Chapter 16

Analysis of Transposable Element Sequences Using CENSOR and RepeatMasker

Ahsan Huda and I. King Jordan

Abstract

Eukaryotic genomes are full of repetitive DNA, transposable elements (TEs) in particular, and accordingly there are a number of computational methods that can be used to identify TEs from genomic sequences. We present here a survey of two of the most readily available and widely used bioinformatics applications for the detection, characterization, and analysis of TE sequences in eukaryotic genomes: CENSOR and RepeatMasker. For each program, information on availability, input, output, and the algorithmic methods used is provided. Specific examples of the use of CENSOR and RepeatMasker are also described. CENSOR and RepeatMasker both rely on homology-based methods for the detection of TE sequences. There are several other classes of methods available for the analysis of repetitive DNA sequences including de novo methods that compare genomic sequences against themselves, class-specific methods that use structural characteristics of specific classes of elements to aid in their identification, and pipeline methods that combine aspects of some or all of the aforementioned methods. We briefly consider the strengths and weaknesses of these different classes of methods with an emphasis on their complementary utility for the analysis of repetitive DNA in eukaryotes.

Key words: Transposable elements, sequence analysis, bioinformatics, Repbase, CENSOR, RepeatMasker.

1. Introduction

Transposable elements (TE) are repetitive DNA sequences capable of moving from one chromosomal locus to another. The ubiquity of TEs has been appreciated for some time; they have been found in the genomes of a wide variety of species from all three domains of life. However, one of the major revelations of eukaryotic genome sequencing projects was the staggering abundance of

David Posada (ed.), *Bioinformatics for DNA Sequence Analysis,* Methods in Molecular Biology 537

DOI 10.1007/978-1-59745-251-9_16

TE-related sequences in large genomes. For instance, approximately one half of the human genome sequence was shown to consist of the remnants of TE insertion events *(1)*. In light of the sustained efforts underway to sequence and characterize numerous eukaryotic genomes, the prevalence of TEs necessitates the development and use of computational tools aimed at their detection, characterization, and analysis. After all, it is simply not possible to fully comprehend the structure, function, and evolution of eukaryotic genomes without a deep understanding of their TEs.

The most commonly used programs for the detection and analysis of TE sequences employ comparisons of genomic sequences to a library of consensus sequences that represent families of known repetitive (transposable) elements. This is the so-called homology-based method for the detection of TEs in genomic sequence. The Repbase Update *(2, 3)* is a comprehensive database of known eukaryotic repetitive sequence elements maintained by the Genetic Information Research Institute (GIRI; http://www.girinst.org). The developers of the Repbase Update, led by Jerzy Jurka, pioneered computational approaches toward the automatic detection of TEs in genomic sequences. DNA sequence searches against very early versions of Repbase, aimed primarily at the detection of Alu elements, were first carried out by the Pythia server *(4, 5)*. The Pythia server later gave way to the program CENSOR *(6, 7)*, which is still maintained and distributed by the GIRI. The tight integration of CENSOR with the Repbase Update library provides the user with access to the latest available TE annotations, which are constantly being updated at the GIRI. In addition to identifying known TEs in genomic sequence, CENSOR also provides for the de novo identification of simple sequence repeats that are characteristic of low complexity DNA regions *(8)*.

Arian Smit's RepeatMasker is another widely used program that identifies the location and identity of TEs in genomic sequence via searches against the Repbase Update library *(9)*. RepeatMasker employs a similar approach to compare genomic sequences against Repbase as the CENSOR program does. Additionally, RepeatMasker incorporates a great deal of *ad hoc* post-processing in order to try and ensure the best representation of TEs as single contiguous regions in genomic sequence. RepeatMasker has been used to annotate the TEs of numerous eukaryotic genomes, including the human genome sequence, and static releases of RepeatMakser annotations are widely distributed on various genome databases. Insight gained from RepeatMasker analyses has been critical to the field of genomics.

In this chapter, we will provide specific information on, and examples of, the use of the programs CENSOR and RepeatMasker along with a description of several other complementary classes of methods available for the analysis of repetitive DNA sequences.

1.1. Complementary Methods

CENSOR and RepeatMasker represent one general class of methods for the detection and analysis of repetitive DNA sequences. There are several additional classes of methods for the analysis of repetitive DNA: (i) de novo methods, (ii) class-specific methods, and (iii) pipeline methods. All of the different classes of methods have different strengths and weaknesses with respect to their ability to detect and characterize TEs in eukaryotic genome sequences. As such, they may be considered to be complementary, and indeed when different methods are compared on the same query sequence, they are often found to identify substantially non-overlapping parts of the sequence as being repetitive. Thus, investigators should be careful not to rely overly on one method or another. Homology-based methods in particular are limited by the extent of knowledge that already exists concerning the repetitive elements of a given genome or evolutionary lineage. In other words, the TEs, or their relatives, must have been previously characterized in order to be detected by homology-based methods. For this reason, these methods will perform poorly when applied to genomes that have many uncharacterized TE families. Homology-based methods will also be unable to detect novel TE families with distinct sequences. De novo methods, on the other hand, are ideal for identifying previously unknown repetitive DNA elements. However, de novo methods provide no information on the identity of these elements, or whether they are even TEs at all, and as such can be best used to simply mask repetitive elements. Clearly, homology-based methods are far better suited for investigations into the biology and genome dynamics of the TEs themselves.

1.1.1. De Novo Methods

Another general class of applications for identifying repeats in genomic sequence entails the so-called de novo methods that identify repeats by comparing genomic query sequences against themselves. Repeats are characterized in this way by clustering the similar groups of sequences that emerge from self comparison. De novo methods are interesting from an historical perspective because they represent the computational analogs of the re-association kinetic experiments that were first used to demonstrate the repetitive nature of eukaryotic genomes *(10)*.

De novo methods are naïve in the sense that they do not require any prior knowledge of the repetitive elements that may be present in the query sequence. This has the effect of eliminating ascertainment biases leading to false negatives for unknown repetitive elements. So in the formal sense de novo methods represent the most sensitive approach for the detection of repetitive DNA, and the recently developed WindowsMasker de novo method *(11)* has the added advantage of being much faster than homology-based methods. However, to work properly de novo methods require long and complete (or nearly so) query

sequences (i.e., whole contigs or genomes). More importantly, these methods do not provide any information on the characteristics of the repeats that are detected. De novo methods will report repeats of very different classes, such as tandem repeats, large segmental duplications, and interspersed repeats (TEs), together without discriminating among them. In other words, de novo methods work well for the detection and/or masking of repeat elements but do not aid in their characterization or analysis. De novo methods are also generally ineffective in identifying repetitive elements that are in low copy number as well as relatively ancient repetitive elements that may be too divergent from one another to be recognized as repetitive. RECON is another de novo method available for the detection of repetitive DNA sequences *(12)*.

1.1.2. Class-Specific Methods

Class-specific methods are a relatively recent development in the detection and analysis of TE sequences. For these methods, experts in the analysis of TEs have taken advantage of particular genomic features characteristic of specific classes of elements to aid in their identification. This approach has been most widely implemented with the LTR_STRUC program that identifies members of the long terminal repeat (LTR) containing class of TEs by virtue of the direct repeat sequences that are present at both ends of the elements *(13, 14)*. A recent publication presents a newly implemented method for the identification of LTR elements in eukaryotic genomes based on the same underlying rationale as LTR_STRUC *(15)*. However, in addition to identifying full length elements, this new program can also identify solo LTRs.

Since these kinds of methods do not rely on sequence identity (similarity) searches, they are particularly well suited to the identification of novel element families and low copy number elements. However, these methods are limited to families of elements that possess well-defined structural characteristics such as LTR elements and miniature-inverted repeat containing TEs (MITEs). Class-specific methods also enable the detection of novel TE sequences from a given element class while allowing for a deep interrogation of elements from that class. On the other hand, these methods will be particularly sensitive to sequence changes that accumulate after TE insertion and obscure the structural characteristics, such as inverted repeats, that they use to identify TEs.

1.1.3. Pipeline Methods

Pipeline methods, which combine aspects of all the aforementioned approaches to TE detection, probably represent the most rigorous and accurate class of method available for the annotation of TE sequences in eukaryotic genomes. Examples of pipeline methods are the MITE analysis toolkit (MAK) *(16)* and a more

recently proposed pipeline method, which promises to provide the most accurate and reliable annotations of TE sequences in eukaryotic genomes to date *(17)*. While these methods are very powerful in principle, they are also among the least accessible to the user because their use entails far more effort than any of the other single methods. Because pipeline methods integrate so many distinct applications, they also require a high level of sustained development and maintenance. Pipeline methods may well become the standard approach for genome annotation and serve the community best by providing static TE annotations of eukaryotic genomes as opposed to readily usable tools for investigators to query their own sequences of interest.

2. Program Usage

2.1. CENSOR

CENSOR allows for the identification and characterization of repetitive elements in genomic sequences. CENSOR can be used to mask repetitive sequences to allow for the more efficient use of downstream applications that are confounded by the presence of repeats and it can also be used to identify and characterize repetitive sequences in order to study the biology of the elements themselves.

2.1.1. Purpose

2.1.2. Availability

CENSOR is freely available to download from the GIRI for local installation (http://www.girinst.org/censor/download.php). CENSOR can be run locally using Unix type computer operating systems. Running CENSOR locally requires the installation of a local version of Repbase, which is optionally included in the download package, as well as the WU-BLAST package *(18)*. CENSOR can also be run from a server on the GIRI website (http://www.girinst.org/censor/index.php).

2.1.3. Input

Sequences in FASTA, GENBANK, and EMBL formats can be submitted to CENSOR by uploading a file to their server or by pasting them in the query textbox. CENSOR accepts DNA as well as protein sequences as input and decides the version of BLAST to use given a particular query sequence. One or more sequences can be submitted in a particular query.

2.1.4. Output

CENSOR runs yield a number of distinct kinds of output including (a) a repeat map indicating the location of repeats on the query sequence, (b) annotation of the repeat location, type, and its similarity and positive score values, and (c) a "masked" sequence file that returns the repetitive sequences replaced by Ns or Xs.

2.1.5. Method

CENSOR uses WU-BLAST *(18)* or NCBI BLAST *(19)* algorithms to search the query sequence against the Repbase Update library of repetitive sequences. CENSOR can be run on three different speed/sensitivity settings (*see* **Note 1**). It can automatically run an appropriate version of BLAST such as BLASTN, BLASTP, BLASTX, and TBLASTN in order to accommodate the various input types used for querying repetitive elements. This feature adds flexibility to the algorithm in contrast to RepeatMasker, which only uses DNA sequences in its searches. All options available through BLAST can also be incorporated in CENSOR searches. CENSOR uses an information theoretic method to detect simple sequence repeats such as satellite DNA and low complexity sequences. CENSOR also post-processes data to give an interactive positional map of the query sequence (similar to the NCBI BLAST web interface). In addition, it calculates the similarity values and positive score values for alignments between query and element consensus sequences. The similarity value can be used to approximate the evolutionary of the age of the TEs.

2.2. RepeatMasker

RepeatMasker serves to identify, characterize, and mask repetitive elements in genomic sequences. It is most often used to simply mask identified repeats in genomic sequence so that other analyses can be run on the resulting non-repetitive DNA sequences. However, RepeatMasker also characterizes repeats by class, family, and individual element name based on the Repbase library, and this information is critical to the study of TEs. Divergence values between TEs and their family consensus sequences are also provided and these can be used to determine the relative age of the elements.

2.2.1. Purpose

2.2.2. Availability

RepeatMasker can be run in two different ways. The program can be downloaded from http://www.repeatmasker.org/RMDownload.html and installed locally, or it can be run on a web server http://www.repeatmasker.org/cgi-bin/WEBRepeatMasker. To install and run RepeatMasker locally, users will also need to install a local copy of the Repbase library as well as the programs WU-BLAST *(18)* and CROSS_MATCH *(20)*.

2.2.3. Input

RepeatMasker works only on DNA sequences and the query sequences have to be in FASTA format. Sequences can be submitted using a file with one or more sequences or by pasting the sequence(s) in the submission box. Extremely long sequences, or files with numerous sequences, will be automatically broken down into batches to be run by RepeatMakser.

2.2.4. Output

RepeatMasker runs yield three files: (a) annotation of the location, type, and percent divergence of repeat from the consensus sequence, (b) a sequence file that has the repetitive sequences replaced by Ns or

Xs, and (c) a summary of the repetitive content of the query sequence. Additional output files, including alignments between query and consensus sequences, can be optionally included.

2.2.5. Method

RepeatMasker scans the query sequence using the program CROSS_MATCH *(20)* against the library of consensus sequences provided by Repbase Update. CROSS_MATCH implements the Smith-Waterman (SW) dynamic programming algorithm *(21)* that guarantees optimal pairwise sequence alignments. Using CROSS_MATCH, a score matrix is first constructed based on exact word matches between the library sequences and the query sequence. This is then expanded to include a "band" of sequences that surround the exact match. The band is based on the overlap of SW scoring matrices. The width of the band, and thus the sensitivity of RepeatMasker, can be adjusted using different speed settings to allow for wider or narrower acceptance of sequences surrounding the band. Since there can be many consensus sequences in the Repbase Update library that match the same region of the query sequence, the search engines return the matrices that have less than 80–90% overlap with each other. Typically the sequence with the highest SW score is selected for annotation after various approximation improvements. RepeatMasker can also use WU-BLAST to search against Repbase to improve the speed of searches *(22)*. Simple repeats are detected by computing the AT or GC content for overlapping windows of 200 bp and then checking for characteristics attributed to most simple repeats. RepeatMasker uses stringent criteria for identifying simple repeats and low-complexity DNA, which can result in omission of some repeats.

3. Examples

3.1. CENSOR

We provide an example of running CENSOR from the GIRI web server. The URL http://www.girinst.org/censor/index.php points to the CENSOR submission page (**Fig. 16.1**). We used a 2-kb DNA sequence from the proximal promoter region of the human hydroxysteroid (17-beta) dehydrogenase 13 gene as an example query (Genbank mRNA accession NM_178135). The FASTA format sequence is pasted into the submission page textbox as shown; note that a file with the sequence could also be uploaded using the Browse and Submit buttons shown (**Fig. 16.1**). For the purposes of this search, the "Mammalia" option of the "Sequence source" is chosen. This option specifies which subset of Repbase will be searched and in this case the subset will include all repeat sequences that are common to mammals as

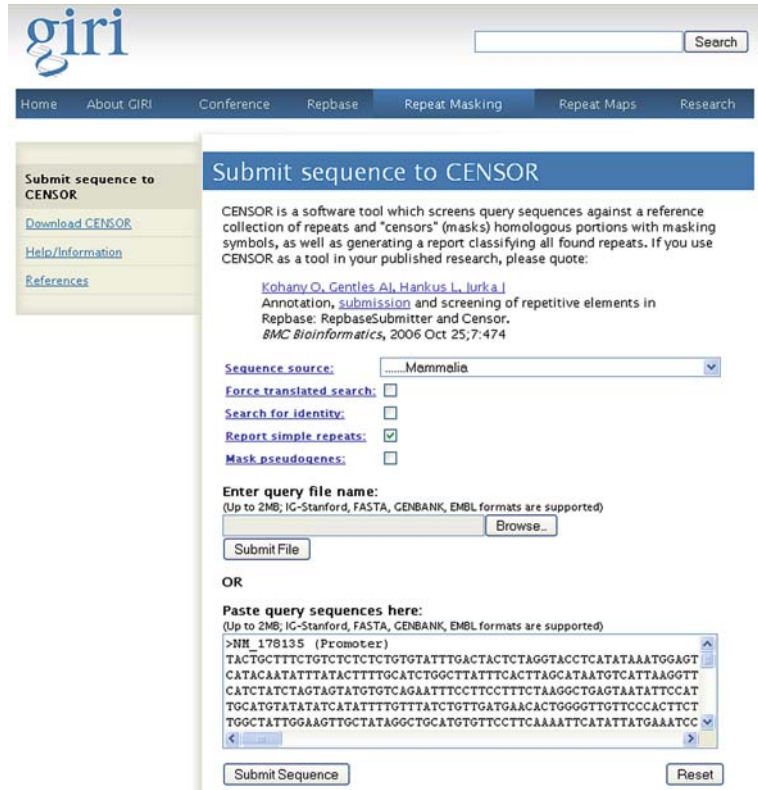


Fig. 16.1. CENSOR web server query submission page.

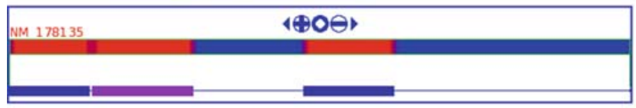
well as those specific to individual mammalian species. The "Report simple repeats" option is also selected to identify simple sequence repeats. Since the sequence is non-coding a translated search is not used. Neither the option "Search for identity," which forces the program to search for only identical or nearly identical sequences, nor the option "Mask pseudogenes," which searches for pseudogenes, is selected in this example.

Once the query sequence is pasted (or uploaded) and the appropriate options are selected, the search is run using the "Submit Sequence" button (**Fig. 16.1**). There are several output displays provided by CENSOR. CENSOR post-processes data to give an interactive positional map of the query sequence along with a summary table of identified elements (**Fig. 16.2**). On the positional map, the query sequence is represented by the horizontal bar with red representing repetitive (masked) DNA and blue representing non-repetitive DNA. The individual repeats and their annotations are shown below the bar; mouse-overs yield the element name and classification, and clicking on the element links to its Repbase entry. A masked version of the sequence is also provided (**Fig. 16.3**), as are alignments of the query sequence with

Map of Hits

SVG viewer is required to view graphical representation of the map as Scalable Vector Graphics (SVG plot).

NM_178135 (SVG Plot; Alignments; Masked)



Name	From	To	Name	From	To	Class	Dir	Sim	Pos	Score
NM_178135	3	260	L1MB4	448	714	Non-LTR/L1	c	0.8605	0.86	1597
NM_178135	269	592	MLT1A0	19	374	ERV/ERV3	d	0.7798	0.78	1369
NM_178135	948	1239	AluSq	1	282	Non-LTR/SINE/SINE1	c	0.7746	0.77	1317

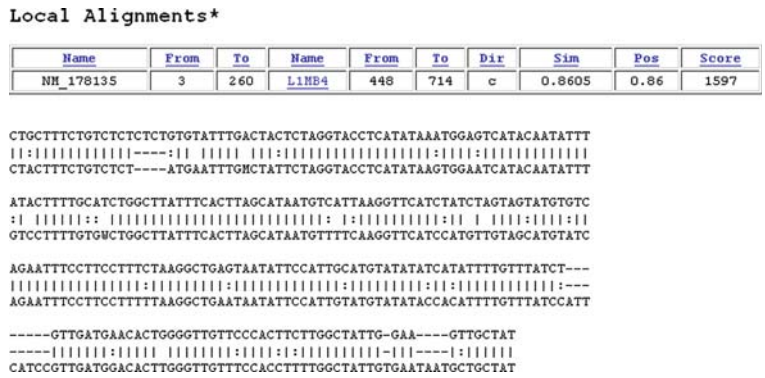
Fig. 16.2. CENSOR output repeat map and summary table. The location of repeats and their identity are shown as a graphical schematic and also listed line-by-line. Links to the Repbase entries for individual elements are provided in the schematic Figure and the table.

Masked Sequence

```
>NM_178135
TAXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXAGGCTGCAXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXGAAGCCACCCCACTTTAAAGTTGAGATTCCCGACTCCT
GGTCTAATGCTCTTTCCAGTATTATCACAAAGGGAACTGATGGTTCTGTATCCTAAAATCTCCCCCAAAC
TCAAGAGATTTTCAGGGAAATGGTCATCATGTGTAAAATAATTAGCCAGTTGAAATATTGATGCTGAGCT
TTTGTGAATAAATGAATCAAATAGCTAAGCTGGGTTCATTCAGACTTTATCTAAGTTTCTTTAAGCTTAT
CTCTACATTCTCCTTTTTCATATTGAAAAAAGGTAACAATTTCCAGAAAGATACCAACAGATCATTAGCT
GATGCAGTATCTCCAGAATTCTTTTTCTTTTTTTTTTXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXCCTAGAATTGTTTCTAGAGGT
GAAACTTCAAGGTGAAATATAGTACATAACTGCTTTTCAGATAAACAAGTCCAGAGAGCACACTCTCTTG
TGCTCTTGGCATCACTTGGCATCACTTCATATTTGAGGTGTTTCAAACCCATTAGAACACGTGAACAAGG
CCTGCTTCCAAAGCTGGCTTCCATCTGGTAGTCCCATTAACAACTGGGCACACCCCTTCCCTAGAGCTCT
GTGTAGACAGTACCTCCTCCCTAGGACTACACAAGGACTGAACCAGAAGGAAGAGGACAGAGCAAAGCCA
TGAACATCATCCTAGAAATCCTTCTGCTTCTGATCACCATCATCTACTCCTACTTGGAGTCGTTGGTGAA
GTTTTTCATTCCTCAGAGGAGAAAATCTGTGGCTGGGGAGATTGTTCTCATTACTGGAGCTGGGCATGGA
ATAGGCAGGCAGACTACTTATGAATTTGCAAAACGACAGAGCATATTGGTTCTGTGGGATATTAATAAGG
TAATGTATACATCTTCCAACTTTTTAAAGTCACAGAGTAAGATATGTATTTTAAGAATTATTTGACTTAC
CATCTACTTATCTTTGTATTTTTGTTTTTCAAAGTTTGATAAATTCCCTGGTCCCTTAGTCTGTATATGT
GTCAGGTTAGTTAGATGAAGGGAATGTAATTAAGAACTAAGCAGCGATTTTTATGACATGGTGTGCAGGT
TGATAGAAAGACTCAGGAGCCAGTCTCCTTCCAAGCTGCT
```

Fig. 16.3. Query sequence with repetitive elements masked by “X”s.

each repeat consensus sequence found in Repbase (**Fig. 16.4**). CENSOR also shows the Repbase records for all the repeats found in the query. Along with the alignments, CENSOR calculates the similarity values and positive score values (**Figs. 16.2** and **16.4**). The similarity value (Sim) reports the number of matches in the alignment normalized by the length of the alignment, while the positive value (Pos) reports the normalized number of alignment


Local Alignments*

Name	From	To	Name	From	To	Dir	Sim	Pos	Score
NM_178135	3	260	L1MB4	448	714	c	0.8605	0.86	1597

```
CTGCTTTCTGTCTCTCTCTGTGTATTTGACTACTCTAGGTACCTCATATAAATGGAGTCATACAATATTT
||:||||||||||||----:|| ||||| |||:||||||||||||||||||:||||:||||||||||||
CTACTTTCTGTCTCT----ATGAATTTGMCTATTCTAGGTACCTCATATAAGTGGAATCATACAATATTT

ATACTTTTGCATCTGGCTTATTTCACTTAGCATAATGTCATTAAGGTTCATCTATCTAGTAGTATGTGTC
:| |||||:: |||||||||||||||||||||||||||||: |:||||||||||:|| | ||||:||||:||
GTCCTTTTGTGWCTGGCTTATTTCACTTAGCATAATGTTTTCAAGGTTCATCCATGTTGTAGCATGTATC

AGAATTTCCTTCCTTTCTAAGGCTGAGTAATATTCCATTGCATGTATATATCATATTTTGTTTATCT---
||||||||||||||||:||||||||:|||||||||||||:||||||||:||:|||||||||||:---
AGAATTTCCTTCCTTTTTAAGGCTGAATAATATTCCATTGTATGTATATACCACATTTTGTTTATCCATT

-----GTTGATGAACACTGGGGTTGTTCCCACTTCTTGGCTATTG-GAA----GTTGCTAT
-----|||||||:||||| |||||||:||||:|:||||||||||-|||----|:||||||
CATCCGTTGATGGACACTTGGGTTGTTTCCACCTTTTGGCTATTGTGAATAATGCTGCTAT
```

Fig. 16.4. CENSOR alignment between query sequence and its closest related Repbase consensus sequence. Summary data on the alignment are provided.

positions that produce positive scores in the alignment matrix. Finally, a summary table is provided listing the total number and length of elements identified from different Repbase classes and families.

3.2. RepeatMasker

RepeatMasker can also be run from a web server as shown in the following example. For consistency, the same 2-kb human promoter sequence (Genbank mRNA accession NM_178135) that was used in the CENSOR example is used for RepeatMasker. The URL http://www.repeatmasker.org/cgi-bin/WEBRepeatMasker points to the RepeatMasker submission page (**Fig. 16.5**). The first part of the query submission form is the "Basic Options" section. This allows for the uploading of a FASTA format sequence file or pasting sequence(s) into the text box. Users can select from a number of options that allow for the specification of the search algorithm to be used, the speed/sensitivity settings (*see* **Note 2**), the source of the query sequence, and the return format and method. More advanced search options can be found under the "Lineage Annotation Options" and "Advanced Options" sections (**Fig. 16.6**). The "Lineage Annotation Options" settings allow users to choose one or two comparison species against which the query can be surveyed for lineage-specific repeats. This can help to refine and narrow search results by eliminating elements common to many species. These options only work for mammalian query sequences due to the greater coverage of mammalian repeats in Repbase. The "Advanced Options" allow users to display alignments, choose how sequences are masked, decide on what kinds of repeats are to be masked, and adjust the sequence similarity matrix to be used based on GC-content of the query sequence.

RepeatMasker returns a number of output files describing the repeat content of the query and providing the masked sequence. A RepeatMasker summary file (or screen on the web server) is a

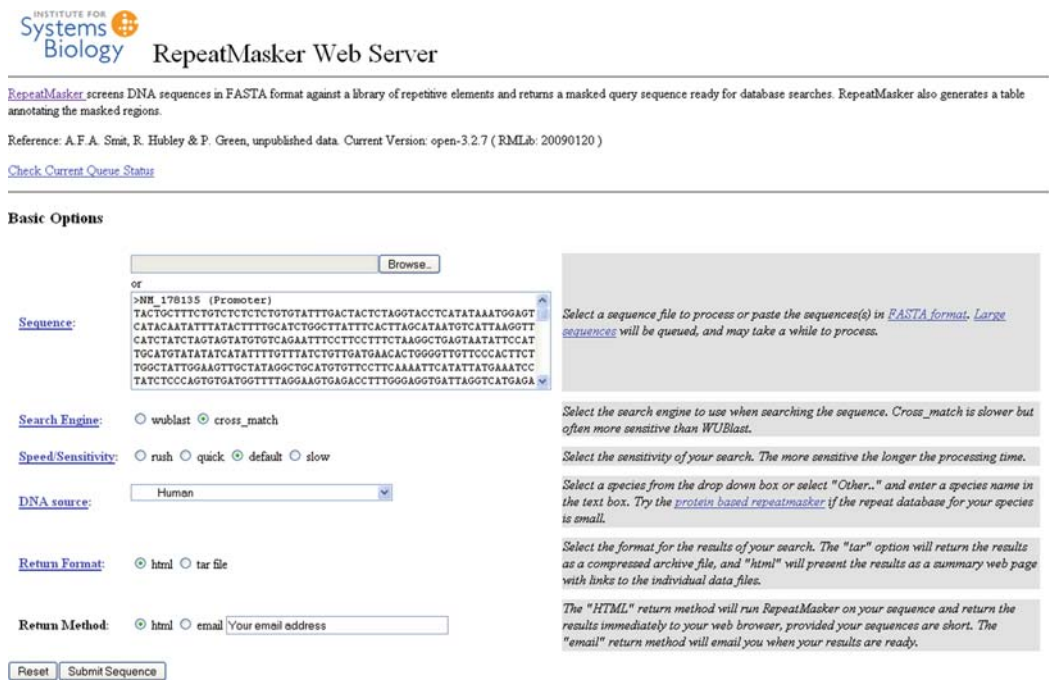


Fig. 16.5. RepeatMasker web server query submission page (Part 1). The "Basic Options" part of the submission page is shown.

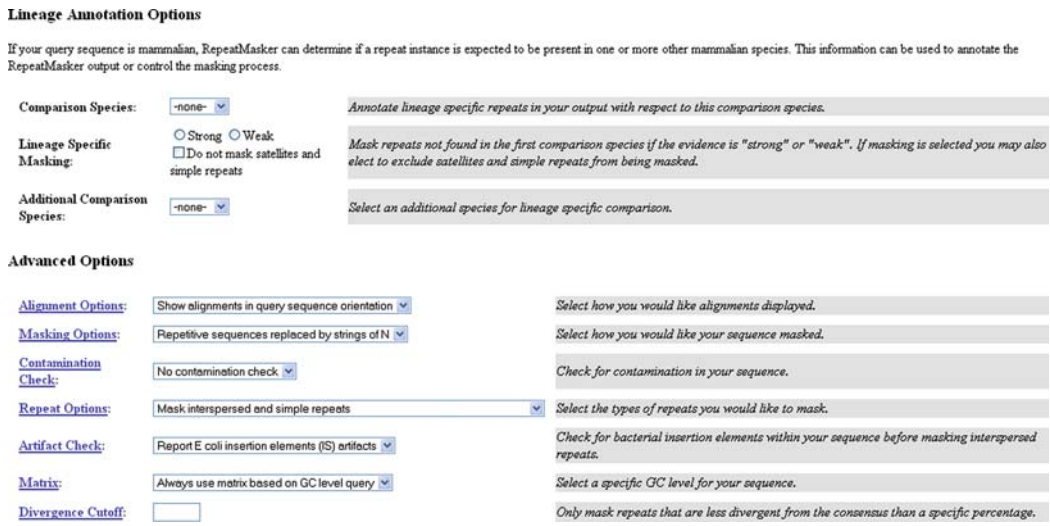


Fig. 16.6. RepeatMasker web server query submission page (Part 2). The "Lineage Annotation Options" and "Advanced Options" parts of the submission page are shown.

summary table that lists the percentage of query sequence masked by the different types of repeats (**Fig. 16.7**). A more detailed table is also provided with information on each individual repeat that is identified (**Fig. 16.8**). This table includes data on the

Summary:

```
==================================================
file name: RM2sequpload_1233250641
sequences:             1
total length:       2000 bp  (2000 bp excl N/X-runs)
GC level:          40.95 %
bases masked:        896 bp ( 44.80 %)
==================================================
                number of      length   percentage
                elements*    occupied  of sequence
--------------------------------------------------
SINEs:                 1          311 bp   15.55 %
      ALUs             1          311 bp   15.55 %
      MIRs             0            0 bp    0.00 %

LINEs:                 1          247 bp   12.35 %
      LINE1            1          247 bp   12.35 %
      LINE2            0            0 bp    0.00 %
      L3/CR1           0            0 bp    0.00 %

LTR elements:          1          338 bp   16.90 %
      ERVL             0            0 bp    0.00 %
      ERVL-MaLRs       1          338 bp   16.90 %
      ERV_classI       0            0 bp    0.00 %
      ERV_classII      0            0 bp    0.00 %

DNA elements:          0            0 bp    0.00 %
     hAT-Charlie       0            0 bp    0.00 %
     TcMar-Tigger      0            0 bp    0.00 %

Unclassified:          0            0 bp    0.00 %

Total interspersed repeats:       896 bp   44.80 %


Small RNA:             0            0 bp    0.00 %

Satellites:            0            0 bp    0.00 %
Simple repeats:        0            0 bp    0.00 %
Low complexity:        0            0 bp    0.00 %
==================================================
```

Fig. 16.7. RepeatMasker summary table output. Data on the length (bp) and percentage of different classes of identified repeats are provided.

SW score	perc div.	perc del.	perc ins.	query sequence	position in query begin	end	(left)		matching repeat	class/family	position in repeat begin	end	(left)	ID
1440	12.3	3.2	1.6	NM_178135	3	249	(1751)	C	L1MB3	LINE/L1	(216)	5967	5717	1
1514	19.1	8.9	0.9	NM_178135	255	592	(1408)	+	MLT1A0	LTR/ERVL-MaLR	1	365	(0)	2
1932	14.5	0.6	4.2	NM_178135	929	1239	(761)	C	AluJo	SINE/Alu	(12)	300	1	3

Fig. 16.8. RepeatMasker table output. Data for each individual repetitive element identified are provided.

levels of divergence between the query and consensus sequences along with location information specifying where the repeats are found in the query and which part of the repeats are represented. As was shown for CENSOR, RepeatMasker also provides a FASTA file with the repeats masked out, and the program can be configured to show alignments between repeats and their family consensus sequences.

RepeatMasker can also be run from the command line on Unix type operating systems. An example of the command line for the same search that was demonstrated for the web server is "RepeatMasker NM_178135.fasta -species human -alignments." Running RepeatMasker locally allows users to employ their own repeat libraries to search against. Another one of the advantages of the local RepeatMasker installation is the very detailed documentation that is provided including information on all command line options and flags. A list of all command line flags with brief descriptions can be obtained by simply typing "RepeatMasker" at the prompt. Typing "RepeatMasker –h(elp)" will print out all of the documentation.

4. Notes

1. Since CENSOR is powered by BLAST searches, search time varies directly with the length of the query and database and it can be run in three different speed/sensitivity settings. CENSOR uses WU-BLAST or NCBI-BLAST heuristics, which are both several times faster than the CROSS_MATCH dynamic programming algorithm employed as default by RepeatMasker.
2. The time complexity of the SW algorithm used by RepeatMasker is $O(n^2)$ where n is the word length. Therefore the time to process sequences increases sharply with length. Consequently, the speed settings are directly related to the word length used in CROSS_MATCH searches. In general, the program loses 5–10% sensitivity at each step of the speed settings, while gaining speed at a much higher rate. The time difference between the fastest and the slowest settings is approximately 30X. Heuristic WU-BLAST searches with RepeatMasker are generally much faster and compare to the fastest setting using CROSS_MATCH search algorithm.

Acknowledgments

The authors wish to thank Leonardo Mariño-Ramírez and Jittima Piriyapongsa for comments and technical support. Ahsan Huda and I. King Jordan are supported by the School of Biology at the Georgia Institute of Technology.

References

1. Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., et al. (2001) Initial sequencing and analysis of the human genome. *Nature* **409,** 860–921.
2. Jurka, J. (2000) Repbase update: a database and an electronic journal of repetitive elements. *Trends Genet* **16,** 418–20.
3. Jurka, J., Kapitonov, V. V., Pavlicek, A., Klonowski, P., Kohany, O., and Walichiewicz, J. (2005) Repbase Update, a database of eukaryotic repetitive elements. *Cytogenet Genome Res* **110,** 462–7.
4. Jurka, J., and Milosavljevic, A. (1991) Reconstruction and analysis of human Alu genes. *J Mol Evol* **32,** 105–21.
5. Jurka, J., Walichiewicz, J., and Milosavljevic, A. (1992) Prototypic sequences for human repetitive DNA. *J Mol Evol* **35,** 286–91.
6. Jurka, J., Klonowski, P., Dagman, V., and Pelton, P. (1996) CENSOR – a program for identification and elimination of repetitive elements from DNA sequences. *Comput Chem* **20,** 119–21.
7. Kohany, O., Gentles, A. J., Hankus, L., and Jurka, J. (2006) Annotation, submission and screening of repetitive elements in Repbase: RepbaseSubmitter and Censor. *BMC Bioinformatics* **7,** 474.
8. Milosavljevic, A., and Jurka, J. (1993) Discovering simple DNA sequences by the algorithmic significance method. *Comput Appl Biosci* **9,** 407–11.
9. Smit, A. F. A., Hubley, R., and Green, P. (1996–2004) RepeatMasker Open-3.0 http://www.repeatmasker.org
10. Britten, R. J., and Kohne, D. E. (1968) Repeated sequences in DNA. Hundreds of thousands of copies of DNA sequences have been incorporated into the genomes of higher organisms. *Science* **161,** 529–40.
11. Morgulis, A., Gertz, E. M., Schaffer, A. A., and Agarwala, R. (2006) WindowMasker: window-based masker for sequenced genomes. *Bioinformatics* **22,** 134–41.
12. Bao, Z., and Eddy, S. R. (2002) Automated de novo identification of repeat sequence families in sequenced genomes. *Genome Res* **12,** 1269–76.
13. McCarthy, E. M., Liu, J., Lizhi, G., and McDonald, J. F. (2002) Long terminal repeat retrotransposons of Oryza sativa. *Genome Biol* **3,** RESEARCH0053.
14. McCarthy, E. M., and McDonald, J. F. (2003) LTR_STRUC: a novel search and identification program for LTR retrotransposons. *Bioinformatics* **19,** 362–7.
15. Rho, M., Choi, J. H., Kim, S., Lynch, M., and Tang, H. (2007) De novo identification of LTR retrotransposons in eukaryotic genomes. *BMC Genomics* **8,** 90.
16. Yang, G., and Hall, T. C. (2003) MAK, a computational tool kit for automated MITE analysis. *Nucleic Acids Res* **31,** 3659–65.
17. Quesneville, H., Bergman, C. M., Andrieu, O., Autard, D., Nouaud, D., Ashburner, M., and Anxolabehere, D. (2005) Combined evidence annotation of transposable elements in genome sequences. *PLoS Comput Biol* **1,** 166–75.
18. Gish, W. (1996–2004) WU-BLAST http://blast.wustl.edu
19. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* **25,** 3389–402.
20. Green, P. (1994–1999) PHRAP and CROSS_MATCH http://www.phrap.org/phredphrap/phrap.html
21. Smith, T. F., and Waterman, M. S. (1981) Identification of common molecular subsequences. *J Mol Biol* **147,** 195–7.
22. Bedell, J. A., Korf, I., and Gish, W. (2000) MaskerAid: a performance enhancement to RepeatMasker. *Bioinformatics* **16,** 1040–1.