

# **Automated *de novo* Identification of Repeat Sequence Families in Sequenced Genomes**

**Zhirong Bao and Sean R. Eddy<sup>1</sup>**

*Howard Hughes Medical Institute and*

*Department of Genetics*

*Washington University School of Medicine*

*St. Louis, MO 63110 USA*

**running title:** *De novo* Repeat Identification

**key words:** repetitive sequences, genome analysis.

---

<sup>1</sup>To whom correspondence should be addressed. Tel: +1 314 362 7666; Fax: +1 314 362 7855; Email: eddy@genetics.wustl.edu.

### **Abstract**

Repetitive sequences make up a major part of eukaryotic genomes. We have developed an approach for the *de novo* identification and classification of repeat sequence families, based on extensions to the usual approach of single linkage clustering of local pairwise alignments between genomic sequences. Our extensions use multiple alignment information to define the boundaries of individual copies of the repeats and to distinguish homologous but distinct repeat element families. When tested on the human genome, our approach was able to properly identify and group known transposable elements. The program, RECON, should be useful for first-pass automatic classification of repeats in newly sequenced genomes.

A significant fraction of almost any genome sequence is repetitive. Repetitive sequences fall primarily into three classes – local repeats (tandem repeats and simple sequence repeats), families of dispersed repeats (mostly transposable elements and retrotransposed cellular genes) and segmental duplications (duplicated genomic fragments). The role of repeated, transposed, and duplicated sequence in evolution is an interesting and controversial topic (Doolittle and Sapienza 1980; Orgel and Crick 1980; McClintock 1984), but repetitive sequences are so numerous that simply annotating them well is an important problem in itself. This is particularly the case for repeat sequence families, which often carry their own genes (transposases, reverse transcriptases, and the like), and can confuse large-scale gene annotation.

Computational tools have been developed for systematic genome annotation of repeat families. Perhaps the best known is the program RepeatMasker (A.F.A. Smit and P. Green unpublished), which uses pre-compiled representative sequence libraries to find homologous copies of known repeat families. RepeatMasker is indispensable in genomes where repeat families have already been analyzed. However, it does not pass the “platypus test”: repeat families are largely species-specific, so if one were to analyze a new genome (like the platypus), a new repeat library would first need to be manually compiled. With sequencing efforts moving towards large-scale comparative genome sequencing of a wide variety of organisms, it is desirable to also have a *de novo* method that automates the process of compiling RepeatMasker libraries.

Several *de novo* approaches have been attempted, with limited success. They generally start with a self-comparison with a sequence similarity detection method to identify repeated sequence, then use a clustering method to group related sequences into families (Agarwal and States 1994; Parsons 1995; Kurtz et al. 2000). Detecting repetition by sequence alignment methods is relatively easy. Automatically defining biologically reasonable families is more difficult. Local sequence alignments do not usually correspond to the biological boundaries of the repeats, due to degraded or partially deleted copies, related but distinct repeats, and segmental duplications covering more than one repeat. Difficulty in defining element boundaries then causes a variety of subsequent problems in clustering related elements into families.

Similar problems arise in automated detection of conserved protein domains. Curated databases such as Pfam (Bateman et al. 2000) play a role equivalent to RepeatMasker by providing pre-compiled libraries of known domains. Automated clustering approaches are used to help detect new domains (Sonnhammer and Kahn 1994; Gracy and Argos 1998). These automated algorithms combine pairwise alignments with a variety of extra information to try to define biologically meaningful domain boundaries: most importantly, they look at multiple sequence alignments, not just pairwise alignments, in order to find significantly conserved

boundaries.

Here, we describe an automated approach for *de novo* repeat identification. Our approach uses multiple alignment information to infer element boundaries, and also to infer biologically reasonable clustering of sequence families.

## Results

Given a set of genomic sequences,  $\{S_n\}$ , our goal is to identify the repeat families therein (denoted by  $\{F_\alpha\}$ ), so that each family corresponds to a particular type of repeat, containing all and only copies of that repeat in  $\{S_n\}$ . Each individual repeat is a subsequence  $S_n(s_k, e_k)$ , where  $s_k$  and  $e_k$  are start and end positions in sequence  $S_n$ . Therefore, the output is  $\{F_\alpha = \{S_n(s_k, e_k)\}\}$ .

We define the following terms – *element*, *image* of an element, and *syntopy*. An individual copy of a repeat,  $S_n(s_k, e_k)$ , is called an *element*. A subsequence involved in an alignment is called an *image* (Fig 1). An element is the biological entity we are trying to infer. Images are observations from a pairwise comparison of the genome sequences  $\{S_n\}$ . One element forms many images, due to its repetitive nature. We call two images of the same element *syntopic images* (*syntopy* is a neologism from *syn* - 'same', *-topy* 'site'). Because observed alignments may extend well beyond the bounds of an element, and may even include unrelated elements (for example, because of segmental duplication or coincidental juxtaposition of abundant repeats), syntopy cannot be inferred just by image overlap – and this is the problem we must address.

### *The Existing Single Linkage Clustering Algorithms*

The existing *de novo* repeat identification algorithms can be summarized in our terms as single linkage clustering algorithms, as follows:

1. Obtain pairwise local alignments between sequences in  $\{S_n\}$ .
2. Define elements  $\{S_n(s_k, e_k)\}$  from the obtained alignments, or, images:
  - (a) Construct graph  $G(V, E)$ , where  $V$  represents all the images and  $E$  represents the syntopy between images. Two images are considered syntopic if they overlap, regardless of strand, beyond some threshold.
  - (b) Find all connected components in  $G$  (Skiena 1997).

- (c) For each connected component, define an element  $S_n(s_k, e_k)$  as the shortest fragment that covers all images in the component.
3. Group defined elements into families on the basis of their sequence similarity:
- (a) Construct graph  $H(V', E')$ , where  $V'$  represents all the elements, and  $E'$  represents similarity (two elements are connected by an edge if they form alignments in step 1).
  - (b) Find all connected components of  $H$ .
  - (c) For each connected component, define a family as the set of all elements in the component.
  - (d) (Unify the sequence direction of the elements in each family, based on their alignments with an arbitrarily chosen member in the family - this choice may not be clearcut for elements with internal inverted repeat structures that are imperfectly palindromic.)

***Problem 1: Inference of syntopy***

The main problems with this approach arise from the use of overlap to infer syntopy. If all repeat elements were full-length, well-conserved, and well-separated by unique sequence in the genome, all syntopic images would be equivalent to their corresponding element, and single linkage clustering would work fine. However, two major phenomena distort this ideal picture. One is drift (both deletion and substitution mutation), which causes partial images (Fig 2B). The other is segmental duplication and juxtaposition of common repeats which produce images containing more than one element (Fig 2C).

Various strategies have been suggested for inferring syntopy from image overlap. Two typical measurements, termed *single coverage method* and *double coverage method*, require the overlap to be longer than a certain fraction of *either* or *both* of the images, respectively. When overlapping images are of different length, the two methods make different inference of syntopy which leads to different definitions of elements (Fig 2A). The single coverage method is suitable for the scenario in Fig 2B, while the double coverage is suitable for that in Fig 2C.

However, either strategy leads to errors. When the double coverage method is applied to partial images (Fig 2B), it yields many spurious, overlapping elements for one true biological copy. When the single coverage method is applied to multielement images (Fig 2C), it yields a composite element corresponding to the whole segmental duplication, which will lump families together later in family definition. Simply tuning the thresholds of these methods will not solve the problem; the two biological scenarios require opposite measurements of overlap in order to correctly infer syntopy (Agarwal and States 1994; I. Holmes

pers. comm.). Furthermore, since these algorithms use only pairwise relationships between images, they are not able to distinguish the two biological scenarios and choose the proper criterion. The example in Fig 2 therefore suggests that no algorithm of this type can work.

However, one also sees in Fig 2 that there is useful information in the pattern of the multiple alignment of the images. In both cases, most image endpoints agree on the boundaries of an independent repeat. The key distinction lies in the endpoints of the shorter images. In Fig 2B, these endpoints are quasi-randomly dispersed throughout the multiple alignment, whereas in Fig 2C, the endpoints pile up. Biologically, this distinction will hold true so long as the independent replication of repeats is more frequent than the generation of composite elements (say, by segmental duplications), and deletion is a random process, which are usually (but not always) the case.

Our approach to the problem is based on the above observation (Fig 3). After an initial definition using the single coverage method, elements are split according to significant aggregations of image endpoints. As shown in Fig 3, a composite element will be split into several pieces (right panel, five pieces in this case), while a full-length element will be preserved (left panel). Details are specified in the **element re-evaluation and update procedure** (see Methods).

Certain images complicate the above splitting process, such as those formed between related but distinct elements (Fig 4A and B), which may lead to an incorrect splitting of an element. Unlike those in Fig 2, these misleading image endpoints do not occur at the termini of either of the two elements involved (Fig 4C, open circles). We use this difference to filter the misleading endpoints prior to the element re-evaluation and update procedure (see **image end selection rule** in Methods).

### ***Problem 2: Inter-family similarity***

Many repeat families are evolutionarily related (for example, the autonomous *C. elegans* Tc1 DNA transposons and the smaller nonautonomous Tc7 elements, Fig 4). Although the reality is that repeats, like Pfam's protein domain families or biological species, are a hierarchical evolutionary continuum that defies classification, it is still desirable to impose a simplistic classification that pretends that repeat families are distinct, for the purpose of practical genome annotation. Since related families may form significant sequence alignments, we will have to impose arbitrary criteria to avoid lumping related but "distinct" families together.

We consider two elements to be distinct if the length of the non-conserved regions adds up to more than certain ratio of both of the two sequences (Fig 4C, dashed lines). The **family relationship determination**

**procedure** (see Methods) implements this definition. When constructing the graph for clustering (step 3 in the above algorithm), elements belonging to the same family are linked with *primary* edges, and those belonging to different families but still forming significant alignments are linked with *secondary* edges. Families (connected components) are defined by primary edges.

Incorrect primary edges can arise in the presence of certain partially deleted elements (Fig 5A). As shown in Fig 5B, primary and secondary edges are properly constructed between full-length copies of Tc1 and Tc7 by the family relationship determination procedure. However, edges between the partial copy of Tc7 and the Tc1s are rendered primary, as there are no non-conserved regions in this Tc7 compared to Tc1s. These false primary edges will lump the two families. Such a situation can be recognized by finding triangles involving two primary edges and a secondary edge (e.g. Tc1-2=>Tc7-1=>Tc7-partial). Once an element yielding incorrect primary edges is recognized, all its primary edges are removed except for the one linking to its most closely related element (Fig 5C). More rules are specified in the **family graph construction procedure with edge re-evaluation** (see Methods).

## The RECON Algorithm

Our algorithm is summarized as follows:

1. Obtain pairwise local alignments between the input sequences.
2. Define elements from the obtained alignments:
  - (a) Elements are first defined using the *single coverage method*, as described in step 2 of the existing algorithm;
  - (b) Each element defined is re-evaluated following the *image end selection rule* (Fig 4) and the *element re-evaluate and update procedure* (Fig 3);
  - (c) If an element defined is considered composite and is split, elements forming alignments with the composite element will be re-evaluated. The process continues till all definitions of elements stabilize.
3. Group elements defined into families on the basis of their sequence similarity:
  - (a) Elements and their family relationship are determined and converted to a graph  $H(V', E')$  according to the *family relationship determination procedure* and the *graph construction procedure with edge re-evaluation* (Fig 5);

- (b) Find all connected components of  $H$  according to the primary edges constructed. For each connected component, define a family as a set of all elements in the component.

The algorithm has been implemented as RECON, a set of C programs and Perl scripts. The RECON package, including a demo and more materials, is available at <http://www.genetics.wustl.edu/eddy/recon/>.

## Assessment

In order to assess the performance of RECON, we used it to analyze a random sample of 3 Mb, or about 0.1%, of the human genome (Lander et al. 2001), and compared the results to RepeatMasker annotation as a “gold standard”. For purpose of comparison, we also implemented and tested the basic single linkage clustering algorithm using both the single or double coverage element definition methods. All three *de novo* methods use the same set of 453,896 pairwise alignments generated by WU-BLASTN (W. Gish unpublished) (see Methods).

It took RECON 4 CPU hours and a maximum of 300 MB RAM to analyze this set of alignments on a single Intel Xeon 1.7GHz processor. A RECON analysis of a set of alignments from a three-fold larger sample (9 Mb) took 39 CPU hours and 750 MB RAM. We cannot give a useful asymptotic analysis of memory/cpu usage in terms of genome or sample size, because RECON’s computational complexity is strongly dependent on repeat density and composition. For example, an analysis of the alignments from the same 3 Mb sample with known repeats masked out by RepeatMasker took less than 1 minute and 900 KB of RAM. This suggests that for a large, repeat-rich genome, it will be possible (and necessary) to carry out an iterative RECON analysis; e.g., first find the most abundant families in a small sample of the genome, then analyze progressively larger samples after masking element families that have already been confidently identified.

As to the quality of the results, we first looked specifically at the definition of Alu, which is the most numerous repeat element and therefore the most prone to many sorts of clustering artifacts (Table 1). We identified each *de novo* constructed family that contained one or more sequences that overlapped Alu elements defined by RepeatMasker. For the largest family defined by each method, we also counted how many of the defined elements contained non-Alu repeat sequences as defined by RepeatMasker. A “correct” result would be that a *de novo* method would identify a single family of 1,260 Alu elements covering 318,927 bases of the genome sample, exactly matching the RepeatMasker annotation.

The single coverage method defined 1,389 elements which overlapped the Alus defined by RepeatMasker. The number is larger than 1,260 because some Alu copies are broken into several fragments by

the method. The 1,389 elements covered too much of the genome (331,593 bp), because some of the “elements” are actually segmental duplications which happen to contain Alus. This method overclusters. In the largest family defined, it mixed 576 of non-Alu sequences (most of which are L1 elements, the second most abundant human repeat family) with the 1,389 Alu elements. The double coverage method underclusters images, defining many “elements” that completely overlap each other, leading to a huge number of “elements” (56,925) clustered into too many families (19). RECON minimizes both problems, leading to 2 Alu-containing families (one of which dominates) with 1,468 elements covering 285,000 bp, with minimal contamination from other repeat families. Some Alu elements are still inappropriately broken into two or more fragments (leading to significantly more than 1,260 elements). The somewhat lower genomic coverage of RECON compared to RepeatMasker results from the higher sensitivity of RepeatMasker’s similarity search algorithm and threshold (CROSSMATCH with an aggressive threshold, as opposed to RECON’s use of WU-BLAST with a conservative threshold).

In order to evaluate how reliable RECON annotation is overall, we systematically compared every RECON family containing  $\geq 10$  elements to RepeatMasker annotation (Table 2). Each RECON family was labeled according to which RepeatMasker annotation made up the majority of its elements. Any element that was annotated as a different family or not annotated at all was considered as false positive elements (cluster fp1 and cluster fp2 columns in the Table, respectively). These results suggest that RECON’s families are almost completely “pure”, with very little contamination from unrelated repeat families. The families are slightly underclustered; for example, one large family with the majority of the L1 elements is found (f7), along with several smaller families of partial L1 elements (f8, f13, f22, f57, f146) which are not clustered with f7. f179, a “new” family, is a family of retroposed protein-coding genes, which are a class of repeats not annotated by RepeatMasker.

An important usage of a *de novo* method is to generate repeat libraries for the incremental analysis of a genome. In order to evaluate how useful RECON families would be for genome annotation of elements in a subsequent sample of human sequence, we compared the consensus sequence of each RECON family to their most similar sequences used in RepeatMasker (Table 2; see Methods). Bases in RECON’s consensus that are not in RepeatMasker’s sequence are counted as false positives (consensus fp column) – measuring to what extent RECON defines too large of a consensus element. Bases in RepeatMasker’s sequence that are not in RECON’s consensus are counted as false negatives (consensus fn column) – measuring to what extent RECON only recovers part of the consensus element. For four out of the six known transposable elements found, the cononical sequence is reconstructed essentially intact (f1/Alu, f7/L1, f46/MaLR and

f28/MER41). For Tigger1 and MER1, however, only part of the canonical sequence is recovered in families f17 and f156. Manual inspection suggests that it is due to the truly fragmented nature of the copies in our sample, rather than erroneous splittings by RECON.

The canonical Alu sequence is dimeric, containing a left (L) and a right (R) monomer (Jurka and Zuckerkandl 1991). Interestingly, the consensus sequence identified by RECON family f1 contains exactly one and a half Alu elements, in the configuration LLR. The longest six elements in f1 are all in this configuration. Such trimeric Alus have been noted before (Perl et al. 2000), and RECON's annotation suggests that they have been actively transposed in the human genome.

## Discussion

The problem of automated repeat sequence family classification is inherently messy and ill-defined, and does not appear to be amenable to a clean algorithmic attack. The heuristic approach we have taken in RECON appears to be satisfactory for many practical purposes. Our use of multiple sequence alignment information, specifically the clustering of observed alignment endpoints, is a significant improvement over single linkage clustering based on pairwise sequence relationships alone.

The evaluation of RECON's performance suggests several issues which could use improvement. It slightly underclusters elements, failing to appropriately link some small fragmentary families to a large full-length family. This might be addressed by a post-processing step that merges RECON families when the consensus of one family covers the consensus of the other.

RECON is sometimes unable to recover a highly fragmented family in one piece. To overcome this, we could employ a statistical test to identify RECON families whose copies tend to be physically adjacent to each other. The more diverged families, such as the ancient human L2 family, was not recovered in our test, due to the chosen sensitivity settings of WU-BLAST.

RECON can also fail when its simple assumptions about alignment end clustering are violated. For example, when a particular form of partial copy is generated preferentially (e.g., solo LTRs for retrovirus-like elements (Kim et al. 1998), formed by high-frequency deletion between the directly repeated LTRs), it can lead to an erroneous splitting of the full-length copies. Also, if a particular combination of repeat elements can itself be duplicated at high frequency (e.g., composite bacterial IS elements (Berg et al. 1989)), it may not be recognized as composite.

We envision using RECON as a tool for initial analysis of a genome sequence. Much like automated PRODOM protein domain family identification aids curated Pfam multiple alignment construction, the fam-

ilies identified by RECON can be the basis of a higher quality level of analysis, such as using RECON families to build a RepeatMasker library, or using RECON multiple alignments to build a library of profile hidden Markov models.

## Methods

### Components of RECON

#### *Image End Selection Rule*

This rule filters misleading images (Fig 4) by considering the length and arrangement of the aligned and unaligned sequences between two elements as follows:

1. For each pair of defined elements that form alignments, find all maximal groups of alignments in which all alignments are part of one (but not necessarily the optimal) global alignment of the two given elements. This is done by finding maximal cliques (Skiena 1997) in a graph where the vertices represent the alignments and two vertices are linked if the two corresponding alignments can be seen as part of one global alignment of the two given elements.
2. For each group found above: order the alignments according to their coordinates; eliminate the group if the sequences outside the out-most alignment or between any two adjacent alignments in the group are longer than a given length cutoff in *both* elements; if not eliminated, assign a score to the group as the sum of scores of all alignments in the group. The length cutoff is chosen so that sequences shorter than the cutoff can be considered as generated by the random extension of true alignments by the pairwise alignment tool.
3. If more than one group remains, take the one with the highest score and discard the others. Ends of the images in the remaining group (if any) are collected for further analysis.

#### *Element Re-evaluation and Update Procedure*

This procedure updates the definition of a given element (Fig 3) by evaluating the aggregation of image endpoints collected according to the rule above.

1. Choose a length cutoff so that sequences shorter than the cutoff are considered as generated by the random extension of true alignments by the pairwise alignment tool.

2. Slide a window of the chosen length cutoff along the given element. Within each window, cluster the collected image ends as follows: seed a cluster with the leftmost end not yet clustered; if an end is within certain distance to any member in the cluster, it is assigned to the cluster; when no more ends can be assigned to the cluster, start a new cluster if necessary, till all ends in the window are clustered.
3. For each cluster found above, let  $n$  denote the number of ends in the cluster,  $c$  denote the mean position of these  $n$  ends, and  $m$  denotes the number of images of the given element spanning position  $c$ . If  $n/m$  is greater than a given threshold,  $c$  is considered a significant aggregation point.
4. If no significant aggregation point is accepted, the original definition of the given element is maintained.
5. Otherwise, update the given element as follows: split the element and its alignments at the aggregation points; discard the original definition of the given element; discard the split products (new elements and alignments) that are shorter than the chosen length cutoff at the beginning; assign alignments to proper new elements.
6. If more than one new element remains, the original element is considered composite.

#### ***Family Relationship Determination Procedure***

This procedure determines for a given pair of defined elements that form alignments, whether the two belong to the same family, or to two related but distinct families (Fig 4). The procedure, which considers the relative length of the aligned sequences compared to the length of the elements, is as follows:

1. For a given pair of defined elements that form alignments, find all maximal groups of alignments in which all alignments are part of one (but not necessarily the optimal) global alignment of the two given elements. See step 1 in the image end selection rule for detail.
2. The total length of each group found above is calculated as the sum of the length of all alignments in the group. The longest total length among the groups is treated as the alignable length between the two elements.
3. If the alignable length is longer than a certain fraction of the length of *either* element, the two elements are considered to belong to the same family. Otherwise, not.

### ***Family Graph Construction Procedure with Edge Re-evaluation***

1. Each element defined is represented by a vertex.
2. Edges are constructed as follows: if two elements are considered to belong to the same family by the family relationship determination procedure, a *primary* edge is constructed between the two corresponding vertices; if two elements form significant alignments but do not belong to the same family, a *secondary* edge is constructed between the two; if two elements do not form significant alignments, no edge is constructed between the two.
3. For each vertex  $v$ , its primary edges are re-evaluated as follows (Fig 5): Let  $N(v)$  denote the set of vertices directly connected to  $v$  via primary edges. If any pair in  $N(v)$  are connected by a secondary edge, then  $\forall v' \in N(v)$ , the primary edge between  $v$  and  $v'$  is removed unless  $v'$  is the most closely related element to  $v$  in  $N(v)$  (based on alignment score and/or percent identity) or  $v$  is the most closely related element to  $v'$  in  $N(v')$ . In the latter case, the primary edges of  $v'$  will be updated as just described.
4. Remove all secondary edges.

### **Implementation details**

RECON starts from a datafile containing pairwise alignments, which allows a user to choose a tool other than WU-BLAST to do the initial all-vs-all comparison of the genome to itself.

A major issue is memory usage. To avoid holding all alignments from a genome-scale analysis in RAM at once, RECON manipulates files on disk (including a separate file for each currently defined element). It is therefore extremely I/O intensive.

RECON is not useful for processing short-period tandem repeats; these are split down to shorter forms or even monomers, a process which can take many iterations to converge. To improve time efficiency, we filter these by ignoring the initially defined elements which have more than 1,000 images and where the number of partner elements is less 1/5 of the number of images. Furthermore, since we discard short elements generated during splitting (element re-evaluation and update procedure), the whole family can suddenly disappear when it falls below the minimum element length cutoff.

Besides the threshold and parameter choices in the initial pairwise comparison, RECON has four tunable parameters:

- The cutoff for fractional overlap between images which is used in the initial inference of syntopy by the single coverage method. (Default = 0.5.)
- The minimum length of an element, e.g. the maximal length that we expect the pairwise alignment tool to spuriously extend by chance from a true element boundary, used in the image end selection rule and the element re-evaluation and update procedure. (Default = 30 nt.)
- The ratio cutoff for splitting an element at a given position, used in the element re-evaluation and update procedure. (Default = 2.)
- The minimal fraction of alignable sequences between two elements before they are considered to belong the same family. (Default = 0.9.)

These parameters were optimized by looking at RECON classification of four experimentally verified DNA transposons (Tc1, Tc2, Tc3, and Tc5 (Plasterk and von Luenen 1997)) in the *Caenorhabditis elegans* genome sequence (The *C. elegans* Sequencing Consortium 1998). The human genome is dominated by retro-transposons (Alu, L1 and MaLR) and old, fragmented DNA transposons (Lander et al. 2001), and these families yield different patterns in multiple alignments than the young DNA transposons in the *C. elegans* training set, so the test on human data was reasonably independent of our training of these few parameters.

## Human Genome Analysis

3 MB of sequence was randomly sampled as 20Kb chunks from the 796 contigs in the Dec 12, 2000 release of the human genome (Lander et al. 2001) (<http://genome-test.cse.ucsc.edu/goldenPath/12dec2000/bigZips>). All-vs-all comparison of the sampled sequences was done using WU-BLASTN 2.0 (W. Gish unpublished) (<http://blast.wustl.edu>) with options M=5 N=-11 Q=22 R=11 -kap E=0.00001 wordmask=dust wordmask=seg maskextra=20 -hspmax 5000.

Known repeats were identified using the 7 July 2001 version of RepeatMasker (<http://ftp.genome.washington.edu/RM/RepeatMasker.html>), with default options.

Consensus sequences of RECON families were made by aligning the ten longest members of the family with DIALIGN2 (Morgenstern 1999), with default options, then selecting a simple majority rule consensus residue for each column.

## Acknowledgements

We thank Dr. Elena Rivas for discussions and advice. The sequence sampling tool was provided by Mr. Robert Klein. We gratefully acknowledge financial support from the National Science Foundation (grant no. DBI-0077709) and the Howard Hughes Medical Institute.

## References

- Agarwal, P., States, D.J. 1994. The Repeat Pattern Toolkit (RPT): analyzing the structure and evolution of the *C. elegans* genome. In *Proc Int Conf Intell Syst Mol Biol* (ed. Altman, R. et al), pp. 1–9. AAAI Press, Menlo Park, California.
- Altschul, S.F. 1991. Amino acid substitution matrices from an information theoretic perspective. *J Mol Biol* **219**: 555–565.
- Bateman, A., Birney, E., Durbin, R., Eddy, S.R., Howe, K.L., and Sonnhammer, E.L.L. 2000. The Pfam protein families database. *Nucleic Acids Res* **28**: 263–266.
- Benson D.A., Karsch-Mizrachi I., Lipman D.J., Ostell J., Rapp B.A., and Wheeler D.L. 2000. Genbank. *Nucleic Acids Res* **28**: 15–18.
- Berg, D.E., Howe, M.M. 1989. *Mobile DNA*. American Society for Microbiology, Washington, D.C..
- Doolittle, W.F., Sapienza, C. 1980. Selfish genes, the phenotype paradigm and genome evolution. *Nature* **284**: 601–603.
- Gracy, J., Argos, P. 1998. Automated protein sequence database classification. II. delineation of domain boundaries from sequence similarities. *Bioinformatics* **14**: 174–187.
- Jurka, J., Zuckerkandl, E. 1991. Free left arms as precursor molecules in the evolution of Alu sequences. *J Mol Evol* **33**: 49–56.
- Kim, J.M., Vanguri, S., Boeke, J.D., Gabriel, A., and Voytas, D.F. 1998. Transposable elements and genome organization: a comprehensive survey of retrotransposons revealed by the complete *Saccharomyces cerevisiae* genome sequence. *Genome Res* **8**: 464–478.

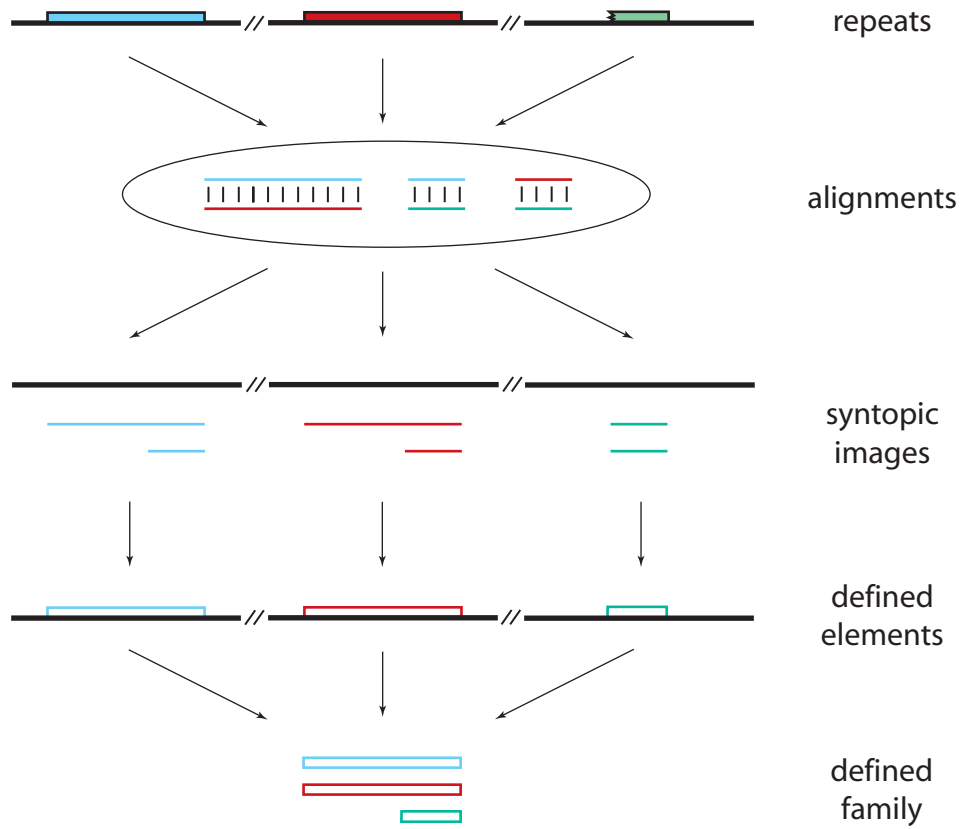
- Kurtz, S., Ohlebusch, E., Schleiermacher, C., Stoye, J., and Giegerich, R. 2000. Computation and visualization of degenerate repeats in complete genomes. In *Proc Int Conf Intell Syst Mol Biol* (ed. Altman, R. et al), pp. 228–238. AAAI Press, Menlo Park, California.
- Lander, E.S., Linton, L.M., Birren, 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.
- Marshall, E. 2001. Genome teams adjust to shotgun marriage. *Science* **292**: 1982–1983.
- McClintock, B. 1984. The significance of responses of the genome to challenge. *Science* **226**: 792–801.
- Morgenstern, B. 1999. DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics* **15**: 211–218.
- Orgel, L.E., Crick, F.H. 1980. Selfish DNA: the ultimate parasite. *Nature* **284**: 604–607.
- Parsons, J.D. 1995. Miropeats: graphical DNA sequence comparisons. *Comput Appl Biosci* **11**: 615–619.
- Perl, A., Colombo, E., Samoilova, E., Butler, M.C., and Banki, K. 2000. Human transaldolase-associated repetitive elements are transcribed by RNA polymerase III. *J Biol Chem* **275**: 7261–7272.
- Plasterk, R.H.A., von Luenen, H.G.A.M. 1997. Transposons. In *C. elegans II* (ed. Riddle, D. et al), pp. 97–116. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, New York.
- Skiena, S.S. 1997. *The algorithm design manual*. Telos/Springer-Verlag, New York.
- Sonnhammer, E.L., Kahn, D. 1994. Modular arrangement of proteins as inferred from analysis of homology. *Protein Sci* **3**: 482–492.
- The C. elegans Sequencing Consortium. 1998. Genome sequence of the nematode C. elegans: a platform for investigating biology. *Science* **282**: 2012–2018.

### Web Site References

Gish, W. WU-BLAST, <http://blast.wustl.edu>

Smit, A.F.A., Green, P. RepeatMasker,

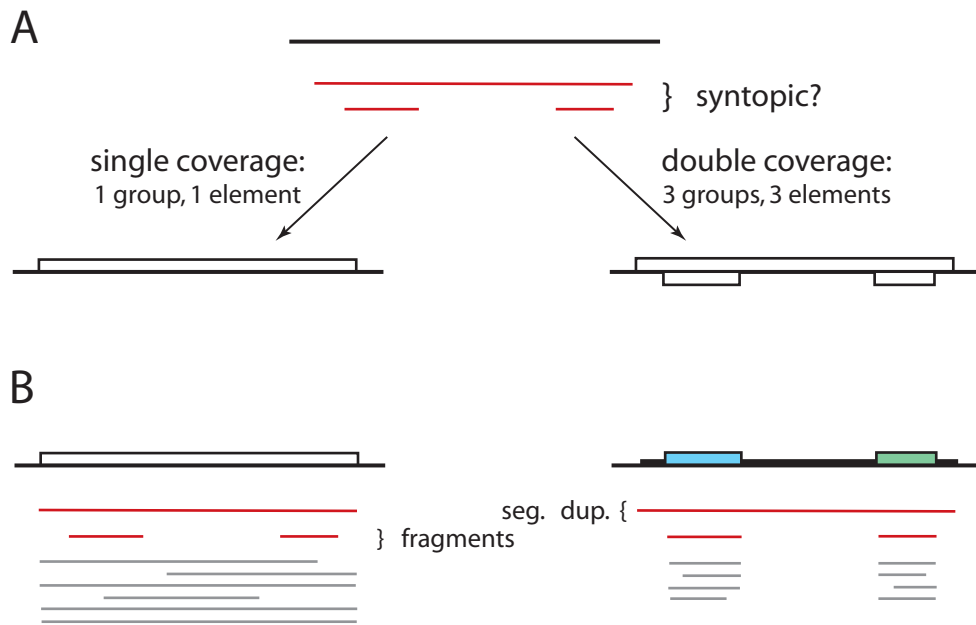
<http://ftp.genome.washington.edu/RM/RepeatMasker.html>



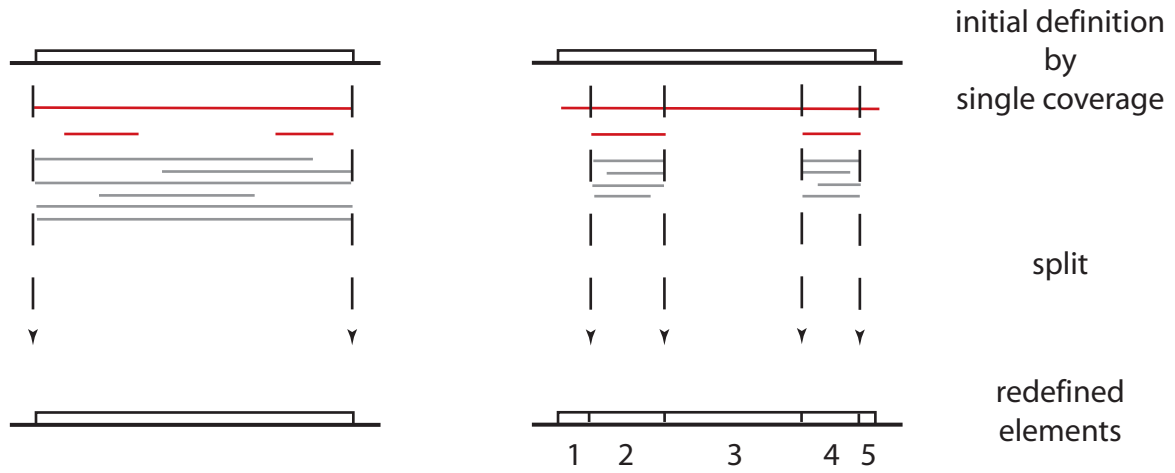
UCSC Human Genome Project Working Draft,

<http://genome-test.cse.ucsc.edu/goldenPath/12dec2000/bigZips>

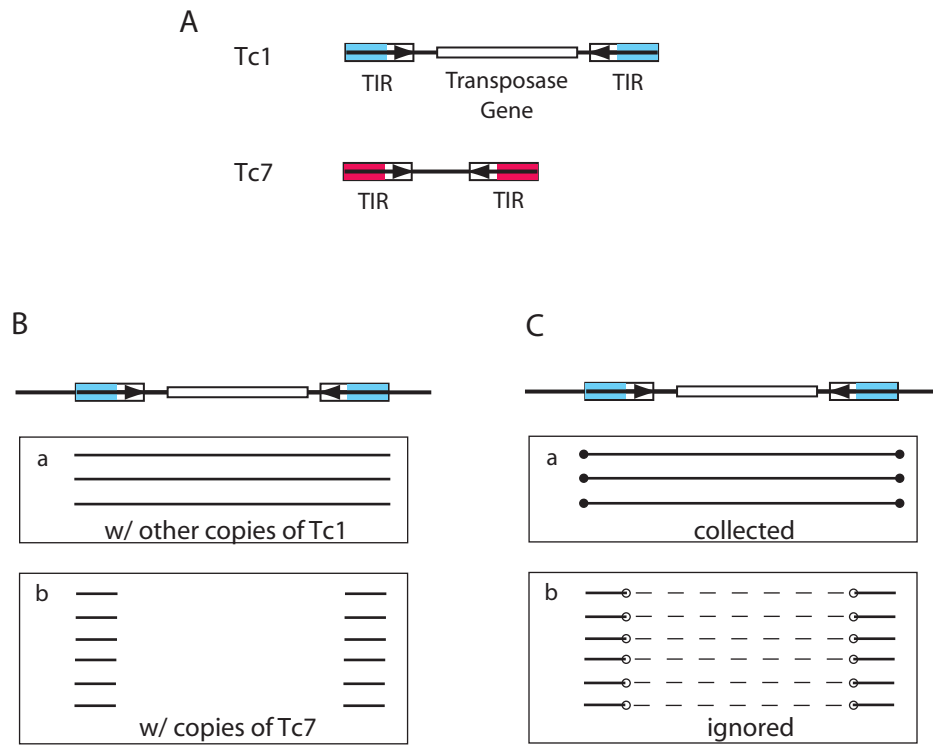
**Figure 1**



**Figure 2**



**Figure 3**



**Figure 4**

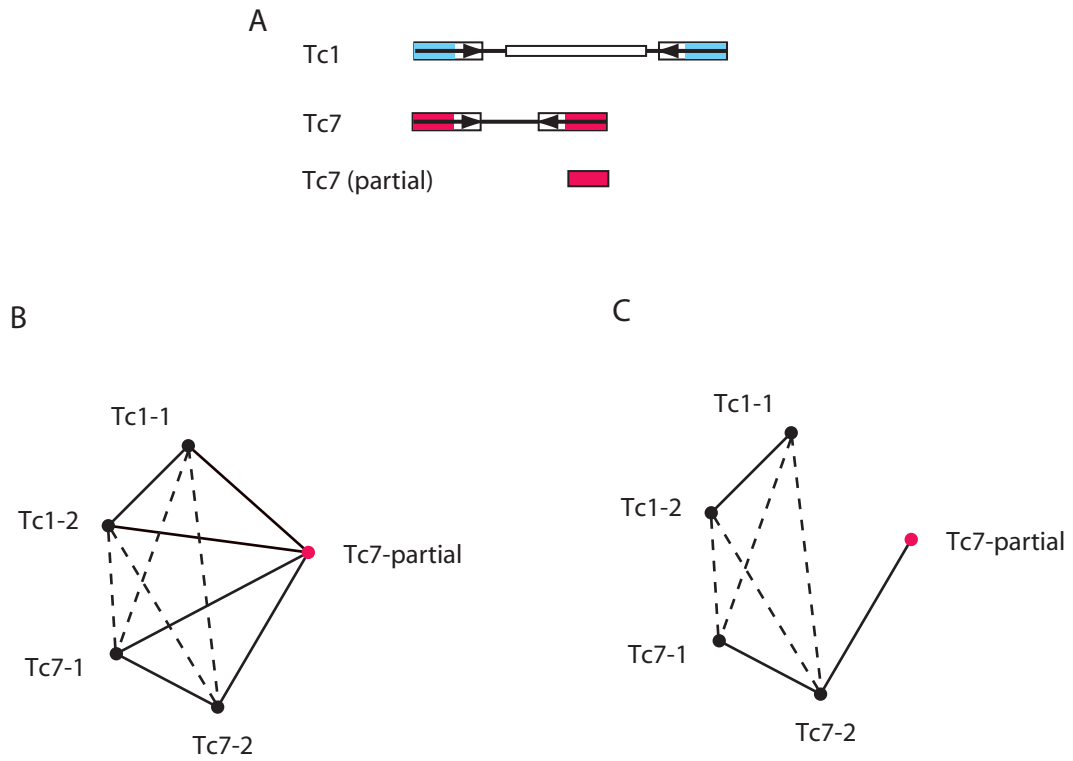


Figure 5

Figure 1: Flowchart of the de novo strategy. Input genomic sequences (black lines on top) contain a family of repeats with three copies (i.e., elements); two full-length (blue and red boxes) and one partially deleted (green box). These elements, unknown at this point, will yield three alignments in an all vs all pairwise comparison of the genomic sequences. The aligned fragments (i.e., images), colored as their corresponding elements for clarity, are sorted to their corresponding genomic region, and those coming from the same element (i.e., syntopic images) can be grouped together according to their overlaps. Based on the syntopic sets, elements can be defined. These defined elements are then clustered into one family as they are all similar to each other.

Figure 2: Different biological scenarios require different methods of syntopy inference. A. For three images (red lines) in a genomic region (top black line), the single and double coverage methods lead to different definitions of elements. B. A full-length element and its images (red and grey lines below). The top long image is formed with another full-length member in its family, while the shorter images are formed with the fragmented members. C. A segmental duplication covering two kinds of elements (blue and green). The top long image is formed with the other copy of this segmental duplication, while the shorter images are formed with other members in the blue and green families, respectively.

Figure 3: The RECON algorithm uses the aggregation of endpoints in the multiple alignment of images to distinguish between different biological scenarios.

Figure 4: Complications due to sequence similarity between related families. A. The schematic structure of Tc1 and Tc7, two related DNA transposons which are similar at the end of their Terminal Inverted Repeats (colored), but not in the rest of the sequences (Plasterk and von Luenen 1997). B. A Tc1 element and its images. C. Images in B are filtered, and only those ends marked with closed circles will be collected to determine whether the element should be split. Open circles in box b mark the misleading ends. Dashed lines link the pairs of images formed with the same copy of Tc7 and represent the unalignable sequences between a Tc1 and a Tc7. Although not shown in the figure, the two TIRs of Tc1 also form alignments in the opposite strands, and images from these alignments are also filtered.

Figure 5: False primary edges due to partial elements. A. The schematic structure of full-length Tc1 and Tc7 (see also Fig 4) and a partially deleted Tc7, which preserves only the region similar to Tc1. B. Graph constructed for Tc1s and Tc7s. Black nodes represent full-length elements. Solid and dashed lines represent primary and secondary edges, respectively. C. Certain primary edges are removed from the partial Tc7 in order to eliminate the false ones.

Table 1: Definition of the Alu family.

Method	# of elements	Total length, bp	Genomic coverage, bp	# of families	Largest Family	
					non-Alus	Alus
RepeatMasker	1,260	318,938	318,927	1	0	1,260
Single	1,389	357,830	331,593	1	576	1,378
Double	56,925	7,908,428	330,830	19	6	54,615
RECON	1,468	285,747	285,000	2	2	1,423

Table 2: The Larger Human Repeat Families Defined by RECON

RECON family	RepeatMasker family	copy <sup>a</sup> number	cluster <sup>b</sup>		consensus <sup>c</sup>	
			fp1	fp2	fp	fn
<b>f1</b>	<b>Alu</b>	<b>1425</b>	<b>1</b>	<b>1</b>	<b>1/424</b>	<b>16/311</b>
f230	Alu	10	0	0	3/77	111/185
<b>f7</b>	<b>L1</b>	<b>292</b>	<b>2</b>	<b>1</b>	<b>0/6139</b>	<b>15/6152</b>
f8	L1	28	0	0	0/906	5391/6305
f13	L1	22	0	0	1/518	5668/6184
f22	L1	17	0	0	3/1481	4655/6146
f57	L1	14	0	0	1/690	5429/6146
f146	L1	13	0	0	2/273	6031/6305
f10	MaLR(LTR)	63	0	0	0/365	1/364
<b>f46</b>	<b>MaLR(LTR+internal)</b>	<b>44</b>	<b>0</b>	<b>0</b>	<b>3/2116</b>	<b>0/1935</b>
f12	MaLR(LTR)	17	0	0	3/211	218/426
<b>f28</b>	<b>MER41</b>	<b>18</b>	<b>0</b>	<b>0</b>	<b>2/559</b>	<b>1/554</b>
<b>f17</b>	<b>Tigger1</b>	<b>14</b>	<b>0</b>	<b>0</b>	<b>2/1021</b>	<b>1405/2418</b>
f179	New	13	0	13	n/a	n/a
<b>f156</b>	<b>MER1</b>	<b>10</b>	<b>0</b>	<b>0</b>	<b>3/199</b>	<b>99/297</b>

<sup>a</sup>number of defined elements in RECON family

<sup>b</sup>fp1: number of elements in RECON family corresponding to a different RepeatMasker family. fp2: number of elements in RECON family not annotated by RepeatMasker.

<sup>c</sup>fp: false positive positions vs length of the consensus. fn: false negative positions vs length of the RepeatMasker sequence. The consensi of the L1-corresponding families match different L1 sequences in RepeatMasker. So do the MaLR-corresponding families.