BIOINFORMATICS ORIGINAL PAPER Vol. 22 no. 18 2006, pages 2232–2236
doi:10.1093/bioinformatics/btl368

Sequence analysis

RBR: library-less repeat detection for ESTs

Ketil Malde[1,*], Korbinian Schneeberger[3], Eivind Coward[2] and Inge Jonassen[1,2]

[1]Computational Biology Unit, Bergen Centre for Computational Sciences and [2]Department of Informatics, University of Bergen, Norway and [3]Genome-Oriented Bioinformatics, Wissenschaftszentrum Weihenstephan, Technische Universität München, Germany

Received on April 19, 2006; revised on June 29, 2006; accepted on July 3, 2006
Advance Access publication July 12, 2006
Associate Editor: Alex Bateman

ABSTRACT

Motivation: Repeat sequences in ESTs are a source of problems, in particular for clustering. ESTs are therefore commonly masked against a library of known repeats. High quality repeat libraries are available for the widely studied organisms, but for most other organisms the lack of such libraries is likely to compromise the quality of EST analysis.

Results: We present a fast, flexible and library-less method for masking repeats in EST sequences, based on match statistics within the EST collection. The method is not linked to a particular clustering algorithm. Extensive testing on datasets using different clustering methods and a genomic mapping as reference shows that this method gives results that are better than or as good as those obtained using RepeatMasker with a repeat library.

Availability: The implementation of RBR is available under the terms of the GPL from http://www.ii.uib.no/~ketil/bioinformatics

Contact: ketil.malde@bccs.uib.no

Supplementary Information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Expressed Sequence Tags (ESTs) are a valuable source of information that allows identification of genes in sequenced genomes. In addition, they provide a low-cost approach to mapping the gene complement of organisms for which genome projects are still too expensive. Clustering ESTs and assembling consensus sequences can reveal important information about the transcriptome, including splice variants and putative single nucleotide polymorphisms (SNPs).

The clustering procedure is based on identifying matches between the ESTs and grouping together sequences with sufficient overlaps. Repeats, in the form of sequencing artifacts, contamination, low quality sequence and genomic repeats, represent a serious problem for clustering, as they can occur in otherwise unrelated sequences. Matches resulting from repeats may lead to erroneously grouping ESTs that arise from different genes, and it is therefore important to mask repeats prior to clustering.

We have previously developed RepeatBeater (Schneeberger *et al*., 2005) and shown that it successfully masks repeats and improves the accuracy of clustering. In this paper we present a new method named RBR that is conceptually simpler than RepeatBeater, and while RepeatBeater was tightly coupled to `xsact` (Malde *et al*., 2003), RBR works independently of clustering method.

To analyze the effectiveness of RBR, we use several representative datasets of realistic sizes. We construct reference clusterings by mapping all ESTs to the genomic sequence, retaining only sequences that can be mapped unambiguously and grouping ESTs whose genomic matches overlap.

By comparing with the reference clusterings, we show that RBR produces results that are at least as good as those produced by the most widely used library-dependent repeat masking methods. Even for the set of human ESTs it gives results superior to those produced by methods using human repeat libraries. For some organisms, the results are dramatically improved over those produced by existing methods.

1.1 Existing methods

There are currently two main approaches to masking EST data. Library-based masking identifies undesired sequence parts by comparing with a library of known repeats. Autonomous masking identifies undesired sequence from properties of the sequence itself.

Repetitive regions often constitute large parts of a genome, and some classes of complex genomic repeats, like those originating from transposon and retrotransposon activity or retroviral elements, are collected in libraries which can then be used for masking.

While libraries derived from the genome are commonly used for masking ESTs, they are not targeted specifically at transcripts. For instance, LINE transposons normally contain genes, and for the purpose of EST clustering, the copies will either be treated as a single gene or as multiple homologues. Some transposons do cause problems, for instance the short SINE/Alu repeat is found in human ESTs (Schneeberger *et al*., 2005).

In EST data, there are also vector sequences and genomic contamination from the host organism, typically *Escherichia coli*. Similar to genomic repeats, these can be masked using a library.

Genomic repeats are in general organism specific. For novel organisms, where the genome is not yet available for constructing a good repeat library, effective masking of genomic repeats is difficult.

Autonomous masking typically deals with simple repeats, which consist of contiguously repeated short subsequences (2–4 nt) often caused by polymerase slippage, and low complexity regions (LCRs), which are regions containing an overabundance of some nucleotides. In addition to poly-A tails, which can be considered a special case of either type, these repeats are common in the UTR regions of genes.

*To whom correspondence should be addressed.

The accuracy of base calling tends to deteriorate towards the ends of the ESTs (Liang *et al*., 2000; Ewing and Green, 1998), and the error rate increases until at some point the remaining sequence is noise. These low-quality sequence parts are commonly trimmed. Usually a cut-off is set from quality values determined by the base-calling software, but heuristics using only the sequence exist.

RepeatMasker (Smit *et al*., 2004) is probably the most widely used tool for masking against a library (typically interspersed repeats and *E.coli* and vector sequences), and by default uses `cross_match` (P. Green, unpublished data) to identify matches. It also masks LCRs. RepeatMasker is distributed (and normally used) with the RepBase (Jurka *et al*., 2005) repeat library.

One popular utility for autonomous masking is `mdust` from TIGR, which implements the DUST algorithm (R. L. Tatusov and D. J. Lipman, unpublished data). Other alternatives include DustMasker (Morgulis *et al*., 2006) and SeqClean (G. Pertea, unpublished data), the latter combines autonomous masking with optional masking against a library. Search and alignment programs, like BLAST (Altschul *et al*., 1990) and BLAT (Kent, 2002), often incorporate autonomous masking as well.

There appears to be little consensus on a 'best practice' of steps to apply. TGICL (Pertea *et al*., 2003), a widely used EST analysis pipeline from TIGR, uses `mdust` to mask sequences for low complexity before clustering and assembly, but does not mask against genomic repeats.

Other pipelines (e.g. Krause *et al*., 2002; HarvESTer from Bio-Max Informatics AG, München) use either RepeatMasker or similar library-based masking (e.g. Pontius *et al*., 2003; Miller *et al*., 1999; D'Agostino *et al*., 2005) in addition to autonomous masking.

2 METHODS

RBR identifies repeats by first calculating the frequencies of word occurrences in the entire dataset. Then, for each sequence the frequencies of the words it contains are collected, and a threshold is determined based on these frequencies. Sequence positions corresponding to frequency peaks above this threshold are then masked. RepeatBeater (Schneeberger *et al*., 2005) used a similar approach, but its algorithm is more complex, and as the implementation relies on pre-processing the sequences with `xsact`, it is slower and more complicated to use.

2.1 The RBR algorithm

Given a word size k, RBR first calculates the frequency of every k-word in the full dataset. For each sequence, the distribution of word frequencies is collected, and the baseline for the sequence is estimated. Words with frequencies significantly above this baseline are then masked as repeats. In addition to the word size k, the algorithm takes additional parameters s, specifying the stringency of the baseline estimate, and d, specifying the threshold's offset from the baseline.

In the absence of repeats or read errors in the sequences, and with the sequences uniformly distributed over the originating gene, we expect the distribution of word counts to approximate a binomial distribution. To see this, let n be the number of sequences whose origin in the genome overlaps the origin of a given sequence S, and p be the average fraction of this overlap. For each position in S, the probability of one particular sequence overlapping it in that position is p, and as there are n sequences, we get a distribution of $B(n,p)$. Furthermore, from the assumed uniform distribution, and with the added assumption that sequences have similar lengths, we expect p to be 0.5.

In practice sequences vary in length and quality, and the distribution is non-uniform. Therefore the assumptions above lead to only an approximation of the actual distribution. Instead of assuming a fixed p, we estimate the distribution empirically.

The variance σ^2 of the binomial distribution $B(n,p)$ is $np(1-p)$, and thus the standard deviation grows with the square root of the mean. We therefore start by identifying the modal interval $(m-\sqrt{m}/2, m+\sqrt{m}/2)$ containing the maximum number of word occurrences.

For a binomial distribution, we can estimate $1-p$ from the empirical variance and mean as $\hat{q} = \hat{\sigma}^2/\hat{\mu}$. We adjust the interval around m so that estimated $\hat{q}$ for the frequencies in the interval is equal to the stringency parameter s. This distribution is used as the baseline to mask all words in the sequence with frequencies above $\hat{\mu} + d\hat{\sigma}$.

2.2 Constructing reference clusterings

When each EST contains enough non-repeat sequence to determine its position in the genome, the ESTs can be clustered by location, which greatly reduces the effect of any repeats. The result is similar to a clustering based on EST sequence similarity alone when the ESTs are masked optimally, and we use this to serve as a reference for comparing maskings (Kalyanaraman *et al*., 2003; Wang *et al*., 2004; Wu *et al*., 2004; Wu and Watanabe, 2005). We emphasize that this is only a coarse approximation to the biological transcripts, for example, incompleteness, sense-antisense transcription (Shendure and Church, 2002; Yelin *et al*., 2003) and trans-splicing events (Huang and Hirsh, 1992) will still contribute to inaccuracies in the clustering.

To assess RBR's performance, we developed GEM, a system for clustering ESTs based on overlap in genomic position. It uses BLAT (Kent, 2002) to determine a spliced alignment to the genome, and has several parameters to set criteria for the clustering.

To verify that RBR is able to identify and eliminate repeats, we used a set of 1 355 human ESTs taken from a cluster known to contain multiple genes SINE/Alu repeat (Schneeberger *et al*., 2005). The sequences were mapped to the human genome using GEM, resulting in 339 distinct clusters.

We also mapped 50 000 ESTs from *Oryza sativa*, and 100 000 from each of *Caenorhabditis elegans* and *Arabidopsis thaliana* to their respective genomes using GEM.

In order not to end up with artificially clean datasets, we used relatively lax parameters (90% sequence similarity and 75% of the sequence length) to match ESTs against the genome and discarded sequences that matched multiple locations equally well. The ESTs were then clustered if they overlapped 20 nt or more in the matches against the genome.

The resulting data sets contained 47 302 ESTs from *O.sativa*, 84 655 ESTs from *C.elegans*, and 97 919 from *Arabidopsis*.

2.3 Comparing clusterings

Many of the commonly used indices for comparing clusterings are based on categorizing and counting pairs of clustered objects. Specifically, for a set of objects $\{x_1, x_2, \ldots, x_n\}$, and clusterings K and C, we denote by $C(x_i)$ and $K(x_i)$ the clusters containing x_i in C and K, respectively. We then define four variables

$$a = |(i,j) \text{ where } C(x_i) = C(x_j), K(x_i) = K(x_j)|$$

$$b = |(i,j) \text{ where } C(x_i) = C(x_j), K(x_i) \neq K(x_j)|$$

$$c = |(i,j) \text{ where } C(x_i) \neq C(x_j), K(x_i) = K(x_j)|$$

$$d = |(i,j) \text{ where } C(x_i) \neq C(x_j), K(x_i) \neq K(x_j)|,$$

where $1 \leq i < j \leq n$.

Many cluster indices can then be calculated from these variables. In particular, the Jaccard index (Jain and Dubes, 1988) is defined as

$$J = \frac{a}{a+b+c}.$$

One problem with Jaccard, and indeed with all the measures based on counting pairs, is that modifications to large clusters will affect the score much more than modifications to small clusters.

A different approach that suffers less from this drawback is based on entropy. Meila (2005) introduces the variation of information VI to compare clusterings. For two clusterings K and C, VI is defined as

$$\mathrm{VI}(C,K) = H(C,K) - I(C,K)$$

where $H(C,K)$ is the total entropy of the clusterings, and $I(C,K)$ denotes the mutual information, defined as $I(C,K) = H(C) + H(K) - H(C,K)$.

The confusion matrix M is the $c \times k$ matrix defined by $M_{i,j} = |C_i \cap K_j|$. If we define $M_{i\cdot} = \sum_j M_{i,j}$ and similarly, $M_{\cdot j} = \sum_i M_{i,j}$, we have the following definitions

$$H(C) = \sum_i \frac{M_{i\cdot}}{n} \log \frac{M_{i\cdot}}{n}$$

$$H(K) = \sum_j \frac{M_{\cdot j}}{n} \log \frac{M_{\cdot j}}{n}$$

$$H(C,K) = \sum_{i,j} \frac{M_{i,j}}{n} \log \frac{M_{i,j}}{n}$$

VI can now be calculated directly from M as

$$\mathrm{VI} = \frac{1}{n}\Big(\sum_i M_{i\cdot} \log M_{i\cdot} + \sum_j M_{\cdot j} \log M_{\cdot j} - 2 \sum_{i,j} M_{i,j} \log M_{i,j} \Big)$$

We believe VI is a more appropriate measure, but for consistency with previous work, we will also provide the Jaccard index.

3 RESULTS

We have selected two parameter settings for RBR, the default configuration of $s = 1.5$ and $d = 6$, and a more aggressive configuration, using $s = 1$ and $d = 4$. We also masked the datasets with RepeatMasker, masking with lower case letters using the `-xsmall` option, and supplying the appropriate `-species` parameter.

TGICL pre-processes sequences with `mdust`, and uses MegaBLAST (Zhang *et al*., 2000) to identify matches, using exactly matching words of length 18 to seed pairwise alignments. Lower case letters are interpreted as masked sequence and excluded from the seeding, but not from the subsequent alignment. For clustering, we ran TGICL with its default parameters, and `xsact` with a word size (`-k` option) of 25 and a threshold (`-n`) of 60.

3.1 Human dataset

To verify that RBR produces a result similar to RepeatBeater, we first ran RBR on the set human ESTs. Using the default configuration, RBR masks 4.6% of the nucleotides differently from RepeatBeater. Comparing RBR to RepeatMasker results in a difference of 7.8%, while comparing RepeatBeater with RepeatMasker gives 7% differently masked nucleotides (Schneeberger *et al*., 2005). Thus, RBR and RepeatBeater are more similar to each other than either is to RepeatMasker.

From Figure 1 we see that RepeatMasker masks a relatively large part of the human dataset. A large fraction of the masked nucleotides are masked as SINE/Alu repeats, and these repeats constitute almost all nucleotides that are masked by both methods (6.3% versus 6.4%).

RBR masks a smaller amount of sequence, but with a large degree of overlap, and the differences are mainly owing to RBR only masking the most conserved parts of the repeats. In particular,

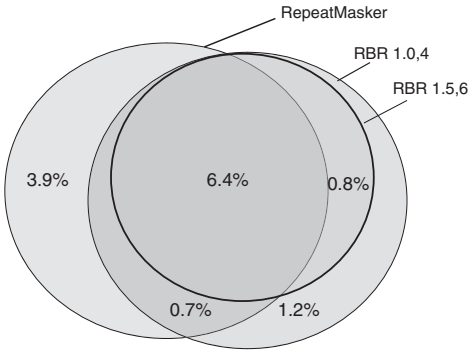

Fig. 1. Venn-diagram showing the correspondence between the different methods. A colour version of this figure is available as Supplementary data.

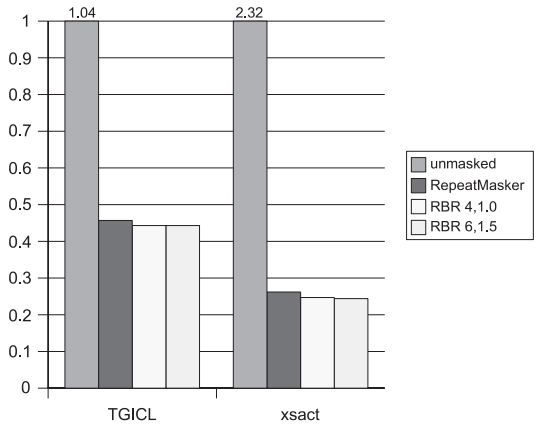

Fig. 2. Comparing the quality of clusterings using the different masking methods on the human dataset. The results are measured using the Variation of Information, using both `xsact` and TGICL for clustering. Lower scores are more similar to the reference. Note that the bars are truncated where the value exceeds one, and the correct value is given as a number.

only two sequences in the dataset are masked by RepeatMasker without being masked by RBR, and they turn out to contain be duplicates of the same sequence. Most of the sequence masked by RBR but not RepeatMasker is low complexity and simple repeats, either too short or too 'noisy' to be detected by RepeatMasker.

We also compared the clusterings obtained by running `TGICL` and `xsact` on the masked sequences with the clusterings obtained by mapping the ESTs to the genome using GEM. The results are displayed in Figures 2 and 3.

We see that the performance of both `TGICL` and `xsact` is poor on the unmasked data. The two maskings using RBR and the masking using RepeatMasker lead to similar clusterings, indicating that although RBR does not mask all SINE/Alu repeats, it masks it to the extent it adversely affects the clustering.

3.2 Large datasets

To see how RBR performs on more realistic datasets, we processed the larger datasets with RepeatMasker, `mdust` and RBR. The fraction of masked sequence is given in Figure 4. On these sets, RepeatMasker only masks a small fraction of the sequence data,

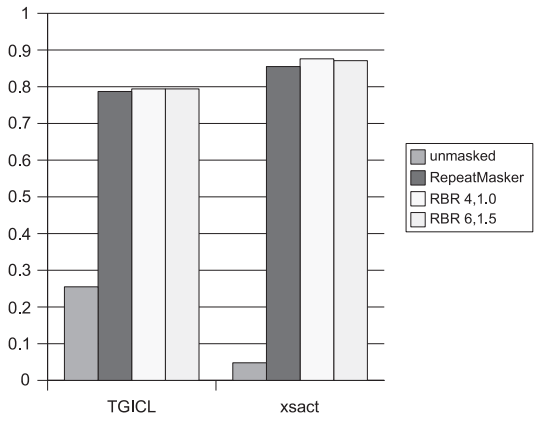

Fig. 3. Comparing the clusterings for the human dataset using the Jaccard index. Higher scores are more similar to the reference.

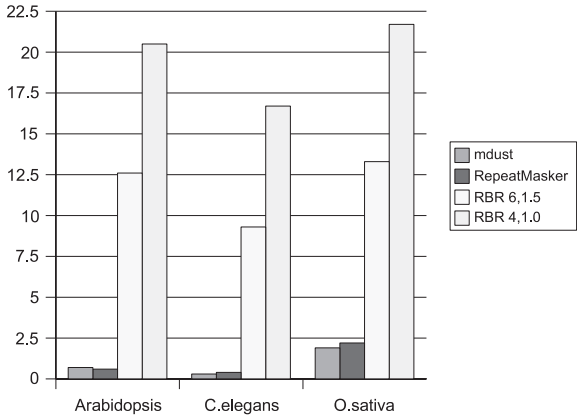

Fig. 4. Percentage of nucleotides masked for the larger datasets.

comparable with `mdust`. RBR masks a similar amount of sequence as it does on the human dataset.

Again we clustered the masked datasets with TGICL and `xsact`, and compared with the clusterings derived from the genomic mapping. The results are given as Variation of Information in Figures 5 and 6, and Jaccard indices in Figures 7 and 8.

We see that TGICL performs well on *C.elegans* and *O.sativa* using only the intrinsic masking. The *Arabidopsis* dataset, however, is markedly improved using RBR. For `xsact`, the clusterings generally see an even greater improvement.

4 DISCUSSION AND CONCLUSION

Except for the human dataset, which was selected explicitly for containing a known repeat, masking against genomic repeats in ESTs generally appear to have little effect on the clustering. Transcript-specific repeat databases would likely improve this situation, or at least reduce the amount of nucleotides masked unnecessarily.

In our experiments, we find no case where using RBR significantly worsens the results, and often the result is markedly improved. We think RBR, or a similar tool, should become an important element in EST analysis, complementing the standard approaches. In particular for exploring new organisms, where no good repeat library exists, we believe RBR provides the best available alternative for masking ESTs.

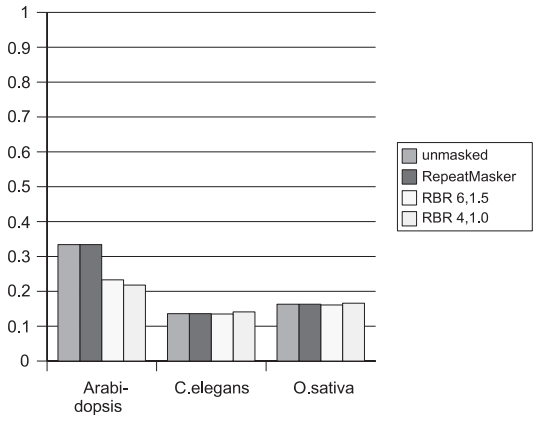

Fig. 5. Comparing clustering quality using TGICL on the large datasets using the Variation of Information metric. Lower scores are more similar to the reference clustering.

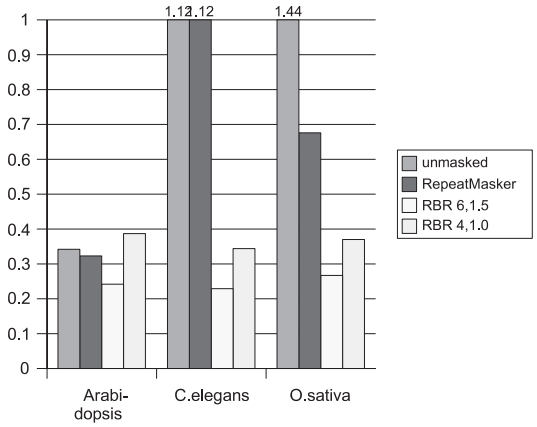

Fig. 6. Comparing clustering quality using `xsact` and Variation of Information.

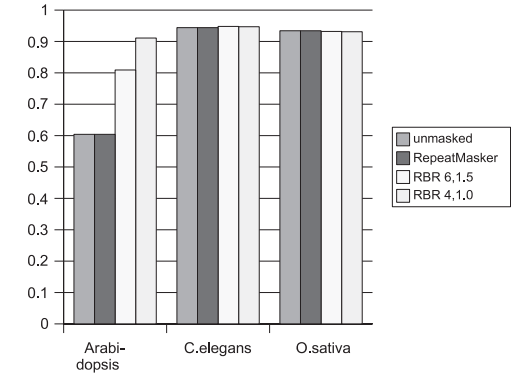

Fig. 7. Comparing clustering quality TGICL and the Jaccard index. Higher scores are more similar to the reference clustering.

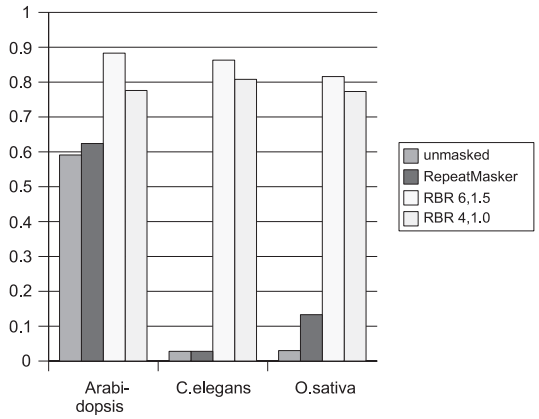

Fig. 8. Comparing clustering quality using xsact and the Jaccard index.

Our implementation of RBR masks the *C.elegans* dataset in 16 min on a 2.4 GHz Athlon 64, consuming <500 Mb of memory. Using word sizes >16 increases both time and memory consumption somewhat.

Although we have addressed repeat masking only in the context of EST clustering, we believe the method can be applied to related problems. For instance, many genomes are available as unassembled contigs, an approach similar to RBR could be used to aid assembly, or alternatively, enable analyses using the unassembled contigs directly.

4.1 Availability

RBR is licensed under the GNU General Public License, and can be downloaded from http://www.ii.uib.no/~ketil/bioinformatics/downloads/index.html.

The program that implements the Variation of Information measure, the Jaccard index and several other cluster comparison indices, is similarly licensed and available from the same URL.

The GEM clustering pipeline, the sequence data, and the resulting clusterings are available on request.

ACKNOWLEDGEMENTS

The authors are grateful to Manuel Spannagl at the GSF National Research Center for Environment and Health for providing the plant genome assemblies. The authors also thank Hans Georg Schaathun for his helpful comments and discussion. This work was funded by the Meltzer Fund and The National Programme for Research in Functional Genomics in Norway (FUGE) in the Research Council of Norway.

Conflict of Interest: none declared.

REFERENCES

Altschul,S. *et al.* (1990) A basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

D'Agostino,N. *et al.* (2005) ParPEST: a pipeline for EST data analysis based on parallel computing. *BMC Bioinformatics*, **6** (Suppl. 4).

Ewing,B. and Green,P. (1998) Base-calling of automated sequencer traces using Phred. II Error probabilities. *Genome Res.*, **8**, 185–194.

Huang,X.-Y. and Hirsh,D. (1992) RNA trans-splicing. *Genetic Eng.*, **14**, 211–229.

Jain,A.K. and Dubes,R.C. (1988) *Algorithms for Clustering Data*. Prentice-Hall, Inc.

Jurka,J. *et al.* (2005) Repbase update, a database of eukaryotic repetitive elements. *Cytogentic and Genome Research*, **110**, 462–467.

Kalyanaraman,A. *et al.* (2003) Efficient clustering of large EST data sets on parallel computers. *Nucleic Acids Res.*, **31**, 2963–2974.

Kent,W.J. (2002) BLAT—the BLAST-like alignment tool. *Genome Res.*, **12**, 656–664.

Krause,A. *et al.* (2002) SYSTERS, GeneNest, SpliceNest: exploring sequence space from genome to protein. *Nucleic Acids Res.*, **1**, 299–300.

Liang,F. *et al.* (2000) An optimized protocol for analysis of EST sequences. *Nucleic Acids Res.*, **28**, 3657–3665.

Malde,K. *et al.* (2003) Fast sequence clustering using a suffix array algorithm. *Bioinformatics*, **19**, 1221–1226.

Meila,M. (2005) Comparing clusterings—an axiomatic view. In *Proceedings of the 22nd International Conference on Machine Learning*.

Miller,R.T. *et al.* (1999) A comprehensive approach to clustering of expressed human gene sequence: the sequence tag alignment and consensus database. *Genome Res.*, **9**, 1143–1155.

Morgulis,A. *et al.* (2006) A fast and symmetric DUST implementation to mask low-complexity DNA sequences. *J. Comput. Biol.*, **13**, 1028–1040.

Pertea,G. *et al.* (2003) TIGR gene indices clustering tools (TGICL): a software system for fast clustering of large EST datasets. *Bioinformatics*, **19**, 651–652.

Pontius,J., Wagner,L. and Schuler,G.D. (2003) *UniGene: A Unified View of the Transcriptome*, chapter 21. NCBI Handbook.

Schneeberger,K. *et al.* (2005) Masking repeats while clustering ESTs. *Nucleic Acids Res.*, **33**, 2176–2180.

Shendure,J. and Church,G.M. (2002) Computational discovery of sense-antisense transcription in the human and mouse genomes. *Genome Biol.*, **3**, RESEARCH0044.

Smit,A.F.A. *et al.* (1996–2004) Repeatmasker open-3.0.

Wang,J.-P. *et al.* (2004) EST clustering error evaluation and correction. *Bioinformatics*, **20**, 2973–2984.

Wu,T.D. and Watanabe,C.K. (2005) GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics*, **21**, 1859–1875.

Wu,X., Lee,W.-J.A., Gupta,D. and Tseng,C.-W. (2004) ESTmapper: efficiently clustering EST sequences using genome maps. *Technical Report CS-TR-4575*. University of Maryland, MD.

Yelin,R. *et al.* (2003) Widespread occurrence of antisense transcription in the human genome. *Nat. Biotechnol.*, **21**, 379–385.

Zhang,Z. *et al.* (2000) A greedy algorithm for aligning DNA sequences. *J. Comput. Biol.*, **7**, 203–214.