7002–7013 *Nucleic Acids Research, 2009, Vol. 37, No. 21* *Published online 28 September 2009*
doi:10.1093/nar/gkp759

Fine-grained annotation and classification of *de novo* predicted LTR retrotransposons

Sascha Steinbiss*, Ute Willhoeft, Gordon Gremme and Stefan Kurtz

Center for Bioinformatics, University of Hamburg, Bundesstraße 43, 20146 Hamburg, Germany

Received July 11, 2009; Revised and Accepted August 28, 2009

ABSTRACT

Long terminal repeat (LTR) retrotransposons and endogenous retroviruses (ERVs) are transposable elements in eukaryotic genomes well suited for computational identification. *De novo* identification tools determine the position of potential LTR retrotransposon or ERV insertions in genomic sequences. For further analysis, it is desirable to obtain an annotation of the internal structure of such candidates. This article presents *LTRdigest*, a novel software tool for automated annotation of internal features of putative LTR retrotransposons. It uses local alignment and hidden Markov model-based algorithms to detect retrotransposon-associated protein domains as well as primer binding sites and polypurine tracts. As an example, we used *LTRdigest* results to identify 88 (near) full-length ERVs in the chromosome 4 sequence of *Mus musculus*, separating them from truncated insertions and other repeats. Furthermore, we propose a work flow for the use of *LTRdigest* in *de novo* LTR retrotransposon classification and perform an exemplary *de novo* analysis on the *Drosophila melanogaster* genome as a proof of concept. Using a new method solely based on the annotations generated by *LTRdigest*, 518 potential LTR retrotransposons were automatically assigned to 62 candidate groups. Representative sequences from 41 of these 62 groups were matched to reference sequences with >80% global sequence similarity.

INTRODUCTION

A considerable part of the genomes of higher eukaryotic species are transposable elements (TE). In case of vertebrates and plants, half or even higher percentages of the genome are composed of TEs (1). According to their transposition mechanism, TEs are divided in DNA transposons and retrotransposons. The latter are further subdivided by structural features (2). One of the well-described retrotransposon subgroups are long terminal repeat (LTR) retrotransposons, which are characterized by LTR at their termini. In vertebrate species, the term endogenous retroviruses (ERVs) is commonly used as a synonym for LTR retrotransposons. Further classification into families is usually carried out by sequence comparison.

Several effects of LTR retrotransposons/ERVs on host genomes, e.g. on gene expression, alternative splicing and implications in diseases were described in the last decade [for a review see (3)]. In addition, several host defence mechanisms against retrotransposition, e.g. epigenetic silencing (4) and RNA interference were found [for a review see (5)].

To identify LTR retrotransposons in genome sequences, several software tools have been developed. These can be divided into homology-based and *de novo* (or *ab initio*) identification tools (6). Sequence homology-based tools, like *RepeatMasker* (http://www.repeatmasker.org), employ alignment algorithms for detecting sequence regions similar to members of a library of known repeats [e.g. *Repbase Update* (7)]. Such tools work best on genomes in which repeats have already been identified, as the sequences of a large number of TE families seem to be species specific. Consequently, reusing library entries from a particular species for detection in another species may not lead to satisfactory results. Furthermore, the primary focus of *RepeatMasker* is the masking of sequences rather than producing annotations. For example, *RepeatMasker* commonly produces several hits for a single TE insertion which are mostly only linked by the name of their *Repbase* reference entry, which can make it difficult to identify individual full-length LTR retrotransposon insertions from *RepeatMasker* results. In contrast, typical *de novo* identification tools (8–12) rely on the repetitive structure of the LTRs. In particular, they identify (possibly degenerated) repeated sequence pairs appearing in a certain distance from each other

*To whom correspondence should be addressed. Tel: +49 40 42838 7322; Fax: +49 40 42838 7312; Email: steinbiss@zbh.uni-hamburg.de

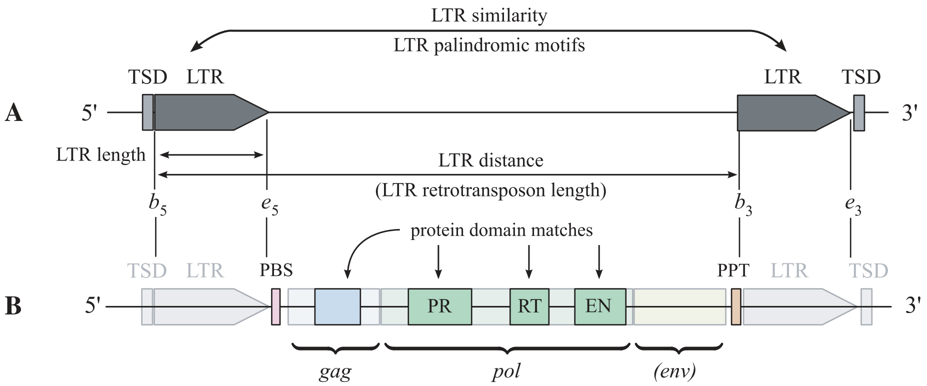


Figure 1. LTR retrotransposon model parameters. (**A**) Model of a single LTR retrotransposon used by *LTRharvest* (12). (**B**) Additional internal features considered in *LTRdigest*. The example depicts a PBS, a domain hit from the *gag* gene, three domain hits from the *pol* gene and a PPT. Of course, usually only a subset of the features is present in the region between the LTR pair. TSD, target site duplication; PR, protease; EN, endonuclease.

(Figure 1A). These are interpreted as putative LTRs of individual LTR retrotransposon insertions, called 'candidates' in this article.

However, LTRs are not the only relevant sequence features in LTR retrotransposons and ERVs. Other features located in their internal region (Figure 1B) are important for successful retrotransposition. First, polyproteins encoded in the internal genes *gag* and *pol* contain catalytic domains providing, for example, reverse transcriptase (RT), protease, ribonuclease H or integrase functions (13). Furthermore, a ∼8–22 bp long purine-rich region directly upstream of the inner 3′-LTR boundary, called polypurine tract (PPT), is needed as a primer for synthesis of the second (plus) DNA strand after reverse transcription (14,15). A uracil-rich region, called a U-box, is sometimes associated with this region and function (16). Finally, a ∼8–18 bp long primer binding site (PBS) motif near the inner 5′-LTR boundary is essential as a complementary hybridization partner for a transfer RNA acting as a primer for the RT encoded by the retrotransposon (14,17,18).

The dependence of a successful retrotransposition on the presence of these features implies that in an active or recently inactivated full-length LTR retrotransposon (retaining all genes and regulatory sequences), all these sequence features are expected to be found to some degree. In case of ERVs, a remainder of the *env* gene can be found in some elements (19).

Up until now, PBS and PPT search functionality is implemented in the *LTR_FINDER* (9) and *RetroTector* (20) software tools. *LTR_FINDER* also contains a simple search for a RT coding sequence. *RetroTector* models the internal region as a chain of matches to a set of motif sequences. Another software tool described in (8) (from now on referred to as *LTR_Rho*) utilizes profile hidden Markov models (pHMMs) of internal protein domains to eliminate false positives from the output set.

This article presents a new software tool, called *LTRdigest*, implementing a combination of various methods to detect and annotate internal features in LTR retrotransposon candidates, e.g. derived from *de novo* prediction programs. As another main contribution, we describe a work flow using the result of an *LTRdigest* run to perform a genome-wide *de novo* annotation and classification of LTR retrotransposon insertions. Such analyses produce a valuable genome-wide census of LTR retrotransposons, including information about individual insertions (e.g. number and positions of internal features). This information can improve discrimination between full-length, truncated or nested elements. Knowledge about internal features can also assist in classification of *de novo* predicted LTR retrotransposon candidates into families.

Recently, two programs were developed (21,22) that classify *de novo* predicted TEs into the major classes of DNA transposons, LTR retrotransposons and non-LTR retrotransposons, respectively. A software like *LTRdigest* with the potential of identifying individual families of *de novo* predicted LTR retrotransposons would assist the task of *in silico* classification on a different level—the assignment of families.

In combination with filtering out truncated or nested elements, family assignment will help in creating species-specific libraries of representative LTR retrotransposon sequences for each family in the organism. These libraries are then ready to be used to identify solo LTRs or heavily fragmented copies in a genome-wide masking effort.

We are not aware of any comparable previous tool implementing a generic approach specific for fine-grained LTR retrotransposon annotation.

MATERIALS AND METHODS

To reliably identify features inside an LTR retrotransposon candidate, computational models for PPT, PBS and protein domains, including their model parameters, are required. Feature identification in this context means obtaining the start and end positions of a particular sequence feature inside each LTR retrotransposon candidate. Depending on the feature, one usually derives additional detailed information such as scores or binding partners.

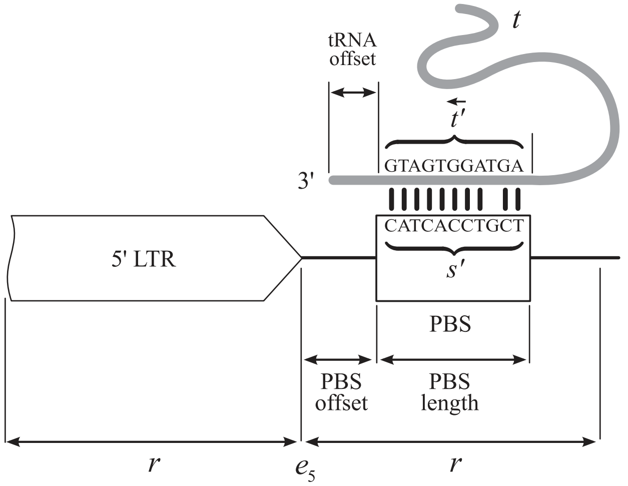


Figure 2. Overview of the *LTRdigest* primer binding site model. A putative primer binding site is represented by a high-scoring local alignment of an area of length $2r+1$ around the 5′-LTR boundary and a sequence from a tRNA library. Candidate alignments are scored according to alignment length, offset and the distance of the aligned substrings.

PBS detection

PBSs are identified by detecting regions complementary to host tRNA. This process requires a tRNA sequence library T. Such a library can be predicted from the genomic sequence with high accuracy, e.g. using tRNAscan-SE (23). To model the complementarity of a tRNA $t \in T$ and a putative PBS on the retrotransposon sequence, the reverse complement of t (to be supplied in 5′–3′ direction) is locally aligned to the genomic sequence.

The PBS is expected to start at close distance to the inner 5′-LTR boundary e_5 in the LTR retrotransposon candidate sequence u (Figure 1). Thus, the region to align to the tRNAs can be restricted to an area around this boundary. In our model, we use a user-specified radius r which defines a search interval $u[e_5{-}r \ldots e_5 + r]$ of length $2r + 1$ around e_5 (Figure 2). High-scoring local alignments under conservative alignment scores (user-configurable with default scores: 1 for a match, −2 for a mismatch, −4 for an indel) are then computed by the Smith–Waterman local alignment algorithm (24) and considered as potential PBS candidates.

The start and end positions of the aligned substring s' of the retrotransposon sequence mark the location of a putative PBS, whereas the respective coordinates for the aligned tRNA substring t' mark its hybridizing counterpart.

Furthermore, several distance-based constraints, such as maximum allowed offsets and minimal alignment length, are imposed on the alignment (Figure 2). All alignments satisfying these constraints are then scored based on their length, tRNA offset, PBS offset and tRNA length (see Supplementary Data 1 for details). The genomic substring involved in an alignment maximizing this score is finally reported as a putative PBS for the examined LTR retrotransposon candidate.

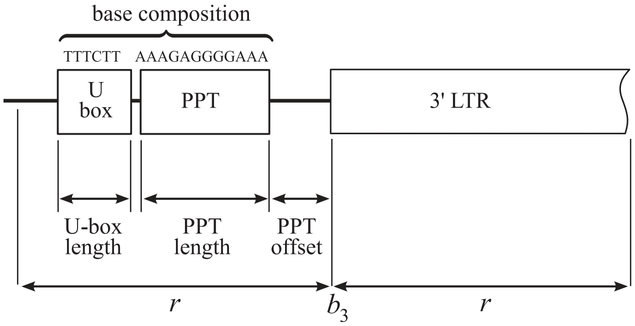


Figure 3. Overview of the *LTRdigest* PPT model. An area of length $2r+1$ around the 3′-LTR boundary is partitioned with respect to differences in base composition. Results are then subject to scoring and length constraints to produce a final PPT hit.

PPT detection

The PPT is described by the base composition of the PPT and U-box as well as by the length and exact position of the PPT and U-box within a given LTR retrotransposon candidate. These values are measured relative to the inner 3′-LTR boundaries (Figure 3).

To identify all sequence regions with a purine-rich base composition, we implemented a HMM (25). In this base composition HMM, characters of the DNA alphabet are emitted according to region-specific base distributions dependent on the set of states $Q = \{out, in_p, in_u, in_n\}$. We have chosen uniform transition probabilities of 0.05 and additional parameters p_R and p_T, describing the expected purine or thymine content in the PPT or U-box DNA sequences. Their default values are given in Supplementary Data 1. In the PPT state in_p, purines (A and G) are both emitted with probability $p_R/2$. That is, in the model there is a probability p_R of observing a purine at a specific position inside a PPT. T and C are both emitted with probability $(1 - p_R)/2$. Inside a U-box (state in_u), we define P_T to be the probability of observing a T. Consequently, the probability of observing one of the other nucleotides is $(1 - p_T)/3$. The state in_n is introduced to model stretches of N characters with a probability of 1, like they appear in a previously masked or poorly determined region. In addition, a background (e.g. uniform) probability distribution is used for the *out* state used to model sequence regions that are neither inside a PPT nor a U-box.

As the PPT must be located in the vicinity of the 3′-LTR, the sequence region of interest in this model is again restricted to a substring of the whole LTR retrotransposon sequence u. If the inner 3′-LTR boundary is denoted b_3 (Figure 1A), the radius r defines a sequence region $v = u[b_3{-}r \ldots b_3 + r]$ of length $2r+1$. This region v is then processed by the Viterbi algorithm (25), which delivers the sequence of states π maximizing $P(v,\pi)$, i.e. the combined probability of v and π. From π we can read potential PPT and U-box candidates as sequences of states consisting of in_p and in_u, respectively.

Afterwards, candidates are filtered according to user-defined minimum and maximum length constraints. Each of the remaining PPT candidate is scored based on the distance of its end position to the inner 3′-LTR boundary (see Supplementary Data 1 for details). A candidate with maximum score is then reported as the most likely PPT in the given LTR retrotransposon candidate sequence. A possible immediately preceding U-box candidate is reported as well.

Protein domain detection

pHMMs (25,26) are a widely used probabilistic representation of protein domain families and can conveniently be used to search for known domains in given protein sequences. pHMMs are versatile: First, models are publicly available, e.g. from the Pfam protein family database (27). This database contains various prebuilt models of protein domains associated with the process of retrotransposition. Secondly, pHMMs can easily be built from custom multiple sequence alignments. Due to this flexibility, pHMMs were chosen to model protein domains in LTR retrotransposon candidates. For the analyses performed in this work, collections of protein domain models associated with LTR retrotransposons were compiled (Supplementary Data 1, Tables B1 and B2). Given such a user-configurable set D of domain models in HMMER format, *LTRdigest* searches for all models in the translations of all six reading frames of a LTR retrotransposon candidate sequence. In the case of frame shifts, it is possible to obtain multiple partial hits per protein domain occurring in different reading frames. If more than one hit per domain model is found in a candidate, individual hits are combined using a chaining algorithm adapted from the gene prediction software *GenomeThreader* (28). This algorithm is able to find an optimal sequence of individual hits representing the model-sequence alignment best. Finally, the amino acid start and end positions in the translated sequences of all hits in the optimal chain below a user-defined E-value threshold are mapped back to the respective coordinates in the DNA sequence before they are reported.

Strand determination

For each individual PBS hit, PPT hit and protein domain hit, we obtain a strand assignment for a set of hits from the same candidate as follows: if all feature hits share a common strand property, then this strand is taken as the strand of the whole element. If individual hits have been discovered on contradictory strands, they are ordered by their evidence. We consider protein domain hits found in the internal region to be the strongest evidence identifying the orientation of a candidate. Thus, the protein domain hit with the smallest E-value is chosen to determine the direction of the whole candidate. All protein domain hits in other directions are disregarded for strand assignment. If no protein domain hits are present, the strand property of a PBS hit determines the strand property of the whole LTR retrotransposon candidate. If no PBS hit is present either, the strand of the PPT hit is used. Finally, if no hits exist, the strand property of the whole candidate remains unchanged.

De novo classification approach

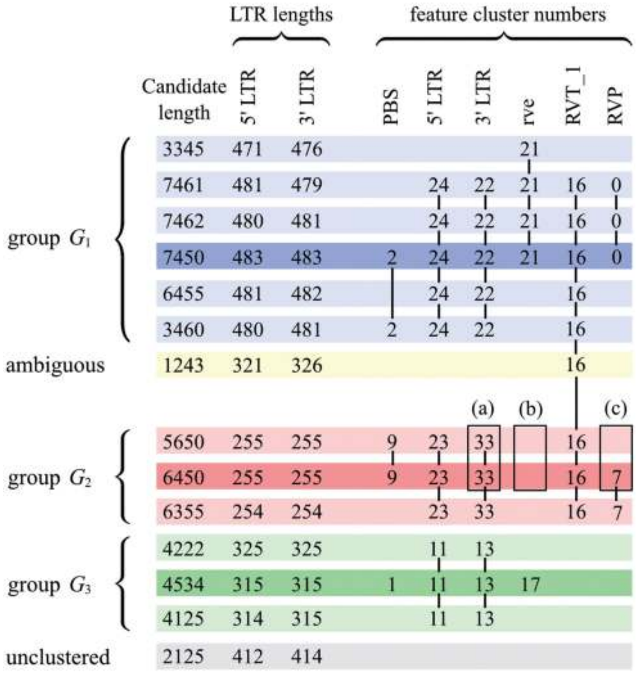
Members of a family of LTR retrotransposons are expected to share sequence identity as they are derived from a common ancestor. In the context of *de novo* LTR retrotransposon analysis, the task of family classification addresses the problem of identifying how predicted LTR retrotransposon insertions can be assigned to a number of specific families. Another task is the identification of those sequences which represent the whole family in an optimal way (full-length and/or near full-length sequences). This task is different from recognition of known families by comparison to a database of reference sequences. The approach presented here relies on the comparison of individual internal features in predicted LTR retrotransposon candidates and is, therefore, independent of reference sequences from other organisms. Candidates with similar feature sequences are treated as potential members of the same family by combining evidence across all detected features. This approach takes into account the homology of internal protein-coding regions—independent of the surrounding sequence context—as well as the family-specific conservation of the LTR sequences and regulatory signals like PBS or PPT.

Our approach to *de novo* family classification consists of several steps which rely on the feature hits assigned to substrings of an LTR retrotransposon candidate. Let C be the set of LTR retrotransposon candidate sequences. Let D be the set of protein domain models. Then $F = \{ltr_5, ltr_3, pbs, ppt\} \cup D$ is the set of possible features assigned by *LTRdigest*. This assignment is represented by a function φ such that $\varphi(c,f) = (i,j)$, if candidate c has feature f in its substring $c[i \ldots j]$. If c does not have feature f, we write $\varphi(c,f) = \perp$, where $\perp$ stands for undefined. The first step of the classification consists of clustering each set of substrings having the same feature. That is, for each $f \in F$ we perform a separate single linkage clustering for all sequences $c[i \ldots j]$ such that $c \in C$ and $\varphi(c,f) = (i,j)$ for some i,j. We use the dbcluster tool from the *Vmatch* software (http://www.vmatch.de) to perform the single linkage clustering. The clustering parameters specifying under what conditions two sequences go into the same cluster are calculated automatically from the individual sequence sets. The respective rules can be found in Supplementary Data 1.

For each $c,c' \in C$ and $f \in F$, we write $(c,f) \approx (c',f)$ if $\varphi(c,f) = (i,j)$, $\varphi(c',f) = (i',j')$ for some i,j,i',j' and $c[i \ldots j]$ and $c'[i' \ldots j']$ are in the same cluster with respect to the single linkage clustering according to feature f. Two candidates $c, c' \in C$ are compatible if the following holds:

(i) for all $f \in F$, we have $\varphi(c,f) = \perp$ or $\varphi(c',f) = \perp$ or $(c,f) \approx (c',f)$, and
(ii) there is at least one $f \in F$ such that $(c,f) \approx (c',f)$.

That is, candidate sequences are pairwise compatible if they share cluster memberships for at least one feature (Figure 4). The compatibility relation on C splits this set into subsets such that all pairs in each subset are pairwise compatible. The subsets need not necessarily be disjoint. We obtain unique memberships by discarding all



	Candidate length	5′ LTR	3′ LTR	PBS	5′ LTR	3′ LTR	rve	RVT_1	RVP
group G_1	3345	471	476				21		
	7461	481	479		24	22	21	16	0
	7462	480	481		24	22	21	16	0
	7450	483	483	2	24	22	21	16	0
	6455	481	482		24	22		16	
	3460	480	481	2	24	22		16	
ambiguous	1243	321	326					16	
group G_2	5650	255	255	9	23	33		16	
	6450	255	255	9	23	33		16	7
	6355	254	254		23	33		16	7
group G_3	4222	325	325		11	13			
	4534	315	315	1	11	13	17		
	4125	314	315		11	13			
unclustered	2125	412	414						

Figure 4. Illustration of the candidate joining approach (using example data). Each line depicts a single candidate. Maximal groups of compatible candidates are indicated by colours (G_1: blue, G_2: red, G_3: green). Lines between cluster numbers indicate shared clusters among compatible elements. Candidates marked as ambiguous (yellow) may possibly be assigned to more than one group and are excluded from further analyses. Candidates whose features do not belong to any cluster are shown in grey. Most complete elements are drawn using stronger colours. For two candidates from group G_2, examples for the compatibility requirements used in the joining criterion are shown. Case (a): $(c,ltr_3)\approx(c',ltr_3)$, case (b): $\varphi(c,rve)=\varphi(c',rve)=\perp$, case (c): $\varphi(c,RVP)=\perp\neq\varphi(c',RVP)$.

candidates that could not unambiguously be assigned to a unique group. This cleaning step gives us disjoint groups $G_1, G_2, \ldots, G_h$, such that for all i, $1 \leq i \leq h$ all elements in G_i are pairwise compatible.

Afterwards, we determine which members of each group could likely be complete representatives of their group. As potentially good representatives, we prefer to select candidates with the most frequently appearing feature characteristics (see Supplementary Data 1 for details).

Implementation

The models and algorithms described above were implemented in ANSI C based on the *GenomeTools* genome analysis package (http://genometools.org), a free, open source collection of bioinformatics software. The *GenomeTools* package also contains the *LTRharvest* software (12) used to predict LTR retrotransposon candidates.

As input, *LTRdigest* requires the sequence coordinates of the LTR retrotransposon candidates to be provided in GFF3 format with feature types conforming to the Sequence Ontology [SO, (29)]. Each candidate is represented by a line of type *LTR_retrotransposon*. Additionally, two lines of type *long_terminal_repeat* are required for each candidate, representing the 5′- and 3′-LTR boundaries (Figure 5). The sequences that the coordinates refer to must be provided as an encoded sequence as delivered by the *GenomeTools suffixerator* tool. All these preconditions are satisfied by the output delivered by *LTRharvest* (12). Other tools producing GFF3-formatted output can be used as well.

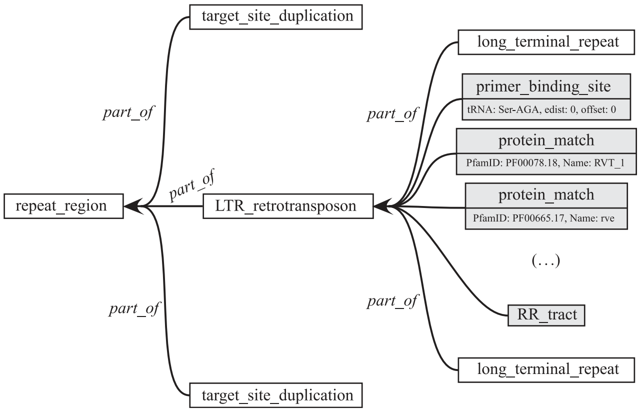


Figure 5. GFF3 annotation graph structure for a single LTR retrotransposon candidate produced by *LTRdigest*. Additional nodes (drawn in grey) for a PPT (*RR_tract*), a PBS (*primer_binding_site*) and several protein domain matches (*protein_match*) are present. The feature types conform to the SO (29) (in the *mobile_genetic_element* subtree).

LTRdigest extends the annotation graph implicitly given in the GFF3 file with new features. This is done by applying the internal feature detection algorithms described above to the sequence of each LTR retrotransposon candidate. New nodes are added to the annotation graph as features are detected. PPT features are represented by nodes of the *RR_tract* SO type, PBS features by the *primer_binding_site* type and protein domain matches by the *protein_match* type (Figure 5).

In addition to the extended annotation in GFF3 format (see Supplementary Data 3 and 5 for examples), *LTRdigest* generates output files with a common, user-defined name prefix. This naming scheme allows the user to keep the output from several runs—for example, with different parameter sets—in the same directory separate from each other. First, sequences are written into separate multiple-FASTA files for,

(i) whole LTR retrotransposon candidate sequences (oriented in the most likely reading direction);
(ii) 5′- and 3′-LTR sequences;
(iii) PPT and PBS sequences; and
(iv) concatenated coding DNA and amino acid sequences for each protein domain recognized.

Each of the sequence files include one FASTA entry per LTR retrotransposon candidate referenced by its source sequence and start/end positions.

Secondly, the tool creates a text file describing the particular *LTRdigest* run parameters and a tabulator-separated text file in which each line contains detailed information about all known and predicted features in each LTR retrotransposon candidate, including all features' start and end positions and lengths. For PBS and PPT features, motifs, offsets and scores are output as well. It also contains a list of the pHMM IDs for the protein domain hits, in order of appearance in the candidate's most likely reading direction. The tab-separated text file is created in addition to the GFF3 output file because it contains all relevant information in one file and can be imported and conveniently examined by a user in a standard spreadsheet application (see Supplementary Data 4 for an example).

The pHMM search is done by the HMMER software (http://hmmer.janelia.org). This widely used software package contains code to create, calibrate and search pHMMs in DNA and amino acid sequences. As in the *hmmsearch* tool from the HMMER suite, a search for more than one domain model can be run simultaneously using multiple concurrent operating system threads to take advantage of parallelization in the increasingly popular multi-core systems.

All detection algorithms and result structures, including those calling HMMER code, are integrated into *GenomeTools* by providing an object-oriented application programming interface (API) which can readily be used by any software utilizing the *GenomeTools* library.

The *LTRdigest* work flow is implemented as a command-line tool callable via the main *GenomeTools* executable `gt`. The *LTRdigest* tool can be compiled for any POSIX-conforming UNIX-like operating system and has been successfully tested on a variety of 32- and 64-bit platforms. A *GenomeTools* source distribution containing LTRdigest is available for download at http://www.zbh.uni-hamburg.de/LTRdigest at the time of publication.

The classification approach described above is implemented as a collection of Ruby scripts processing the candidate sequence files and tabular output data as created by *LTRdigest*. Additional external programs such as *Vmatch* are called from the Ruby scripts.

RESULTS

In this section, we present example use cases for fine-grained LTR retrotransposon candidate annotation using *LTRdigest*. In particular, candidates predicted from *Drosophila melanogaster* and *Mus musculus* genomic sequences were examined. For the *D. melanogaster* application, additionally *de novo* classification of candidate sequences into putative families is performed. Finally, the results of this automated classification step are evaluated by sequence-based comparison of representative sequences to a reference data set.

LTRdigest annotation results for the *D. melanogaster* genome

To exemplify the use of *LTRdigest* for annotation purposes, we used the software to process LTR retrotransposon candidates computationally derived from the *D. melanogaster* release 5.8 genome. These were predicted by *LTRharvest* using the parameters given in Supplementary Data 1. While largely identical to the parameters used in the work of Ellinghaus *et al.* (12), the 'overlap' parameter was set to 'no' to disable reporting of nested or overlapping candidates. A copy of the genomic sequence in FASTA format was obtained from the FlyBase (30) database, excluding mitochondrial genome sequences and unplaced contig sequences (given in the 'U' and 'Uextra' sequence files), but including the heterochromatic sequences (labelled with 'Het'). *LTRharvest* delivered 713 LTR retrotransposon candidates. Of these, 521 are located in the euchromatic portion of the genome and 192 in the heterochromatic region (Table 1). It should be noted that both full-length and truncated LTR retrotransposon insertions are reported as candidates.

Table 1. Statistics for the *LTRharvest*/*LTRdigest* runs on the *D. melanogaster* release 5.8 genome

Chromosome	2L	2R	3L	3R	4	X	Σ
Size (Mbp)	22.3	20	23	27	1.2	22	116
Candidates	108	161	182	138	11	113	713
In euchromatin	96	99	114	90	11	111	521
In heterochromatin	12	62	68	48	0	2	192
With PBS	51	61	72	56	4	48	292
In euchromatin	48	44	50	42	4	47	235
In heterochromatin	3	17	22	14	0	1	57
With PPT	56	75	84	65	4	56	340
In euchromatin	51	46	56	45	4	55	257
In heterochromatin	5	29	28	20	0	1	83
With PBS + PPT	33	38	48	32	2	38	191
In euchromatin	31	26	36	25	2	37	157
In heterochromatin	2	12	12	7	0	1	34
With domains	99	145	166	126	6	94	636
In euchromatin	87	87	101	81	6	92	454
In heterochromatin	12	58	65	45	0	2	182

The first section shows chromosome sizes. The second section shows the number of candidates per chromosome arm containing specific features. The last column aggregates the values over all chromosome arms. The term 'in heterochromatin' refers to candidates found in sequence files labelled 'Het', 'in euchromatin' refers to sequence files not labelled 'Het'.

The LTR retrotransposon candidates were annotated using *LTRdigest*, which took ~12 min on an eight-core Intel Xeon E5410 system (2.33 GHz, SuSE Linux). We used default parameters except for the allowed tRNA offset range used in the PBS detection (see Supplementary Data 1, Table A2). This value was modified to address binding sites farther into the tRNA molecule as the PBS of some *Drosophila* LTR retrotransposons are known not to bind exactly at the 3′-end of the respective tRNA (14). To obtain a tRNA library, we processed the *D. melanogaster* subset from the Genomic tRNA Database (31) (304 sequences) into a non-redundant set of 100 individual tRNA sequences. A set of 22 protein domains specific or related to LTR retrotransposons and retroviruses was selected from the Pfam database (see Supplementary Data 1, Table B1).

The pHMM hits for protein domains were found in 636 candidates (454 in euchromatin, 182 in heterochromatin), which account for 89% of all candidates. The majority of the protein domain hits found in the candidates show the RT–INT order of the RT and integrase domains typical for retroelements of the *gypsy* superfamily

(346 occurrences, see Supplementary Data 1, Table E1). In 41 candidates, an INT–RT domain hit order was observed, indicating the possible presence of retrotransposons of the *copia* superfamily, which show this characteristic order (14). In some of the *gypsy*-like patterns, RT (145 occurrences) or integrase (51 occurrences) are missing, whereas the other domains are present in the correct order. As reviewed in (32), it is possible to distinguish LTR retrotransposon candidates into superfamilies solely on the basis of internal protein domain order. Thus, a first superfamily classification on this basis is possible without closer examination of the feature sequences themselves.

For 292 of 713 (41%) predicted candidates, a potential PBS longer than 11 bp was found. The five most frequently occurring tRNA signatures were consistent with known tRNAs binding to *D. melanogaster* retrotransposons (14,33). Additional PBS locations complementary to 10 additional tRNAs were found (see Supplementary Data 1, Table F1). PBS lengths varied between 11 bp and 28 bp with a median of 15, whereas the median distance of the PBS from the 5′-LTR end was 2 bp, varying between 0 bp and 5 bp. Thus, PBS hits were mostly well located at their expected positions, allowing for slight LTR position inaccuracies. The median length agreed with the length of known PBSs.

PPTs were present in 340 out of 713 candidate sequences (48%). The shortest PPT was 8-bp long, the longest was 30 bp. The median PPT length was 13 bp. While the distance of the predicted PPTs from the 3′-LTR start varied between 0 bp and 28 bp, the median of 1 bp indicated that they were mostly located at their expected positions as well.

LTRdigest annotation results for *M. musculus* chromosome 4

Chromosome 4 of the *M. musculus* genome was chosen as an example to demonstrate the use of *LTRdigest* for efficient identification and annotation of full-length or near full-length LTR retrotransposons in a mammalian genome. This is done by annotation of the predicted candidates using *LTRdigest* and subsequent filtering of these candidates, retaining only those with a complete protein domain set.

The chromosome 4 reference assembly file (build 37, version 1, 158 MB) was downloaded from the NCBI FTP server and candidates were predicted using *LTRharvest*. Relaxed, unrestrictive default parameters were used to ensure high sensitivity (Supplementary Data 1, Table A1). Again, the ‘overlap’ option was set to ‘no’ to disable reporting of nested or overlapping candidates. *LTRharvest* processed the file in 7 min (including enhanced suffix array construction) on an Intel Xeon E5410 system (2.33 GHz, SuSE Linux).

In the next step, *LTRdigest* was used to annotate internal features in these candidates. The *M. musculus* tRNA set (433 sequences) from the Genomic tRNA Database (31) was used. After removing multiple redundant copies, 248 tRNA sequences remained. The set of pHMMs to search for was extended with models of known *gag*, *pol* and *env* domains from mammalian ERVs and retroviruses, taken from Pfam. As a result, we obtained a new set of 21 pHMMs (see Supplementary Data 1, Table B2). The remaining parameters were left at their default values (see Supplementary Data 1, Table A2). The *LTRdigest* run on all 1711 *LTRharvest* candidates took 43 min on the computer system mentioned above.

The pHMM hits to the set of 21 protein domain models were found in 471 candidates. While 164 candidates contained only a RT domain, there was a significant number of candidates with a complete or near-complete set of protein domains needed for retrotransposition. A total of 88 candidates contained at least one pHMM hit for each of the following features: a *gag* domain model, a protease domain, a RT domain and an integrase domain. Figure 6 shows an example for a particularly detailed

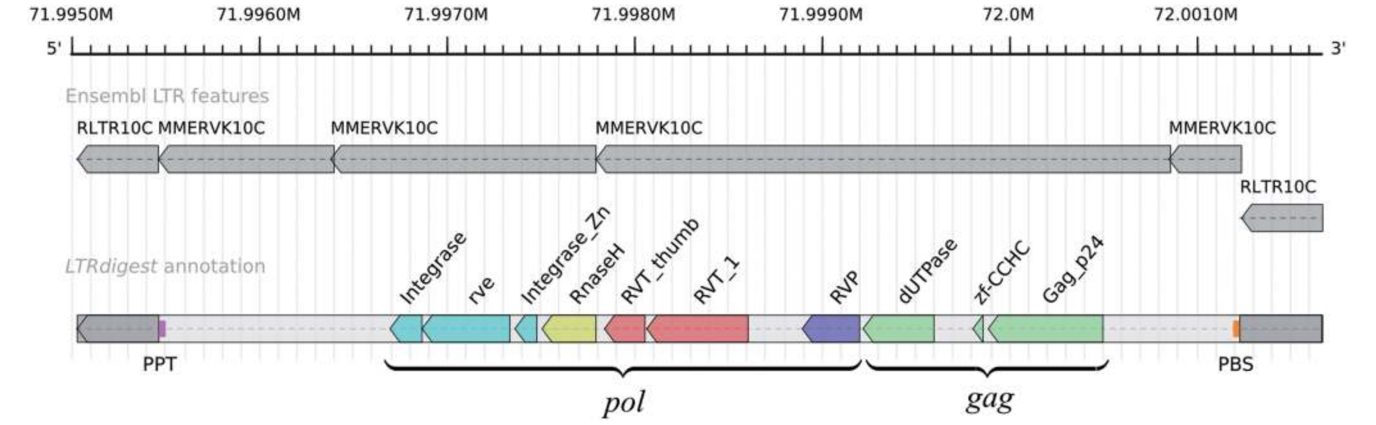


Figure 6. Example annotation of an LTR retrotransposon candidate from *M. musculus* chromosome 4, positions, 71 995 000–72 001 670. The image was created using the *AnnotationSketch* software (42). The top track shows sequence-based LTR retrotransposon matches from Ensembl (34) release 54 (May 2009). LTR matches and internal region matches can only be linked by their Repbase feature identifier (‘RLTR10C’ for the LTR matches and ‘MMERVK10C’ for internal region matches). The bottom track contains a representation of a hierarchical LTR retrotransposon annotation graph, as reported by *LTRdigest*, collapsed into a single track. The grey blocks at the end of the element represent the LTRs, whereas the coloured elements in the middle represent protein domain hits. The orange and purple blocks at the inner LTR boundaries represent PBS and PPT features, respectively. The image shows that *LTRdigest* creates detailed, integrated annotation data for each full-length insertion, while a sequence-based approach can result in fragmented matches which are not easily recognized as parts of a single full-length LTR retrotransposon.

annotation. These 88 candidates are very likely to represent full-length representatives of their respective families. An *env* domain hit was not required for making a candidate (near) full-length because—in contrast with infectious retroviruses—*env* genes are not necessarily present in LTR retrotransposons and ERVs (19). Nevertheless, 20 candidates contained a hit to the 'TLV_coat' Pfam model representing part of an *env* domain.

To assess whether these 88 putative full-length representatives are supported by a reference dataset, we compared the positions of these insertions with the LTR retrotransposon annotation from the Ensembl release 54 database (34), containing RepeatMasker-based matches to Repbase LTR retrotransposon sequences. The comparison was done by counting the number of Ensembl LTR features overlapping with each putative full-length LTR retrotransposon. As LTR and internal sequences for a specific LTR retrotransposon are stored as separate entities in Repbase, at least three hits per recognized full-length representative are expected: one for each LTR match and at least one for the internal region. Indeed, 87 of 88 representatives overlap with three or more Ensembl hits, spanning the whole predicted candidate. The majority (53 of 87) overlap with exactly three Ensembl hits. However, for 34 candidates, the corresponding hits are apparently not stored in Ensembl as continuous features, but instead fragmented into several individual feature entries (see example in Figure 6). The internal region of the remaining one of the 88 representatives overlaps with two long Ensembl matches with no corresponding LTR matches.

In turn, we checked whether features from Ensembl likely to be full-length LTR retrotransposons in the mouse genome overlap with the 88 putative full-length insertions. Autonomous (and thus full-length) retroviral-like elements in the mouse genome are commonly considered to be of length between 6000 bp and 9000 bp (35). In the Ensembl release 54 database, there are 133 sequence-based LTR retrotransposon entries of length at least 5000 bp on chromosome 4 of *M. musculus*. The minimum length was decreased by 1000 bp to compensate for the lengths of LTRs stored separately from the internal regions in Ensembl.

Of these 133 Ensembl features, 67 overlap with one of the 88 putative full-length representatives, supporting their full-length status and leaving 66 features with no corresponding *LTRdigest* candidate. Thirty-five of these represent incomplete LTR retrotransposons which contain protein domain hits but do not show the required full set of features. That is, either a *gag*, protease, RT or integrase function is missing. Twenty-three others were dropped due to the strict *LTRharvest* 'overlap' option which was set to disregard candidates overlapping with each other (e.g. nested insertions).

As for other internal features in the 88 putative full-length candidates, PBSs complementary to seven host tRNAs were present in 39 of the 88 candidates (see Supplementary Data 1, Table F2). PBS lengths varied across a range of 11–18 bp with a median of 15 bp. The PBS offset from the 5′-LTR end ranged from 0 bp to 5 bp with a median of 4 bp. Of 88 putative full-length candidates (94%), 83 contained PPT regions satisfying the default model. They are of length between 10 bp and 30 bp (median 17 bp). Offsets from the 3′-LTR start varied across a range of 0–28 bp with a median of 2 bp.

Family classification results for the *D. melanogaster* genome

The classification process (see Materials and Methods section) does not use reference data, but relies strictly on the *de novo* annotation results generated by *LTRdigest*. Candidates are grouped according to the clustering results of internal features. Out of 713 LTR retrotransposon candidates, 535 were assigned to 79 candidate groups. Of the 178 candidates which were not assigned to any group, 77 were discarded because they did not contain any protein domain hits. Twenty-three remained singlets for all features during the sequence clustering step and therefore were excluded from further analyses. Another 78 candidates could not be unambiguously assigned to a group and were excluded as well. From the 79 groups of 535 candidates altogether, 17 groups containing only one member were discarded, as were their member candidates. This step resulted in 62 final candidate groups, containing 518 candidates altogether. From these 518 candidates, 376 were selected as good representative sequences (see Materials and Methods section), ready to be used for comparison to a reference data set.

To evaluate if the candidate groups detected in the last step represent known LTR retrotransposon families in *D. melanogaster* and to assess how well the current reference sequences could be reproduced, the sequences of all representative candidates in the groups were compared with a reference set of known *Drosophila* transposon sequences using BLAST (36). The reference set was obtained from the FlyBase database and contains one representative full-length sequence per known TE family. From this reference set, only the 49 LTR retrotransposons marked as complete were considered (see Supplementary Data 1). Additionally, the global sequence identity was calculated for all members of a group in comparison to the single reference sequence resulting in the best BLAST match. This was done using the *needle* tool from the EMBOSS suite (37).

In this sequence-based comparison, representative sequences from 41 out of these 62 candidate groups matched to the reference set with high global sequence identity (82.2% to 100%, median 99.5%, see Table 2). We considered a group matched to a reference sequence if a global alignment of at least one sequence in the group exceeded a sequence identity threshold of 80%. Though no such match was found in the reference dataset for the sequences of groups Dmel-17 and Dmel-37, a subsequent search in the *Repbase* database (7) using the CENSOR tool (38) revealed high-quality matches of these sequences to the only recently described *Bica* (39) and *Chimpo* element sequences which were not present in the FlyBase reference set in the first place. Group Dmel-36 was considered to represent the *accord2* family although an initial global alignment with its reference sequence did not reach

Table 2. Candidate groups of LTR retrotransposons in the *D. melanogaster* genome with high similarity (at least 80%) to a reference sequence of a known family

Group	No. of members	Candidate groups			Reference	
		PBS	PPT	Domains	Identity (%)	Family
Dmel-0	24	Arg-TCG	aaaaaagaggggagg	PR–RT–INT	99.9	blood
Dmel-1	64	–	–	PR–INT	99.9	roo
Dmel-2	2	Leu-CAA	–	RT–INT	99.4	mdg3
Dmel-3	17	Lys-TTT	gaggggggaggag	RT–INT	100.0	opus
Dmel-4	21	Met-CAT	–	INT–RT	99.9	copia
Dmel-5	19	Ser-AGA	aagggaaggggag	PR–RT–INT	99.2	297
Dmel-6	13	Arg-TCG	aaaaggagggaga	PR–RT–INT	98.7	mdg1
Dmel-7	2	–	gaggagggaa	PR–RT–INT	99.2	springer
Dmel-8	7	Ile-AAT	–	PR–INT	99.6	diver
Dmel-9	9	Lys-TTT	aagaggggaggag	RT–INT	100.0	HMS-Beagle
Dmel-10	10	Ser-AGA	ggggggggaggag	PR–RT–INT	99.9	Tirant
Dmel-11	4	Arg-TCG	aaaaggagggaga	PR–RT–INT	100.0	Tabor
Dmel-12	7	Ser-AGA	aaaggatggggaag	PR–RT–INT	100.0	Quasimodo
Dmel-13	9	–	–	PR–RT–INT	99.8	Transpac
Dmel-14	13	–	–	PR–RT–INT	99.6	flea
Dmel-15	2	Leu-CAA	–	INT	98.3	invader4
Dmel-16	8	–	agaaggggaggag	PR–RT–INT	99.7	Burdock
Dmel-17	5	Leu-CAA	aagaggaagagcatgagagaggggg	PR–RT–INT	99.0	Bica
Dmel-19	3	–	ggggaggag	PR–RT–INT	97.0	gypsy4
Dmel-20	3	Leu-CAA	–	RT–INT	97.3	invader2
Dmel-21	3	Tyr-GTA	aaggggggggagaa	RT–PR–INT	99.5	Max-element
Dmel-23	3	Tyr-GTA	–	RT–PR–INT	82.2	GATE
Dmel-24	3	–	–	RT–INT	99.6	invader3
Dmel-26	3	Leu-CAA	–	INT	99.1	invader1
Dmel-27	6	–	–	PR–INT	100.0	3S18
Dmel-30	3	Arg-TCG	aaaaagggagg	PR–RT–INT	99.0	Stalker4
Dmel-32	4	Ser-AGA	aaagggagggaag	PR–RT–INT	100.0	rover
Dmel-33	3	Trp-CCA	gggggagga	INT	95.9	diver2
Dmel-34	3	–	gaggggggag	PR–RT–INT	99.9	gypsy
Dmel-36	2	–	–	RT–INT	100.0	accord2
Dmel-37	2	Arg-TCT	ggagggggag	RT–INT	99.0	Chimpo
Dmel-38	7	Ser-AGA	aagggaagggaag	PR–RT–INT	98.9	17.6
Dmel-40	17	Arg-TCG	aaaaggagggaga	PR–RT–INT	100.0	412
Dmel-43	6	Cys-GCA	aaaaagggagg	PR–RT–INT	99.3	Stalker2
Dmel-44	4	Ser-AGA	gagaatggaaaaaa	PR–RT–INT	98.7	Idefix
Dmel-45	2	–	–	PR–RT–INT	96.1	gypsy6
Dmel-49	3	Lys-TTT	aagaggggaggag	RT–INT	99.9	HMS-Beagle2
Dmel-52	2	–	ggggggggaggag	RT–INT	85.8	ZAM
Dmel-54	3	Arg-TCG	aaaaagggagg	PR–RT–INT	99.6	Stalker4
Dmel-59	2	–	–	PR–RT–INT	91.6	blood
Dmel-60	2	–	ggggaggag	PR–RT–INT	98.7	gypsy4

The left side of the table gives the name of each group, along with their number of representatives and their features. The right side shows the *D. melanogaster* family determined by matching representative sequences against a set of reference sequences. The last column gives the similarity value of the best global alignment between the reference sequence and the representative sequences.

the 80% threshold because the reference sequence is apparently given on the opposite strand. Comparing the sequences on group Dmel-36 with the reverse complement of the *accord2* sequence led to correct results. The orientation was confirmed by submitting the reference sequence to CENSOR, which also indicated a perfect complementary match.

In three cases, pairs of these 41 candidate groups were matched to the same known LTR retrotransposon family because they were separated due to slight differences in LTR (Dmel-0, Dmel-59), PBS (Dmel-30, Dmel-54) or *gypsy* domain clustering (Dmel-19, Dmel-60).

The remaining sequences from 21 of 62 groups (not listed in Table 2) did not show any satisfactory hit to the reference sets (see Supplementary Data 1, Table C for more information). Most of these 21 groups are small groups, likely created by repetitive sequences from regions with nested (non-LTR) transposons. These gave rise to pHMM protein domain matches (mostly very few per candidate) and therefore led to grouping of the respective candidates. In some groups, the sequences in the groups were not complete but subject to internal deletions or insertions (e.g. Dmel-29 or Dmel-57, which represent incomplete members of the *Circe* and *micropia* families).

Of the 49 known LTR retrotransposon families marked as 'complete' in the *D. melanogaster* reference sequence file, 12 were not matching any sequence for any candidate family and thus remain unaccounted for (see Supplementary Data 1, Table D). A BLAST search for their reference sequences in the initial candidates produced by *LTRharvest* revealed that the only full-length hits for these reference sequences were found in candidates which were discarded because they either

(i) could not unambiguously be assigned to a group, or
(ii) were singlets for all features during the feature clustering process, or
(iii) were the only members of their respective groups, or
(iv) were not recognized in any of the initial candidates given as input to *LTRdigest*.

DISCUSSION

This article presents *LTRdigest*, a software tool for flexible identification of internal features inside LTR retrotransposon or ERV sequences given in a standardized format. *LTRdigest* utilizes a variety of sequence analysis methods, some of which are already successfully used in the context of LTR retrotransposon detection.

For example, *LTR_FINDER* (9) uses local alignments between tRNA sequences and LTR retrotransposon sequences to detect possible PBSs. The implementation of the PBS detection in *LTRdigest*, however, improves on the *LTR_FINDER* implementation in terms of flexibility. For example, it allows fine tuning of Smith–Waterman alignment scores, binding specificity and locality of the PBS prediction. In *LTR_FINDER*, only a minimal alignment length and a tRNA library file can be specified. Unfortunately, a direct quality comparison of feature prediction results from *LTRdigest* and *LTR_FINDER* is difficult because either the parameters of the prediction methods implemented in *LTR_FINDER* are too different in nature to produce equivalent conditions, or configuration of the necessary parameters is not possible. This cannot be compensated for by adjusting *LTRdigest* parameters because, in the case of *LTR_FINDER*, the parameters used (like Smith–Waterman alignment scores, distance constraints, etc.) are not documented and the source code is not openly available.

The *RetroTector* software (20), primarily intended for the identification of ERVs in genomic sequences, includes search capabilities for PPT, PBS and domain motifs. It also attempts to reconstruct ORFs for the *gag* and *pol* genes, something not implemented in *LTRdigest* yet. The TE model used in *RetroTector*, however, is relatively fixed and relies on a database of known motif sequences, thus limiting its use in *de novo* annotation of genomes for which such a database has not been built yet.

Although *LTRdigest* and *LTR_Rho* (8) both use HMMER as a protein domain scanning engine, a comparison is difficult: First, *LTRdigest* post-processes the pHMM hits by a match chaining step to detect domains even if they span mutations breaking open reading frames. Secondly, *LTR_Rho* does not output a detected protein domain, but uses it solely to filter out candidates during the detection process. That is, there is no separate domain annotation component in *LTR_Rho* to which we could compare *LTRdigest*. Finally, detection of PPT or PBS features is not implemented in *LTR_Rho* and therefore the prediction performance cannot be compared for these features. Protein domain detection in *LTRdigest* is implemented in a way similar to *LTR_Rho* (8), using pHMMs in a standardized format, like those available in the Pfam database. Unlike *LTR_Rho*, the domain detection in *LTRdigest* is not solely used for filtering purposes but for annotation as well, employing match chaining to obtain the most complete protein domain match possible. This emphasizes the purpose of *LTRdigest* as an annotation tool primarily generating further information about LTR retrotransposon sequences without being tied to a specific problem, making it usable in a wide variety of application areas.

The purpose of *LTRdigest* as an annotation tool is also apparent from the fact that the feature detection process is separated from the candidate prediction itself. For example, in *LTR_FINDER* both are intertwined in a way that prevents separate use. The separation makes *LTRdigest* usable on any given dataset of LTR retrotransposon candidates as long as they are sufficiently described by their LTR positions in GFF3 format, a standard format for genome annotations. Furthermore, unlike for *LTR_FINDER*, feature detection in *LTRdigest* is not tied to fixed filters, allowing a flexible combination of *LTRdigest* with other prediction tools and the integration into annotation and filtering pipelines. To simplify this, an open, object-oriented API facilitates integration of feature detection capabilities into custom C software built upon the *GenomeTools* toolkit. *LTRdigest* also outputs its results in a variety of formats. The GFF3 output can be used for computational post-processing or visualization (e.g. in a genome browser), while the tab-separated output file is readable in any common spreadsheet software. Detailed sequence and alignment output allows for sequence-based analyses of the results.

Another advantage of *LTRdigest* over existing tools is that the time-consuming feature identification process can run in parallel, if desired. As candidate sets are essentially GFF3 text files, it is possible to partition them (e.g. according to appropriate boundaries) and distribute the corresponding computational tasks on a compute cluster. In addition, a concurrent search for multiple protein domain models can be run simultaneously by multiple CPU cores on each individual cluster node. Therefore, the use of *LTRdigest* for feature identification is expected to scale well for growing LTR retrotransposon candidate and protein domain model sets.

LTRdigest is useful in a variety of applications. In a simple use case not presented here, the presence of protein domains was used to filter an LTR retrotransposon candidate set from the work of Ellinghaus *et al.* (12) for the *D. melanogaster* release 3 genome sequence by discarding every candidate not showing any protein domain, similar to an existing method used by Rho *et al.* (8). Using this simple filtering approach, the specificity of *LTRharvest* predictions could be increased from 59% to 75% with no loss of sensitivity. That is, out of 506 predicted candidates, the number of candidates not matching to a reference database of known full-length insertions (40) was decreased from 207 to 101, without removing any of the true positives (see Supplementary Data 1, Table G). Several other filtering conditions based on the presence of specific feature combinations were evaluated, but did not result in significant specificity improvement.

In another use case, this article shows that *LTRdigest* is suitable to identify (near) full-length ERV insertions in the chromosome 4 sequence of *M. musculus*. The internal features identified by *LTRdigest*, like PBS, PPT and protein domains, are not visible in the corresponding Ensembl features originating from sequence based hits to the Repbase database. The pHMM search step identified a fine-grained set of protein domain hits in the *M. musculus* candidates. Not only hits to the core domains are present, but also hits to *gag* sequences and small connecting domains between several functional units, e.g. between subdomains of the RT and integrase functions. A reason for this may be that the application of *LTRdigest* to mammalian ERVs benefits from the availability of many retroviral RT or integrase sequences (e.g. from HIV) for pHMM building, resulting in a high sensitivity of the protein domain pHMMs available in the Pfam database.

In the use case of *D. melanogaster* release 5.8 LTR retrotransposon predictions, the *LTRharvest* parameters were adjusted to report full-length and truncated insertions, as long as they contain both LTRs. It is useful to include truncated insertions in the pipeline as these insertions can support identification of families with few full-length copies present in the genome. In release 5.8 of the *D. melanogaster* genome sequence, 713 candidates were reported, while 1321 LTR elements were detected in release 4 (41). Due to the ‘overlap = no’ parameter setting, most of the 327 nested insertions of release 4 were not identified by *LTRharvest*. In addition, an unknown percentage of the remaining 994 elements are either solo LTRs or copies showing deletions in one or both LTRs, effectively preventing them from being detected by *LTRharvest*. Thus, a direct comparison of both data sets does not make sense.

In addition, we described an automated classification process, which shows good results for the *D. melanogaster* test case. A large number of known LTR retrotransposon families could be reproduced with several representative full-length copies. The representative sequences show high similarity to the complete reference sequences from FlyBase, indicating that reconstruction of element families from internal feature sequences is feasible at least in genomes with evolutionary rather young LTR retrotransposons. False positives, that is, groups of candidates classified as a potential family of LTR retrotransposons not matching any known reference, mostly contain very few feature hits, often only for single protein domains like RT or zinc knuckle domains which are also found in other kinds of retrotransposons. Matches of that kind occasionally result in misclassification of such candidates as LTR retrotransposons and subsequent grouping on the basis of the similarity of such features alone. Also, candidates from highly nested regions (e.g. in heterochromatin) with few good internal pHMM hits can lead to grouping of unrelated candidates. However, such groups can easily be identified by examination of a multiple alignment of all sequences in the group.

While this automated classification is neither able nor intended to completely replace careful examination by a human expert, it supports analysis by biologically meaningful preparation of potential candidate families for later examination. The classification approach as described in this article is not intended to assign a family to each and every candidate regardless of its completeness. Instead, it creates groups of candidates linked by combined feature sequence similarity, possibly in various stages of evolutionary rearrangement. This preliminary classification information can then be used as a starting point to determine a most complete representative for a given group using the given annotation for further investigation.

As for future improvements of the *LTRdigest* software, it will certainly be useful to extend the protein domain detection to derive a possible open reading frame from the protein hits, which can then be annotated as a *CDS* or *transposable_element_gene* SO entity in the GFF3 output. This functionality would widen the model of internal polyprotein genes from single domain hits to whole translational units.

CONCLUSION

While fine-grained annotation of genomic features is common in the context of gene prediction, which aims at precisely determining the intron–exon boundaries, most information about LTR retrotransposon insertions is still limited to broad-scale sequence-based matches. In summary, our software will be helpful to improve this data by annotating LTR retrotransposons and ERVs, especially in the context of *de novo* identification projects. In such projects, this annotation data will prove valuable in separating full-length copies from truncated insertions and other retrotransposons. It also facilitates classification of LTR retrotransposons into families. The results can then be used for further analyses, e.g. for building a reference sequence library or for phylogenetic studies. The feature identification process utilizes parallelization and requires little memory, facilitating analyses of large candidate sets, e.g. from mammalian genomes. Results of exemplary applications on the *M. musculus* and *D. melanogaster* genomes show that both the software and the methods presented here are suitable for such applications.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

The authors would like to thank Jan Sellmann for large-scale testing of *LTRdigest* and providing valuable bug reports as well as helpful suggestions for usability improvement.

FUNDING

Funding for open access charge: Center for Bioinformatics, University of Hamburg, Bundesstrasse 43, 20146 Hamburg, Germany.

Conflict of interest statement. None declared.

REFERENCES

1. Biémont,C. and Vieira,C. (2006) Junk DNA as an evolutionary force. *Nature*, **443**, 521–524.
2. Finnegan,D.J. (1989) Eukaryotic transposable elements and genome evolution. *Trends Genet*., **5**, 103–107.
3. Jern,P. and Coffin,J.M. (2008) Effects of retroviruses on host genome function. *Annu. Rev. Genet*., **42**, 709–732.
4. Maksakova,I.A., Mager,D.L. and Reiss,D. (2008) Keeping active endogenous retroviral-like elements in check: the epigenetic perspective. *Cell Mol. Life Sci*., **65**, 3329–3347.
5. Slotkin,R.K. and Martienssen,R. (2007) Transposable elements and the epigenetic regulation of the genome. *Nat. Rev. Genet*., **8**, 272–285.
6. Bergman,C.M. and Quesneville,H. (2007) Discovering and detecting transposable elements in genome sequences. *Brief. Bioinform*., **8**, 382–392.
7. 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**, 462–467.
8. 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.
9. Xu,Z. and Wang,H. (2007) LTR_FINDER: an efficient tool for the prediction of full-length LTR retrotransposons. *Nucleic Acids Res*., **35**, W265–W268.
10. McCarthy,E.M. and McDonald,J.F. (2003) LTR_STRUC: a novel search and identification program for LTR retrotransposons. *Bioinformatics*, **19**, 362–367.
11. Kalyanaraman,A. and Aluru,S. (2006) Efficient algorithms and software for detection of full-length LTR retrotransposons. *J. Bioinform. Comput. Biol*., **4**, 197–216.
12. Ellinghaus,D., Kurtz,S. and Willhoeft,U. (2008) *LTRharvest*, an efficient and flexible software for de novo detection of LTR retrotransposons. *BMC Bioinformatics*, **9**, 18.
13. Vogt,P.K. (1997) Retroviral virions and genomes. In Coffin,J.M., Hughes,S.H. and Varmus,H.E. (eds), *Retroviruses*. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, New York.
14. Wilhelm,M. and Wilhelm,F.X. (2001) Reverse transcription of retroviruses and LTR retrotransposons. *Cell Mol. Life. Sci*., **58**, 1246–1262.
15. Wilhelm,M., Uzun,O., Mules,E.H., Gabriel,A. and Wilhelm,F.X. (2001) Polypurine tract formation by Ty1 RNase H. *J. Biol. Chem*., **276**, 47695–47701.
16. Wilhelm,M., Heyman,T., Boutabout,M. and Wilhelm,F.X. (1999) A sequence immediately upstream of the plus-strand primer is essential for plus-strand DNA synthesis of the *Saccharomyces cerevisiae* Ty1 retrotransposon. *Nucleic Acids Res*., **27**, 4547–4552.
17. Marquet,R., Isel,C., Ehresmann,C. and Ehresmann,B. (1995) tRNAs as primer of reverse transcriptases. *Biochimie*, **77**, 113–124.
18. Mak,J. and Kleiman,L. (1997) Primer tRNAs for reverse transcription. *J. Virol*., **71**, 8087–8095.
19. Havecker,E.R., Gao,X. and Voytas,D.F. (2004) The diversity of LTR retrotransposons. *Genome Biol*., **5**, 225.
20. Sperber,G.O., Airola,T., Jern,P. and Blomberg,J. (2007) Automated recognition of retroviral sequences in genomic data–RetroTector. *Nucleic Acids Res*., **35**, 4964–4976.
21. Feschotte,C., Keswani,U., Ranganathan,N., Guibotsy,M.L. and Levine,D. (2009) Exploring repetitive DNA landscapes using REP-CLASS, a tool that automates the classification of transposable elements in eukaryotic genomes. *Genome Biol. Evol*., **2009**, 205–220.
22. Abrusán,G., Grundmann,N., DeMester,L. and Makalowski,W. (2009) TEclass–a tool for automated classification of unknown eukaryotic transposable elements. *Bioinformatics*, **25**, 1329–1330.
23. Lowe,T.M. and Eddy,S.R. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res*., **25**, 955–964.
24. Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol*., **147**, 195–197.
25. Durbin,R., Eddy,S.R., Krogh,A. and Mitchison,G. (2006) *Biological Sequence Analysis – Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge.
26. Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.
27. Finn,R.D., Tate,J., Mistry,J., Coggill,P.C., Sammut,S.J., Hotz,H.-R., Ceric,G., Forslund,K., Eddy,S.R., Sonnhammer,E.L.L. *et al*. (2008) The Pfam protein families database. *Nucleic Acids Res*., **36**, D281–D288.
28. Gremme,G., Brendel,V., Sparks,M.E. and Kurtz,S. (2005) Engineering a software tool for gene structure prediction in higher organisms. *Inf. Softw. Technol*., **47**, 965–978.
29. Eilbeck,K., Lewis,S.E., Mungall,C.J., Yandell,M., Stein,L., Durbin,R. and Ashburner,M. (2005) The sequence ontology: a tool for the unification of genome annotations. *Genome Biol*., **6**, R44.
30. Tweedie,S., Ashburner,M., Falls,K., Leyland,P., McQuilton,P., Marygold,S., Millburn,G., Osumi-Sutherland,D., Schroeder,A., Seal,R. *et al*. (2009) Flybase: enhancing *Drosophila* gene ontology annotations. *Nucleic Acids Res*., **37**, D555–D559.
31. Chan,P.P. and Lowe,T.M. (2009) GtRNAdb: a database of transfer RNA genes detected in genomic sequence. *Nucleic Acids Res*., **37**, D93–D97.
32. Wicker,T., Sabot,F., Hua-Van,A., Bennetzen,J.L., Capy,P., Chalhoub,B., Flavell,A., Leroy,P., Morgante,M. *et al*. (2007) A unified classification system for eukaryotic transposable elements. *Nat. Rev. Genet*., **8**, 973–982.
33. Llorens,C., Futami,R., Bezemer,D. and Moya,A. (2008) The Gypsy Database (GyDB) of mobile genetic elements. *Nucleic Acids Res*., **36**, D38–D46.
34. Hubbard,T.J.P., Aken,B.L., Ayling,S., Ballester,B., Beal,K., Bragin,E., Brent,S., Chen,Y., Clapham,P., Clarke,L. *et al*. (2009) Ensembl 2009. *Nucleic Acids Res*., **37**, D690–D697.
35. McCarthy,E.M. and McDonald,J.F. (2004) Long terminal repeat retrotransposons of *Mus musculus*. *Genome Biol*., **5**, R14.
36. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol*., **215**, 403–410.
37. Rice,P., Longden,I. and Bleasby,A. (2000) EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet*., **16**, 276–277.
38. 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.
39. Bartolome,C., Bello,X. and Maside,X. (2009) Widespread evidence for horizontal transfer of transposable elements across *Drosophila* genomes. *Genome Biol*., **10**, R22.
40. Kaminker,J.S., Bergman,C.M., Kronmiller,B., Carlson,J., Svirskas,R., Patel,S., Frise,E., Wheeler,D.A., Lewis,S.E., Rubin,G.M. *et al*. (2002) The transposable elements of the *Drosophila melanogaster* euchromatin: a genomics perspective. *Genome Biol*., **3**, RESEARCH0084.
41. Bergman,C.M., Quesneville,H., Anxolabéhère,D. and Ashburner,M. (2006) Recurrent insertion and duplication generate networks of transposable element sequences in the *Drosophila melanogaster* genome. *Genome Biol*., **7**, R112.
42. Steinbiss,S., Gremme,G., Schärfer,C., Mader,M. and Kurtz,S. (2009) *AnnotationSketch*: a genome annotation drawing library. *Bioinformatics*, **25**, 533–534.