequences from mRNAs/cDNAs according to user-specified parameters, and *(ii)* ECLEST (*Evaluator for CLusterings of ESTs*), which computes and evaluates a clustering of a set of input ESTs, where the dissimilarity measure, the clustering algorithm, and the clustering validity index can be specified independently.

We demonstrate the method on a sample set of 699 cDNAs taken from a public mammalian gene collection: We generated approximately 16,000 simulated ESTs from this set according to parameters that follow the guidelines laid down by NCBI, and compared the clusterings produced using five different dissimilarity measures, while fixing the clustering algorithm and clustering validity index. We then repeated the experiment with higher error parameters. We have been able to derive statistically significant results from this study, where we were able to show, for example, that with a single linkage clustering algorithm, some subword-based dissimilarity measures produce output comparable to alignment-based ones, in a significant number of cases.

We argue that the method presented, and in particular, the use of simulated data, are well suited for this type of study because they allow for quality evaluation w.r.t. a known ideal clustering. Our method has important applications in the large number of studies on gene expression, alternative splicing, and gene discovery currently under way, where there is a pressing need for fast but correct clustering of increasingly large sets of ESTs.

Key words string similarity and dissimilarity measures, EST clustering, transcriptome, simulated data, benchmarks

Corresponding Author

Scott Hazelhurst, School of Computer Science, University of the Witwatersrand, Private Bag 3, 2050 Wits, South Africa.
Email: scott@cs.wits.ac.za; Phone: +27 11 717 6181; Fax: +27 11 717-6199

*Research Group 'Algorithms, Data Structures, and Applications', Institute of Theoretical Computer Science, ETH Zurich, CH-8092 Zurich. judithz@student.ethz.ch

†Universität Bielefeld,Technische Fakultät, AG Genominformatik, 33594 Bielefeld, Germany. zsuzsa@CeBiTec.Uni-Bielefeld.DE . Most of this work was done while Zs.L. was working in the Research Group 'Algorithms, Data Structures, and Applications', Institute of Theoretical Computer Science, ETH Zurich, and at the South African National Institute of Bioinformatics (SANBI), Cape Town.

‡School of Computer Science, University of the Witwatersrand, Johannesburg, Private Bag 3, 2050 Wits, South Africa. scott@cs.wits.ac.za . Partially supported by SA National Research Foundation (GUN2053410)

1 Introduction

ESTs (Expressed Sequence Tags) are short (usually between 300 and 500 bp) DNA sequences derived from mRNA transcripts. ESTs have been widely used during the past two decades (e.g. in gene expression projects, for gene discovery, for discovery of products of alternative splicing, and even for estimating the number of human genes). With mRNA sequencing still being a biochemically difficult and costly process, ESTs, which can be produced cheaply and in a high-throughput manner, remain the primary access to the cell's transcriptome. ESTs are derived from mRNAs in a process that involves capturing mRNAs present in the cell, reverse transcription, cloning, and high-throughput sequencing. In addition to individual EST projects, many researchers make use of the large number of ESTs publicly available in EST databases. In either case, one is faced with a large set (up to million or more in size) of short DNA sequences (ESTs) that needs to be interpreted. This leads to the problem of *EST Clustering*, which has the goal of producing a partition (clustering) of the original set such that each cluster corresponds to a gene, i.e., two ESTs are members of the same cluster if and only if they have been derived from mRNAs that are transcripts of the same gene. See [15] for more background on ESTs.

Several software packages exist that perform EST clustering; among the most widely used are UniGene [3], TIGR [31], and StackPack [7], which all make available a database of EST clusters. The clustering packages usually consist of several steps that include contamination and repeat masking, clustering, and consensus computation. In this paper, we focus on the *clustering step*. For the clustering, different algorithms are used, both with respect to the *clustering algorithms*, and to the criteria (*similarity or dissimilarity measures*[1]) according to which sequences are clustered together. In the following, we shall refer to the two sub-steps of the string dissimilarity measure and the clustering algorithm as defining the *clustering method*. We will give exact definitions in Sections 2 and 3.

In spite of the variety of EST clustering methods, to the best of our knowledge, no rigorous technique has been put forward for evaluating which one of the existing or newly suggested ones is best suited. Instead, in most studies, it is not the underlying clustering algorithms and sequence dissimilarity measures that are being evaluated but the output: The clustering produced is evaluated either according to expert knowledge, or according to how well it corresponds to the output of some software that has proved to be good in the past. While we believe that the former is a valid and in fact, invaluable, method of evaluation, it is in most cases not feasible. The latter, on the other hand, hinders innovation for intrinsic reasons. Moreover, since ESTs are produced according to different processes, they can have different properties (such as the distribution of errors of different types), and thus, different algorithms may be more or less appropriate depending on the type of input data.

[1] For simplicity, we will often refer to both types as *dissimilarity measures*.

1.1 Contributions

In the present paper, we propose a rigorous method for evaluating the suitability of string dissimilarity measures and clustering algorithms for EST clustering, depending on the characteristics of the input data. We distinguish between four different components: (1) the input data, (2) the string dissimilarity measure, (3) the clustering algorithm, and (4) the clustering validity index, according to which the quality of the output is judged.

As *input data*, we generate *simulated* ESTs from real cDNAs or full-length mRNAs according to specifiable error parameters, using our dedicated tool ESTSim (*EST Simulator*). ESTSim generates simulated ESTs from input cDNAs or mRNAs according to carefully chosen user-specifiable parameters, and outputs EST-like sequences, identifying which input sequence they were derived from.

We then compute a partition of the set of simulated ESTs with our tool ECLEST (*Evaluator for CLusterings of ESTs*). ECLEST computes clusterings of a set of input sequences, using specified *string dissimilarity measures* and *clustering algorithms*, where both components can be chosen independently. It then computes a score of the resulting partition by comparing it to the ideal partition, using a specified *validity index*. Again, the validity index can be chosen independently of the other two components. ECLEST has a modular architecture, which allows different string dissimilarity measures, clustering algorithms, and clustering validity indices to be used.

We believe that using simulated data rather than real ESTs is better suited for this type of study, for two reasons: First, for real ESTs, the ideal clustering is never known and can at best be approximated by, for example, aligning ESTs to the genome, or by relying on annotation information, or both. Therefore, any evaluation will be subject to possible errors made during the experimental phase. Second, simulated data allow for careful tuning of parameters, such as average sequence length or single base error distribution, and thus for estimating the impact of an individual parameter on the methods used: This is, of course, impossible with real data.

We demonstrate the fitness of our method by presenting the results of a test study we carried out on 699 cDNA sequences from a mammalian gene collection. From these, we derived approximately 16,000 simulated ESTs and clustered them using five different string dissimilarity measures and a single linkage clustering algorithm, computing the clustering quality scores using the Rand Index (for details, see Section 6). We ran two sets of tests, where we varied the error types, in the first simulating error types and levels as laid down in the guidelines of NCBI, and in the second, increasing these error levels. We were able to draw statistically significant conclusions as to the quality of clusterings produced using these different string dissimilarity measures.

1.2 Related work

Most literature on *EST clustering* proposes a particular EST clustering algorithm, and validates the clustering computed by comparing it to similar tools, or by expert examination of the outcome. This is the case, for example, for literature on well known clustering tools such as UniGene [3], TIGR [31], and StackPack [7]. In 1999, Hartuv *et al.* [13] proposed an EST clustering algorithm that employs a string dissimilarity measure and clustering algorithm which both differ from the ones commonly used. The validation of the clusterings is made both with simulated and with real input data: For simulated data, the result is compared to the known ideal clustering, while for real data, it is compared to a very carefully generated clustering, which is judged correct by the authors. The authors do not claim that their algorithm produces the best clustering, but that it allows for a large speedup as a preprocessing step to a more careful but more costly EST clustering method. Recently, two new EST clustering algorithms using suffix tree-type data structures were suggested: In [24], Malde *et al.* use suffix arrays, and the clusterings computed are compared to the output of other clustering tools. Kalyanaraman *et al.* [20], on the other hand, validate their clusterings by comparing them to clusterings computed by aligning the ESTs to the genome. Burke *et al.* [5] compare the results of d^2-cluster, the clustering method used by StackPack, to results produced by UniGene. It is shown that d^2-cluster is more sensitive, and this claim is supported by *(i)* examining the resulting clusters using biological expertise, and *(ii)* by deriving upper bounds on the probability of incorrectly clustering sequences; in addition, a comparison with the Smith-Waterman algorithm is performed, with the explicit assumption that it produces correct results. A large-scale comparison of four EST assembly tools was conducted by Liang *et al.* in [22], using both real and simulated ESTs; the study includes a comprehensive discussion of the exact behaviour of the four programs and their sensitivity to different parameters. The focus there, however, is on the assembled sequences (tentative consensi) rather than on the clusters produced.

Some approaches to *protein clustering*, e.g [4, 21], implicitly assume one clustering algorithm to be better than others, and then use that as a benchmark. Several studies have been conducted on clustering for *gene expression*. While Wicker *et al.* [36] concentrate on finding the optimal number of clusters, Yeung *et al.* [37] develop an internal clustering validation index, which they use to evaluate the quality of clusterings produced by three different clustering algorithms, both on real and on simulated data. This validation index is a (simpler) variation of one of the four different indices of internal clustering validation used by Datta and Datta [8], who compare six clustering methods for microarray data. This study is the closest to the present one in that it aims at comparing different clustering methods, without claiming that one particular validation or clustering method is optimal. Instead, after having demonstrated that the outcome varies significantly depending on the clustering method used, the paper "offers some guidelines in the choice of a clustering technique to be used in connection with a particular microarray data set."

The review by Quackenbush *et al.* [32] discusses relevant issues in expression analysis with microarray data, including an overview of different clustering algorithms and distance measures, with the focus on the former; several of these are compared on a simulated data set.

One example of a large-scale comparison of two *string (dis)similarity measures* is work by Nash *et al.* [27]. Here, the Smith-Waterman algorithm and BLAST are compared for pairwise alignment of protein sequences, and it is shown that in some cases, BLAST finds matches that Smith-Waterman does not. Even though the implicit assumption is made that the Smith-Waterman algorithm produces the 'correct' answer, no method is supplied for evaluating the quality of the results.

Our methodology is most similar to that used in *phylogenetic studies*, where a phylogenetic tree is synthetically generated according to some evolutionary model, and then phylogenetic algorithms are evaluated according to how well they can reconstruct the known tree from the leaf data, see e.g. [25]. We are not aware of any study that attempts a rigorous comparison of EST clustering methods, separating the effects of the string dissimilarity measure and the clustering algorithm.

2 Terminology

ESTs are produced in a laboratory process. The mRNAs present in the cell (the *transcriptome*) are extracted, reverse transcribed, inserted into vectors, cloned, and then sequenced. The resulting ESTs are thus approximate substrings of the original mRNAs (save the mRNA/cDNA substitution of T for U). These mRNAs, having undergone the process of splicing, are related to the original genes in the known way: They can be seen as the concatenation of certain substrings of the gene (the exons) such that the exon order in the gene is preserved. The phenomenon of alternative splicing means that ESTs that have been derived from two different mRNAs of the same gene do not need to have overlaps; instead, they can have similar substrings which need not be at the ends, or even be contiguous. In addition, due to the high-throughput manner in which ESTs are produced, ESTs are more error-prone than, for instance, sequences for shotgun sequencing. We will discuss characteristics of ESTs in Section 4. Let $\Sigma = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ denote the set of bases[2]. Let $G \subseteq \Sigma^+$ be the set of (not necessarily known) genes.

> **Definition 1 (EST Clustering)** *Given a finite set S of strings (ESTs) over Σ, find a partition $\mathcal{C} = C_1, \ldots, C_k$ of S such that there exist strings (genes) $g_1, \ldots, g_k \in G$ where, for all $1 \leq i \leq k, s \in S : (s \in C_i \iff s$ has been derived from $g_i)$. We refer to the sets C_i as* clusters.

Hereby, a *partition* of a set S is a collection of disjoint subsets whose union equals S. Note that Definition 1 does not require that the genes g_i be specified. The right side of the equivalence is kept in informal

[2]In fact, ESTs are strings over the alphabet $\Sigma \cup \{\mathrm{N}, \mathrm{X}\}$, where characters N and X are interpreted either as any of the four characters from Σ or as strings over Σ^+; usually, the presence of an N is an artifact of the sequencing process, while X's are inserted during masking. For the sake of simplicity, we omit this from our definition.

terms because the question of how to capture formally the (physical) process of an EST sequence having been derived from a gene, is one of the topics of this paper.

We denote the set of non-negative integers by $\mathbb{N}$. For a set X, let $|X|$ denote its cardinality, and $\binom{X}{2}$ the set of its subsets with cardinality 2. For a partition $\mathcal{C}$ of S and $s \in S$, denote by $\mathcal{C}(s)$ the unique cluster $C \in \mathcal{C}$ such that $s \in C$. Let $\mathcal{C}$ and $\mathcal{D}$ be two partitions of the same set S. We call $\mathcal{C}$ a *(proper) refinement* of $\mathcal{D}$ if $|\mathcal{C}| > |\mathcal{D}|$, and for all $s,t \in S$: if $\mathcal{C}(s) = \mathcal{C}(t)$, then $\mathcal{D}(s) = \mathcal{D}(t)$.

For a string $s = s_1 \dots s_n$, its length is denoted by $|s| = n$. If $s = s_1 \dots s_n$ and $t = t_1 \dots t_m$ are strings over the same alphabet, then t is a *substring* of s, denoted $t \sqsubseteq s$, if there is $1 \leq i \leq n - m + 1$ such that $t = s_i \dots s_{i+m-1}$. Substrings of a string are often referred to as *(sub)words*. For strings w, s with $|w| \leq |s|$, let $freq_s(w) := |\{i : w = s_i \dots s_{i+|w|-1}\}|$, the number of times w occurs in s. Note that this definition allows overlapping occurrences of w. Furthermore, let $occ_s(w) = 1$ if w is a substring of s, and 0 otherwise.

In the absence of genes to be compared with, some notion of similarity or dissimilarity is used to cluster individual sequences together. This requires a function, or *dissimalirity measure* of pairs of strings $D : \Sigma^+ \times \Sigma^+ \to \mathbb{R}$. If D is symmetric, obeys the triangle inequality, is positive, and $D(s,t) = 0$ implies $s = t$, then it is called a metric; if $D(s,t) = 0$ can hold for some $s \neq t$, then it is a pseudo-metric. For string dissimilarity measures, often neither is the case. Given a dissimilarity measure D and a positive integer m (the window size), define $\hat{D} : \Sigma^+ \times \Sigma^+ \to \mathbb{R}$ by $\widehat{D}(s,t) := \min\{D(s',t') : s' \sqsubseteq s, t' \sqsubseteq t, |s'| = |t'| = m\}$. $\widehat{D}(s,t)$ is the minimum dissimilarity of any pair of substrings (windows) of s and t.

3 Clustering Algorithms, Dissimilarity Measures, and Validity Indices

3.1 Clustering Algorithms

Data clustering is the task of grouping together a set of objects into subgroups according to some property or properties. There are a large variety of clustering algorithms and a fair amount of terminological inconsistency in the literature. See [18], whose terminology we follow, for an introduction.

For EST clustering, most often, hierarchical clustering algorithms are used. A sequence of increasingly fine partitions of the data is produced, starting from the whole set as one cluster, where each partition is a refinement of the previous one. The process stops when a partition is found that is fine enough according to some previously defined criterion. Most EST clustering algorithms are *single linkage* (nearest neighbour, transitive closure): Two objects x and y are clustered together if there is a finite sequence $x = x_1, x_2, \dots, x_{k-1}, x_k = y$ such that for all $1 \leq i < k$, $D(x_i, x_{i+1}) < \theta$, for some threshold θ. If the threshold is set beforehand, single linkage can be viewed as a partitional clustering algorithm; in its general form, it is a hierarchical clustering algorithm where, usually, different levels correspond to

different values of θ. Other types of clustering algorithms, such as *complete linkage* or *k-means*, are not commonly used for EST clustering.

Clustering can *seeded* or *unseeded*. In seeded clustering, the number of clusters is known beforehand, and a member of each cluster (the *seed*) is supplied as part of the input: in EST clustering, this is typically a full-length mRNA. In unseeded clustering, no additional information is used and, in particular, the number of clusters is unknown.

Of the EST clustering methods mentioned in the Introduction, UniGene uses a hierarchical seeded clustering algorithm, where mRNAs are used as seeds, and different hierarchies represent different types of merging stages: e.g. edges that connect two ESTs are judged less reliable than those connecting an EST to an mRNA. TIGR uses single linkage clustering and produces a tentative consensus for each cluster. TGICL [29], a recent enhancement of TIGR, also uses single linkage, and so do both [20] and [24]. StackPack uses single linkage and different hierarchy levels, where the lowest (least confident) level is "clone linking": Two clusters are merged if they contain two end reads of the same clone. Hartuv *et al.* use the HCS clustering algorithm: highly connected (i.e., particularly dense) subgraphs in a threshold graph are output as clusters.

The prevalence of single linkage is due to the fact that EST clustering constitutes a type of local alignment to an unknown reference sequence (the gene). Thus, the traditional drawback of single linkage, namely that it creates 'elongated clusters,' is a desired effect in EST clustering.

3.2 String Dissimilarity Measures

Two different, but closely related, concepts are in common use in the literature on strings: That of *similarity* and that of *dissimilarity*. Both are usually represented by a function $D : \Sigma^+ \times \Sigma^+ \to \mathbb{R}$, but the former type takes on higher values the closer (more similar) two strings are, while the latter decreases in this case. For the sake of consistency, we refer to all measures as dissimilarity measures, and transform the functions accordingly where necessary.

String dissimilarity measures employed in approximate string matching include those that are based on *alignment* and those that are based on *subword* comparisons. Other approaches exist, such as information theoretic ones (as cited in the review [35]), but are not commonly used. In approximate string matching of biological sequences, the most widely employed string dissimilarity measure is BLAST [2]. For an overview of approximate string matching, see [12, 28].

Alignment and edit distance. The *Levenshtein distance* or *unit cost edit distance* of strings s, t is the minimum length of a sequence of edit operations transforming s into t, where admissible edit operations are substitutions, insertions, and deletions of characters. Enhanced cost functions include different cost attached to different types of operations, and the cost of an operation depending on the characters in-

volved (e.g. substitution of a for c may have different cost from substitution of b for c). The Levenshtein distance is a metric. An optimal sequence of such transformations can be visualised as an *alignment* of the two strings, by placing them under each other character by character, possibly inserting free spaces (*gaps*), such that there are no two gaps in one column. By assigning penalties for mismatches or gaps in an alignment, we can compute an alignment score, a measure of *similarity*. More enhanced scoring functions include affine or more general types of gap penalty functions: here, the score given to a character aligned with a gap may depend on the number of consecutive gaps before. The alignment score of two strings is the score of an optimal alignment. To find high-similarity substrings in two strings, a *local alignment* is sought. The definition differs in that the alignment need not continue to the ends of the strings. Another variant is *end-space free global alignment*, where gaps at the ends of either string have zero cost. Since we discuss measures of *dissimilarity*, we will use alignment scoring functions which assign positive values to mismatches and gaps, and negative values to matches, and will refer to such functions as *penalty functions*.

Computing an optimal local alignment score can be done with the well-known dynamic programming algorithm of Smith-Waterman. To improve performance, heuristic algorithms like BLAST [2] and FASTA [23] are used, which employ filtering techniques to isolate areas where matches are likely to happen. Most biological clustering methods use BLAST as the underlying dissimilarity measure, among them UniGene [3] and TIGR [31].

Word frequency counts. Fix a subword size $q \in \mathbb{N}$. For 'small' q, substrings of length q have been referred to as *q-words*, *q-mers*, or *q-grams*. Any string $s \in \Sigma^+$ can be mapped to its q-gram vector, or word-frequency vector $freq_s \in \mathbb{N}^{|\Sigma|^q}$, whose w'th entry is just $freq_s(w)$. The *q-gram distance* [34] of two strings is $D_{q-\text{gram}}(s,t) := \sum_{w\in\Sigma^q} |freq_s(w) - freq_t(w)|$. This is the L_1-distance, or Manhattan-distance, of the vectors $freq_s$ and $freq_t$. However, $D_{q-\text{gram}}$ is only a pseudo-metric, since $D_{q-\text{gram}}(s,t) = 0$ for all s,t with identical word frequency vectors. Ideas derived from the q-gram distance have been implemented for database search in the QUASAR project [6], and also for clustering ESTs.

Another dissimilarity measure using the word frequency vectors is referred to as d^2, [33]: $D_{d^2,q}(s,t) := \sum_{w\in\Sigma^q}(freq_s(w) - freq_t(w))^2$. This is the squared L_2-distance, or Euclidean distance, of the frequency vectors, hence the name. Note that $\sqrt{D_{d^2,q}}$ is a pseudo-metric, but $D_{d^2,q}$ is not, because it does not obey the triangle inequality. In the more general form, fix a lower and an upper bound l,u on the word size, and set $D_{d^2}(s,t) := \sum_{q=l}^{u} D_{d^2,q}(s,t)$. For EST-clustering, experimental evidence has shown that fixing q is satisfactory, and commonly $D_{d^2,6}$ is used. However, it should be noted that this measure is, in theory, not a good approximation of edit distance: There are, for instance, example sequences s,t of length 100 such that $D_{d^2,6}(s,t) = 0$ but the unit edit distance of s and t is 30. The StackPack EST clustering package uses a variant of $\hat{D}_{d^2,6}$ with window size $m = 100$ (see Section 3.2).

Fingerprints and subword occurrences. Fingerprints have been employed in computational biology for physical mapping of DNA (see [30, Chapter 3]), and, more recently, for EST clustering in [13]. Given a finite set $P \subseteq \Sigma^+$ of words, and a string $s \in \Sigma^+$, the *fingerprint* of s is the set $P \cap \{t : t \sqsubseteq s\}$, or its representation as a Boolean vector in $\{0,1\}^{|P|}$ whose i'th entry is $occ_s(p_i)$ for some enumeration $p_1, \ldots, p_n$ of P. Extending P to all Σ^q, we get the Boolean vector $occ_s \in \{0,1\}^{|\Sigma|^q}$ with w'th entry $occ_s(w)$, and again, any distance measure on Boolean vectors can be applied to define a dissimilarity measure on strings, such as the Hamming distance: $D_{q-\text{occurrence}}(s,t) := \sum_{w \in \Sigma^q} |occ_s(w) - occ_t(w)|$.

Another simple dissimilarity measure using subword occurrences is a Boolean function we refer to as *common word*, which assigns to two sequences distance 0 if they share a subword of fixed size K, and 1 otherwise, formally $D_{\text{cword}}(s,t) := 0$ if there exists $w, |w| = K$, $w \sqsubseteq s,t$, and 1 otherwise. Since typically, K is 'large' (say, around 20), we use a different variable from q, which is usually thought of as 'small', typically under 10. Note that this dissimilarity measure is symmetric, but does not obey the triangle inequality, and can have value 0 for non-identical strings; thus, it is again not even a pseudo-metric. The advantage of the common word function is that clustering can be implemented in linear time, at least for reasonable size K. The clustering is in general too coarse, but provided K is chosen well, a better clustering can be produced by refining it, and is thus a good first phase for a clustering algorithm. It is used, e.g. in [20], for determining the order in which to compare the sequences (done with an alignment-based measure); in [24], a measure is employed that combines several non-contiguous common words into one dissimilarity score.

We define a further dissimilarity measure, *Boolean q-grams*, which is an extension of D_{cword}, and in some sense complementary to $D_{q-\text{occurrence}}$: It counts the number of common q-grams; however, as a dissimilarity measure, we define it to be negative for each of these common q-grams: $D_{\text{B } q\text{-gram}}(s,t) := -\sum_{w \in \Sigma^q} occ_s(w) \cdot occ_t(w)$. So, $D_{\text{B } q\text{-gram}}$ is the negative scalar product of the Boolean vectors defined above. Again, this dissimilarity measure is not a metric, and not even a pseudo-metric.

Sliding windows. In EST clustering, local regions of high similarity are sought. This is reflected by, e.g. computing local alignments. For subword-based measures instead, a 'sliding window' of a fixed size m is used. Pairs of subsequences of size m are compared; if all such pairs are compared, this yields $\hat{D}_m(s,t)$. Note that even if D is a metric, $\hat{D}$ in general is not, since the triangle inequality no longer holds.

To increase time efficiency, sometimes not all pairs of windows are compared. For instance, the StackPack EST clustering tool uses a variant of $D_{d^2,6}$, where all windows in one sequence are compared with every k'th window in the other sequence: $D_{d^2\,\text{asym}}(s,t) := \min\{D_{d^2,6}(s', t^{(i)}) : |s'| = ms' \sqsubseteq s, i = 0, \ldots, \lfloor |t|/k \rfloor\}$, where k is the skip size, and $t^{(i)} := t_{i \cdot k+1} \ldots t_{\min(i \cdot k+m, |t|)}$ is the substring of t starting at position $i \cdot k + 1$. Note that this measure is not symmetric, which is why we refer to it as *asymmetric* d^2. In contrast, we will refer to $D_{d^2\,\text{sym}} := \hat{D}_{d^2}$, which compares all pairs of windows, as *symmetric* d^2.

3.3 Clustering Quality Evaluation

The quality evaluation of a clustering algorithm is referred to in the literature as *cluster validity analysis*, *clustering evaluation*, or *goodness of fit*. A large number of different methods are in use. They can be grouped into methods of *internal* and *external* assessment [18]: Internal assessment methods validate some criterion of internal consistency, while external methods compare the resulting clusters to an ideal solution. We only consider methods of *external* assessment. A score of the goodness of fit of the two clusterings is computed, which is referred to as a *validity index*. For comparing partitions, commonly used validity indices include the Rand Index, the Jaccard Index, and the Minkowski Index. For an overview of cluster validity indices, see [9, 17]. Less formal validation techniques include counting the number of exactly matching clusters or comparing the number of singleton clusters (as in [24] and [5] resp.). Two measures commonly employed in the biological literature are *sensitivity* and *specificity*.

Let S be the ground set of size $|S| = n$, the two partitions under consideration $\mathcal{C} = \{C_1, \ldots, C_k\}$ and $\mathcal{D} = \{D_1, \ldots, D_\ell\}$. All indices mentioned above are functions of the number of unordered pairs of elements that were "treated alike" and "treated differently" by the two clusterings. Set

$$a_1 = \left|\left\{\{s,t\} \in \tbinom{S}{2} : \mathcal{C}(s) = \mathcal{C}(t) \text{ and } \mathcal{D}(s) = \mathcal{D}(t)\right\}\right|, \quad a_2 = \left|\left\{\{s,t\} \in \tbinom{S}{2} : \mathcal{C}(s) \neq \mathcal{C}(t) \text{ and } \mathcal{D}(s) \neq \mathcal{D}(t)\right\}\right|,$$
$$d_1 = \left|\left\{\{s,t\} \in \tbinom{S}{2} : \mathcal{C}(s) = \mathcal{C}(t) \text{ and } \mathcal{D}(s) \neq \mathcal{D}(t)\right\}\right|, \quad d_2 = \left|\left\{\{s,t\} \in \tbinom{S}{2} : \mathcal{C}(s) \neq \mathcal{C}(t) \text{ and } \mathcal{D}(s) = \mathcal{D}(t)\right\}\right|.$$

Further, we set $a = a_1 + a_2$ and $d = d_1 + d_2$. Observe that a is the number of agreements and d the number of disagreements between the two clusterings. In the biological literature, it is customary to speak of 'true/false positives/negatives.' If we view $\mathcal{C}$ as the correct clustering, then a_1 is the number of true positives, a_2 the number of true negatives, d_1 the number of false negatives, and d_2 the number of false positives. We give the definitions of the above validity indices in the table below (transforming the definition of the Minkowski Index into one employing the values introduced above).

Rand Index	$\frac{a}{\binom{n}{2}}$	Minkowski Index	$\left(\frac{2d}{2(a_1+d_1)+n}\right)^{\frac{1}{2}}$	Sensitivity	$\frac{a_1}{a_1+d_1}$
Jaccard Index	$\frac{a_1}{a_1+d}$			Specificity	$\frac{a_1}{a_1+d_2}$

The Rand Index can also be corrected for chance by the expected difference to a randomly chosen clustering with the given number of clusters, see [16]. We have decided to use the uncorrected Rand Index because it is not realistic to assume that the number of clusters is known; on the other hand, correction using the expected score of a random clustering would be minimal, and might not be worth the additional computational effort.

4 ESTSim: Creating Benchmarks of Simulated EST Sets

4.1 Using simulated data for benchmarking

We propose the use of synthetic data for evaluation. Alternative approaches are:

1. The output of a clustering algorithm is examined by experts, who carefully analyse and possibly modify it based on their expert knowledge, before giving it their imprimatur. These sort of data sets are ideal to use for benchmarking, but are rare and very difficult to obtain.

2. Comparison to existing algorithms/software packages. Although there are very pragmatic reasons for adopting this approach, we argue that it inherently skews the results unless there is compelling biological evidence that justifies the result. An additional factor that should be borne in mind is that, considering ESTs are relatively error-prone and can differ significantly, depending on the biochemical production process, there is no guarantee that one measure and clustering algorithm will be superior in all circumstances.

We believe that artificial but realistic data sets should be used as complementary benchmarks to the first approach, and have designed a tool, ESTSim (EST Simulator) to produce these benchmarks. The objective of ESTSim is to produce large sets of artificial but realistic data for testing the effectiveness of different distance measures and clustering algorithms used in clustering DNA (or related sequences). The use of artificial test data enables us to produce data with a range of different error models. Thus, the effect of different error models on the effectiveness of the use of certain EST clustering methods can be tested precisely, which would be impossible using real data.

When using simulated data, the question of validation arises: to draw valid conclusions about real life data, it is vital that the outcome of the simulation be realistic. Two possibilities exist for proving that the data produced are realistic. First, comparing it with real data, and second, showing that the model used in generating it is close to reality. While we feel that the former approach is stronger, it involves certain difficulties: Certain characteristics of the real data are not known, e.g. the genes that the ESTs have been derived from. Furthermore, it is, of course, possible to compare statistics of observable characteristics of the data, such as nucleotide frequency, length etc. However, great care must be taken not to use characteristics that have been considered when generating the data, and instead, to choose characteristics that are independent random variables of those that have been used in the model. One possible candidate for artificial ESTs is pairwise overlap length, whose statistical distribution we are in the process of validating. However, we feel that much is known about the properties of ESTs, including error types and distributions, and that our tool has been very carefully designed to accommodate these.[3]

[3]Our arguments for using artificial data are very similar to those used in [26]: There, a system for generating simulated shotgun sequences was introduced, which was used for training and testing sequence assembly algorithms.

4.2 Details of the EST Simulator

ESTSim creates artificial ESTs from a set of given cDNA sequences using criteria specified by the user. The artificial creation of ESTs in this way will lead to the creation of an EST set whose exact final clustering is known, because each artificial EST carries an identifier of the sequence it was derived from. So, when testing an EST clustering method, the output can be evaluated by comparing it to the known ideal clustering. The approach of ESTSim is similar to that of GenFrag [10], though ESTSim is specifically tailored to EST data and supports more sophisticated error models. Here, we give a brief overview; full details can be found in [14].

EST sequences are submitted in bulk and are single, unverified runs, quality is on average low. We simulate the production of ESTs from cDNAs or any genomic-like data with a variety of models that simulate what happens during the production process. We do not aim to produce one model for ESTs since the biological processes used by different labs at different times manifest different types of errors and error rates. Rather, our tool can generate different types of data with user-specified error models.

1. Given an input sequence (e.g. mRNA or cDNA), it is split into fragments. Splitting can either be done at random, with user-specified parameters, or at user-defined 'restriction' sites.
2. Each fragment is then copied a user-specified number of times.
3. Alternatively, the user can specify in the splitting step (step 1) how many times on average each base should appear in a fragment.
4. Each fragment thus obtained is then mutated with user-specified fault models.

Errors modelled. Real ESTs have various sources of errors, including: contaminants (vector, rRNA, mitRNA, possibly other species, genomic sequence); repeat sequences (simple repeats, complex repeats); base pair errors; frameshift errors; chimeras resulting from artificial ligation of unrelated ESTs; and stutters.

ESTSim has been built on an understanding of the types of errors that are produced in the laboratory. The methodology used was to try to understand the biological processes, and with the aid of a biologist draw the types of error curves. We then found convenient mathematical functions that simulated these curves and that could be easily implemented. Since contaminant and repeat masking are typically done during EST clustering in a preprocessing step, we do not model these. The errors modelled are:

- *Single base errors:* First, single bases may be read incorrectly due to random noise. Second, polymerase decay may cause an increase in the rate of errors as the EST is read (there is gentle decay for the bulk of the EST followed by a very rapid decay at the end). Finally, interference from the primer makes the beginning parts of reads particularly unstable. If an error does happen, there are four possible events. A base can be arbitrarily changed, deleted or inserted, or an N can be inserted. The probabilities of these events are parameterisable.

- *Stuttering:* Stuttering is caused by a problem in the sequencing process: the sequencer slips and a portion of the EST is re-read. Stuttering can occur anywhere, but is most likely to occur after repeated Gs or Ts. We only model stuttering after repeated Ts and Gs since other stuttering events are very rare.
- *Ligation:* Ligation occurs when two ESTs bond together, giving the appearance of a new EST. The two ESTs that join together need not come from adjacent parts of an mRNA; indeed, they could come from different mRNAs. In the occurrence of the errors, ligation happens first, since the physical faults being modelled here occur in the real ESTs – the other errors discussed are artifacts of the sequencing process.

For an example of the probability distribution of different single base errors, see Figure 1. ESTSim has been built in a modular way so that it is relatively easy to build in new fault models. However, there are 10 parameters for the fault models at the moment, and in addition, the user can choose between three parametrised methods for splitting the input sequences; and we believe that this design space gives the user the ability to build realistic data sets. Full details can be found in [14]. The approach of ESTSim is similar to that of GenFrag [10], though it is tailored to EST data and supports more sophisticated error models.

5 ECLEST: A Tool for Evaluating EST Clusterings

The ECLEST tool (Evaluator for CLusterings of ESTs) takes as input a set of DNA-strings in FASTA-format and a text-file specifying the ideal clustering, and computes and evaluates a clustering: (1) using a specified similarity measure; (2) using a specified clustering algorithm; and (3) using a specified clustering validity index.

The interfaces are designed in such a way that ECLEST can be easily extended by new algorithms of any of the three categories. In the current version, five dissimilarity measures have been implemented, which are listed in Table 1, one clustering algorithm and one evaluation method.

Currently, only the single linkage clustering algorithm has been implemented. We compute a partition with the following property: Given threshold θ, $C(s) = C(t) \iff \exists s = s_1, s_2, \ldots, s_{k-1}, s_k = t$ such that for all $1 \leq i < k$, $d(s_i, s_{i+1}) < \theta$. The algorithm can be implemented efficiently with a union-find data structure, starting with a singleton cluster for each element. For small test sets, however, it can be more efficient to implement the clusters as linked lists. For the clustering validation, we implemented the Rand Index.

The ECLEST manual, as well as the Java code, can be found at `www.cebitec.uni-bielefeld.de/~zsuzsa/ESTclust.html`, including instructions on how to extend the application. For further details, including full implementation details, see [38].

dissim. measure	parameters	definition	implementation
end-space free alignment	penalty function f (end-space free)	$D_{\text{la}}(s,t) =$ $\min\{f(A) : A \text{ global alignment of } s,t\}$	dynamic progr. (Smith-Waterman)
common word	word size K	$D_{\text{cword}}(s,t) = \begin{cases} 0 & \text{if } \exists w, \lvert w\rvert = K, w \sqsubseteq s,t \\ 1 & \text{otherwise} \end{cases}$	truncated suffix tree (modification of Ukkonen's algo. [19])
symmetric d^2	bounds u,l window size m	$D_{d^2\,\text{sym}} = \hat{D}_{d^2}$, where $D_{d^2}(s,t) = \sum_{\lvert w\rvert=l}^{u}(freq_s(w) - freq_t(w))^2$	incremental computation with dynamic lists
asymmetric d^2	bounds u,l window size m skip size k	$D_{d^2\,\text{asym}}(s,t) = \min\{D_{d^2}(s',t^{(i)}) : \lvert s'\rvert = m, s' \sqsubseteq s, i = 0,\ldots,\lfloor\lvert t\rvert/k\rfloor\}$, using D_{d^2} as above and $t^{(i)} = t_{i\cdot k+1}\ldots t_{\min(i\cdot k+m,\lvert t\rvert)}$	modification of symmetric d^2
Boolean q-grams	word size q window size m	$\hat{D}_{\text{B }q\text{-gram}}$, where $D_{\text{B }q\text{-gram}}(s,t) = -\sum_{\lvert w\rvert=q} occ_s(w)\cdot occ_t(w)$	modification of symmetric d^2

Table 1: Dissimilarity measures currently implemented in ECLEST. All parameters are specified in a configuration file.

6 Suitability Evaluation for Single Linkage Clustering

6.1 Details of the experiments

Data used. For the experiments reported here, we used ESTSim to generate simulated ESTs from a collection of human cDNAs from a mammalian gene collection at `http://mgc.nci.nih.gov/`. Non-human cDNAs were removed by using BLAST and common contaminant information. Since in these first experiments, we did not want to include the effects of alternative splicing, we normalised the input data in such a way as to have one cDNA sequence per gene, by performing complete pair-wise comparison with BLAST: If the similarity was greater than e-85, one of these two sequences was removed, because we judged the probability that they were from the same gene to be high. This yielded a set of 699 cDNAs, which we split up into sets of 4 to 10 sequences each, such that the overall length (number of nucleotides) in each set was roughly the same. We refer to each of these sets as a *test set*; we had 134 test sets. 10 of these we used for preliminary testing, including determining the number of test sets we needed: These preliminary tests revealed that for statistically significant statements, we needed 120 test sets, hence the choice of the number of test sets.

We ran two experiments on these test sets, varying the parameters used by ESTSim to generate the simulated ESTs. Each experiment consisted of running ECLEST, for each test set, on the ESTs generated by ESTSim from this test set, with the five dissimilarity measures detailed below, and computing the

validity index for each of the five clusterings produced. We ran 10 preliminary tests for estimating the distribution of the dissimilarity measures and for setting up our hypotheses, and used the remaining 124 test sets (123 for the second experiment) to test these hypotheses. Each test set included between 200 and 300 ESTs as input to ECLEST.

The first experiment simulated high quality ESTs that meet the standards laid down by NCBI. The probability graph of a single base error, taking into account random noise, polymerase decay and primer interference is shown in Figure 1. A very modest amount of stuttering was permitted: The parameter was chosen such that stuttering would happen on average with probability 0.005 on a sequence of 10 Gs. In the second experiment, we increased most of the parameters of ESTSim in such a way as to produce ESTs that have twice as high error probabilities for most error types. In both experiments, ESTs of length between 300 and 500 bp were produced, where every base of the original cDNAs appears on average in 5 ESTs; no reverse reads were produced. No ligation was used in our initial tests, since, with single linkage clustering, when ligation occurs it may be impossible to cluster correctly because ligated ESTs can cause two separate clusters to be merged.

A more formal discussion of the parameters settings of ESTSim is out of place here. For the record, the following settings were used:

	manner of EST generation	α	β	γ	ζ	ξ	η	θ	κ, λ, μ	ν
Experiment 1	`samplerandom 300 500 5 0`	0.005	30	0.04	1	2	20	0	10 each	0
Experiment 2	`samplerandom 300 500 5 0`	0.01	50	0.06	2	3	0	0	10 each	0

Table 2: Parameters for ESTSim in the two experiments. See [14] for details.

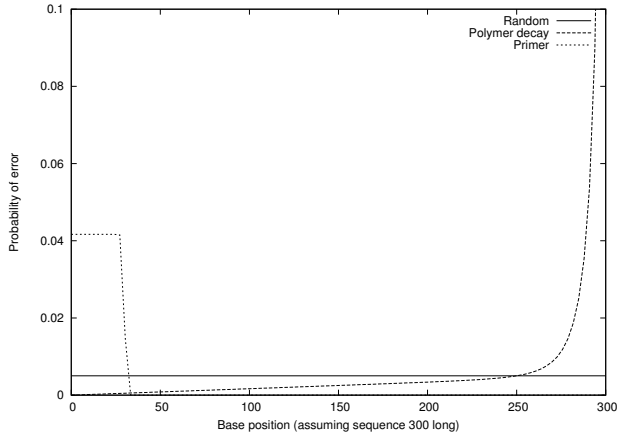

Figure 1: Error probability functions for different single base errors. For clarity, the y-axis is cropped at $y = 0.1$.

Dissimilarity measures compared. We compared the five dissimilarity measures detailed in Section 5; the parameters are shown in Table 3. We chose these dissimilarity measures for the following reasons. BLAST is frequently used as dissimilarity measure in EST clustering, thus in our alignment measure,

we used a penalty function which closely simulates the parameters used by BLAST. In these first experiments, we refrained from using BLAST itself, because we were aiming at a clean comparison which did not include the many heuristics used in a full BLAST implementation. Asymmetric d^2 is used by StackPack: We set all parameters accordingly, except for the threshold, which we optimised ourselves. We chose symmetric d^2 as a comparison to asymmetric d^2, in order to see whether the efficiency gained by comparing only every 50'th window of one of the two sequences can be justified: If the quality of the clusterings decreases dramatically, it cannot. Common word seems a good dissimilarity measure as a preprocessing step because of the possibility of very efficient implementations; however, it may itself already produce good quality clusterings. Finally, we chose Boolean q-grams because they, too, are used in EST clustering. The thresholds were set heuristically by running a few test runs on different size sets and finding the thresholds that maximize the validity score. The parameters are shown in Table 3.

measure	parameters	threshold
alignment	penalty function $f(\text{match}) = -1$, $f(\text{mismatch}) = +3$, $f(\text{gap opening}) = +5$, $f(\text{gap extension}) = +2$, $f(\text{end space gap}) = 0$	-20
symmetric d^2	fixed word size 6 ($u = l = 6$), window size $m = 100$	50
asymmetric d^2	fixed word size 6 ($u = l = 6$), window size $m = 100$, skip size $k = 50$	50
common word	word size $K = 19$	1
Boolean q-grams	word size $q = 6$, window size $m = 100$	-13

Table 3: Dissimilarity measures used in our experiments with parameters and thresholds.

6.2 Results

We ran 10 preliminary test sets to estimate the distributions of the clustering scores computed with the different dissimilarity measures, and to set up our hypotheses. We then ran 124 test sets to test our hypotheses. From the preliminary test sets it could be seen that the convenient assumption of normal distribution was untenable, thus we were restricted in the kind of statistical tests we could use. In all cases, we used either the Friedman ranking test or the binomial test. The Friedman test utilises only a ranking of the scores rather than their values. Another limitation is that the Friedman test has been designed for continuous distributions, which assumption slightly skews the results when several identical scores (*ties*) appear in one test. We therefore ran the test also by only using those test sets where the results were different for all dissimilarity measures in question; however, this reduced dramatically the number of test sets that could be evaluated.

Because of the missing normal distribution assumption, we were unable to make quantitative statements about *how much* better one dissimilarity measure performs than another. Only for the difference between two dissimilarity measures can a normal distribution be assumed (namely, symmetric d^2 and

asymmetric d^2), where we will thus be able to make a quantitative statement, as well; however, this will be reported in a later paper. All our statements (acceptance or rejection of hypotheses) have been made with a 95% confidence. We used the statistics program R, version 1.6.1 [11], to evaluate our data. The first two results are:

1. In both experiments, the clustering scores of the individual dissimilarity measures do not follow a normal distribution.
2. In both experiments, the clustering scores of the five dissimilarity measures do not have the same distribution.

The second statement could be shown both with data that included ties, and with the much smaller data set without ties. Next we compared those dissimilarity measures where it seemed reasonable to assume that either they differed significantly or that they were similar. Since the distribution of the common word measure was very different from the others, we excluded it from these individual comparisons. Finally, we tested whether certain dissimilarity measures can be used as preprocessing for others in order to achieve a good result. For technical reasons, this type of hypothesis was only tested in the first experiment, where ESTs of NCBI-quality were produced. The results are summarised in Table 4.

Hypothesis	with probability	
	NCBI-param.s (Experiment 1)	doubled NCBI-p.s (Experiment 2)
symmetric d^2 performs as well as alignment	< 0.5	> 0.05
asymmetric d^2 performs as well as symmetric d^2	> 0.5	not: > 0.05
symmetric d^2 performs better than Boolean q-grams	> 0.95	> 0.95
asymmetric d^2 performs better than Boolean q-grams	> 0.95	> 0.95
common word followed by symmetric d^2 performs as well as symmetric d^2 alone	> 0.95	—
common word followed by alignment performs better than symmetric d^2 alone	not: > 0.5	—

Table 4: Statistical results (95% confidence). Hypotheses that could not be validated are marked by "not". A dash '—' denotes that the hypothesis was not tested.

In addition to the statistical hypothesis testing, we also report our quantitative results. Table 5 reports the mean and standard deviation[4] of the clustering scores in the two types of experiments. Tables 6 and 7 summarise how the dissimilarity measures compared to each other in the two experiments. We visualise the results for Experiment 2 in Figures 2 and 3.

[4]To be exact, since the distributions are unknown, the values are *(i)* the average, and *(ii)* the average squared difference from the average.

	NCBI-parameters (Experiment 1)		doubled NCBI-param.s (Experiment 2)	
	Mean	**St.dev.**	**Mean**	**St. dev.**
alignment	0.9861	0.0181	0.9834	0.0199
symm. d^2	0.9768	0.0221	0.9568	0.0306
asymm. d^2	0.9718	0.0233	0.9471	0.0334
cword	0.8674	0.2037	0.8405	0.2284
Bool. q-grams	0.8262	0.0586	0.8171	0.0548

Table 5: Rand Index of the clusterings compared with ideal clusterings, with two types of input parameters. Mean and standard deviation taken over 134 tests (for NCBI-like parameters), and 133 tests (for doubled NCBI parameters).

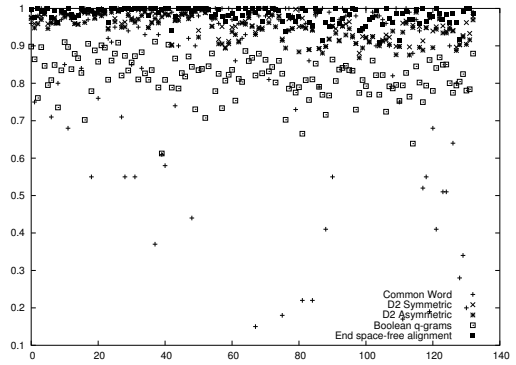

Figure 2: Rand Index for the five different dissimilarity measures for the 133 test sets in Experiment 2.

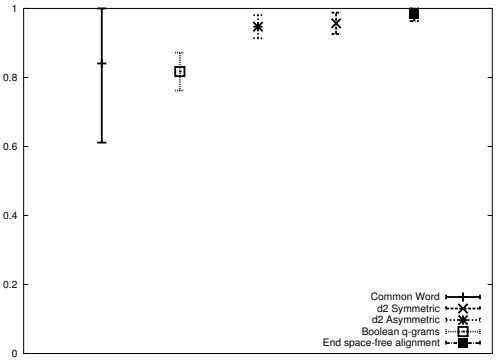

Figure 3: Mean and standard deviation of the Rand Index for the five different dissimilarity measures (Exp. 2).

7 Conclusion

This paper has made two contributions: First, we developed a rigorous methodology for comparing string similarity measures and clustering algorithms for the purposes of EST clustering, and developed two tools that implement this method. Second, we presented a preliminary study for the commonly used single linkage clustering algorithm, using common dissimilarity measures and two sets of typical values for EST error models.

Our evaluation methodology comprises: (1) Generating simulated EST data from cDNA or mRNA sequences, using user-specified error parameters. This is implemented by the system ESTSim. (2) Independently choosing the string dissimilarity measure, the clustering algorithm, and a clustering validity index to be employed, and computing a clustering of the data as well as a quality score. The system ECLEST implements all three components; at present, five similarity measures, one clustering algorithm, and one validity index have been implemented. (3) A statistical test for evaluating the statistical significance of the results.

The particular experiments we ran illustrated the value of the approach. They show that certain

	alignment	symm. d^2	assym. d^2	cword	Bool. q-grams
alignment		77/55/2	99/34/1	66/49/19	134/0/0
symm. d^2	1.5/41/57.5 %		40/94/0	64/21/49	134/0/0
asymm. d^2	0.5/25.5/74 %	0/70/30 %		64/12/58	134/0/0
cword	14/36.5/49.5 %	36.5/15.5/48 %	43.5/9/47.5 %		103/0/31
Bool. q-grams	0/0/100 %	0/0/100 %	0/0/100 %	23/0/77 %	

Table 6: Quantitative comparison of measures, for 134 tests with NCBI-like parameters. With absolute numbers for $i < j$ (top half), percentages, rounded to .5%, for $i > j$ (bottom half). Entry $a/b/c$ in cell (i, j) denotes that measure i produced a better clustering than measure j in a number (percent) of cases; i and j tied in b cases; and j produced a better result in c cases.

	alignment	symm. d^2	assym. d^2	cword	Bool. q-grams
alignment		125/7/1	131/1/1	77/52/4	133/0/0
symm. d^2	0.5/5.5/94 %		102/31/0	68/5/60	133/0/0
asymm. d^2	0.75/0.75/98.5 %	0/23.5/76.5%		66/1/66	133/0/0
cword	39/3/58 %	45/4/51 %	49.5/1/49.5%		93/0/40
Bool. q-grams	0/0/100 %	0/0/100 %	0/0/100 %	30/0/70%	

Table 7: Quantitative comparison, for 133 tests with doubled NCBI-parameters. Entries as in Table 6.

dissimilarity measures yield better results than others, but also that some measures compute very similar results. This gives algorithm designers greater choices in tradeoff considerations, when time and space limitations are vital. Furthermore, it gives rigorous ways of justifying the choice of certain heuristics to speed up the clustering process.

Extending the work. We are currently evaluating other dissimilarity measures such as BLAST and Chaos Game Representation [1]. And just as the choice of the measure will affect clustering, so does the choice of parameters of the particular dissimilarity measures chosen. We may find that the choice of parameters is much more important than the choice of measure or clustering algorithm. This will have important implications both for biological correctness and algorithm design.

Additional experiments to test different error models to see how the relative performance changes, particularly in very error-prone situations, would be useful. We would further like to explore whether other statistical techniques can be used to give better quantitative comparisons. Finally, we would like to make our tools publicly available, and add a graphical user interface for easier use.

Acknowledgements. Winston Hide from the South African National Bioinformatics Institute (SANBI) gave significant encouragement and help at the various stages of this research, which was started while two of the authors were visiting SANBI. Anton Bergheim was central to the development of ESTSim and helped us with the choice of biological parameters. We thank Andrew D. Barbour and the Statistics

Department of ETH Zurich for advice on the statistical test arrangement, Thomas Erlebach at the Computer Engineering and Networks Laboratory (TIK) and Kai Nagel, Institute for Scientific Computing, both ETH Zurich, for computing time. We are grateful to Jens Stoye and Sebastian Böcker for helpful comments on the paper. The program ECLEST and the tests presented in this paper constitute part of Judith Zimmermann's thesis for a Diploma (Masters) in Computer Science.

References

[1] J.S. Almeida, J.A. Carrico, A. Maretzek, P.A. Noble, and M. Fletcher. Analysis of genomic sequences by Chaos Game Representation. *Bioinformatics*, 17(5):429–437, 2001.

[2] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[3] M. Boguski and G. Schuler. ESTablishing a human transcript map. *Nature Genetics*, 10(11):369–371, 1995.

[4] E. Bolten, A. Schliep, S. Schneckener, D. Schomburg, and . Schrader. Clustering protein sequences—structure prediction by transitive homology. *Bioinformatics*, 17(10):935–941, 2001.

[5] J. Burke, D. Davison, and W. Hide. D2_cluster: A validated method for clustering EST and full-length cDNA sequences. *Genome Research*, 9(11):1135–1142, 1999.

[6] S. Burkhardt, A. Crauser, P. Ferragina, H.-P. Lenhof, E. Rivals, and M. Vingron. q-gram based database searching using a suffix array. In S. Istrail, P. Pevzner, and M. Waterman, editors, *Proceedings of the 3rd Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 77–83, Lyon, France, 1999. ACM Press.

[7] Alan Christoffels, Antoine van Gelder, Gary Greyling, Robert Miller, Tania Hide, and Winston Hide. STACKdb: Sequence tag alignment and consensus knowledgebase. *Nucleic Acids Research*, pages 234–238, 2001.

[8] S. Datta and S. Datta. Comparisons and validation of statistical clustering techniques for microarray data. *Bioinformatics*, 19(4):459–466, 2003.

[9] R. Dubes. Cluster analysis and related issues. In . P.S.P. Wang C.H. Chen, L.F. Pau, editor, *Handbook of Pattern Recognition & Computer Vision*, pages 3–32. World Scientific Publ. Co., Inc., River Edge, NJ, 1993.

[10] M.L. Engle and C. Burks. GenFrag 2.1: new features for more robust fragment assembly benchmarks. *Computer Applications in the Biosciences*, 10(5):567–568, 1994.

[11] The R Foundation for Statistical Computing. Available at `http://www.r-project.org/`.

[12] D. Gusfield. *Algorithms on strings, trees, and sequences*. Cambridge University Press, Cambridge, United Kingdom, 1997.

[13] E. Hartuv, A. Schmitt, J. Lange, S. Meier-Ewert, H. Lehrach, and R. Shamir. An algorithm for clustering cDNAs for gene expression analysis. In *Proceedings of the 3rd Annual International Conference on Computational Molecluar Biology (RECOMB 99)*, pages 188–196. ACM, 1999.

[14] S. Hazelhurst and A. Bergheim. ESTSim: A tool for creating benchmarks for EST clustering algorithms. Technical Report TR-Wits-CS-2003-1, School of Computer Science, University of the Witwatersrand, 2003. `ftp://ftp.cs.wits.ac.za/pub/research/reports/TR-Wits-CS-2003-1.pdf`.

[15] W. Hide, R. Miller, A. Ptitsyn, J. Kelso, C. Gopalkrishnan, and A. Christoffels. EST Clustering Tutorial, 1999.

[16] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[17] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice Hall, 1988.

[18] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comp. Surveys*, 31(3):264–323, 1999.

[19] P. Jokinen and E. Ukkonen. Two algorithms for approximate string matching in static texts. In Andrzej Tarlecki, editor, *Proceedings of the 16th Symposium on Mathematical Foundations of Computer Science. (MFCS '91)*, volume 520 of *LNCS*, pages 240–248, Berlin, Germany, 1991. Springer.

[20] A. Kalyanaraman, S. Aluru, S. Kothari, and V. Brendel. Efficient clustering of large EST data sets on parallel computers. *Nucleic Acids Research*, 31(11):2963–2974, 2003.

[21] A. Krause, J. Stoye, and M. Vingron. The SYSTERS protein sequence cluster set. *Nucleic Acids Research*, 28(1):270–272, 2000.

[22] F. Liang, I. Holt, G. Pertea, S. Karamycheva, S. Salzberg, and John Quackenbush. An optimized protocol for analysis of EST sequences. *Nucleic Acids Research*, 26(18):3657–3665, 2000.

[23] D.J. Lipman and W.R. Pearson. Rapid and sensitive protein similarity searches. *Science*, pages 1435–1441, 1985.

[24] K. Malde, E. Coward, and I. Jonassen. Fast sequence clustering using a suffix array algorithm. *Bioinformatics*, 19(10):1221–1226, 2003.

[25] B.M.E Moret, L.-S. Wang, and T. Warnow. Towards new software for computational phylogenetics. *IEEE Computer*, 35(7):55–64, 2002.

[26] G. Myers. A dataset generator for whole genome shotgun sequencing. In *Proceedings of Intelligent Systems in Molecular Biology (ISMB'99)*, pages 202–210. AAAI, 1999.

[27] H. Nash, D. Blair, and J. Greffenstette. Comparing algorithms for large-scale sequence analysis. In *"Proceedings of the Second IEEE International Symposium on Bioinformatics and Bioengineering"*, pages 89–96. IEEE Computer Society Press, 2001.

[28] G. Navarro. A Guided Tour to Approximate String Matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

[29] G. Pertea, X. Huang, F. Liang, V. Antonescu, R. Sultana, S. Karamycheva, Y. Lee, J. White, F. Cheung, B. Parvizi, J. Tsai, and J. Quackenbush. TIGR gene indices clustering tools (TGICL): a software system for fast clustering of large EST datasets. *Bioinformatics*, 19(5):651–652, 2003.

[30] P.A. Pevzner. *Computational Molecular Biology*. MIT Press, Cambridge, Massachusetts, 2000.

[31] J. Quackenbush, J. Cho, D. Lee, F. Liang, I. Holt, S. Karmycheva, Babak Parviyi, Geo Pertea, R. Sultana, and J. White. The TIGR gene indices: analysis of gene transcript sequences in higly sampled eukaryotic species. *Nucleic Acids Research*, 29(1):159–164, 2001.

[32] John Quackenbush. Computational analysis of microarray data. *Nature Reviews Genetics*, 2:418–427, 2001.

[33] D.C. Torney, C. Burks, D. Davison, and K. M. Sirotkin. Computation of d^2: A measure of sequence dissimilarity. In G. Bell and T. Marr, editors, *Computers and DNA*, pages 109–125. Addison-Wesley, 1990.

[34] E. Ukkonen. Approximate string-matching with q-grams and maximal matches. *Theoretical Computer Science*, 92:191–211, 1992.

[35] Susana Vinga and Jonas Almeida. Alignment-free sequence comparison – a review. *Bioinformatics*, 19(3):513–524, 2003.

[36] N. Wicker, D. Dembele, W. Raffelsberger, and O. Poch. Density of points clustering, application to transcriptomic data analysis. *Nucleic Acids Research*, 30(18):3992–4000, 2002.

[37] K.Y. Yeung, D.R. Haynor, and W.L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–318, 2001.

[38] Judith Zimmermann. Suitability Comparison of String Distance Measures for EST Clustering. Master's thesis, ETH Zurich, Dept. of Computer Science, 2003. ETH Zurich, Dept. of Computer Science, `www.ti.inf.ethz.ch/pw`.