LTRharvest
a manual

David Ellinghaus
Stefan Kurtz
Ute Willhoeft

Research Group for Genomeinformatics
Center for Bioinformatics
University of Hamburg
Bundesstrasse 43
20146 Hamburg
Germany

willhoeft@zbh.uni-hamburg.de

In any documentation or publication about research using *LTRharvest* please cite the following paper:

D. Ellinghaus, S. Kurtz, and U. Willhoeft. *LTRharvest*, an efficient and flexible software for de novo detection of LTR retrotransposons, BMC Bioinformatics 2008, 9:18
http://www.biomedcentral.com/1471-2105/9/18

August 6, 2012

1 Introduction

This document describes *LTRharvest* [1], a software tool for *de novo* predictions of LTR retrotransposons in genomic sequences [4]. *LTRharvest* computes boundary positions of potential LTR retrotransposons on a persistent index structure of the genomic target sequence, the *enhanced suffix array* [3]. Usually the genomic target sequence is a complete chromosomal DNA sequence in FASTA format. Nevertheless, every DNA sequence in (multiple) FASTA format can be passed to the software.

For the prediction, *LTRharvest* implements several filters. These are consecutively applied on the sequence data to reject candidates, which are not conform with sequence, length or distance features of LTR retrotransposons. Since these features are mostly species-specific, every filter can be switched on or switched off and is free for parameterisation of a certain LTR retrotransposon model.

LTRharvest is written in `C` and it is based on the *GenomeTools* library [5]. *LTRharvest* is called as part of the single binary named `gt`. *LTRharvest* runs on an enhanced suffix array index, which is stored on files. This index needs to be constructed by the program `suffixerator`, which is also part of the *GenomeTools* binary `gt`.

2 Usage

Some text is highlighted by different fonts according to the following rules.

- `Typewriter font` is used for the names of software tools.
- `Small typewriter font` is used for file names.
- `Footnote sized typewriter font` with a leading `'-'` is used for program options.
- *small italic font* is used for the argument(s) of an option.

2.1 The options of *LTRharvest*

Since *LTRharvest* is part of `gt`, *LTRharvest* is called as follows.

`gt ltrharvest` `-index` *indexname* [*options*]

where *indexname* denotes the enhanced suffix array index of the target sequence(s) constructed by the program *Suffixerator*. An overview of all possible options with a short one-line description of each option is given in Table 1. All options can be specified only once.

2.2 Input options

`-index` *indexname*

Specify the name of the enhanced suffix array index. The index must comprise the tables `suf`, `lcp`, `des` and `tis`. An example of how the enhanced suffix array index is constructed with the program *Suffixerator* can be found in section 3.

Table 1: Overview of the *LTRharvest* options sorted by categories.

Input options	
`-index`	specify the name of the enhanced suffix array index
`-range`	specify the range in the input sequences to be searched for LTR retrotransposon candidates
Output options	
`-out`	specify outputfilename for predictions in multiple FASTA format
`-gff3`	specify outputfilename for predictions in GFF3 format
`-outinner`	specify outputfilename for multiple FASTA file of the inner regions of predictions
Filter options	
`-seed`	specify minimum seed length for exact maximal repeats
`-minlenltr`	specify minimum length for each LTR
`-maxlenltr`	specify maximum length for each LTR
`-mindistltr`	specify minimum distance of LTR startpositions
`-maxdistltr`	specify maximum distance of LTR startpositions
`-similar`	specify similaritythreshold in range [1..100%]
`-mintsd`	specify minimum length for each TSD
`-maxtsd`	specify maximum length for each TSD (requires `-mintsd`)
`-motif`	specify palindromic motif consisting of 2 nucleotides startmotif + 2 nucleotides endmotif,
`-motifmis`	specify maximum number of mismatches in motif (requires `-motif`)
`-vic`	specify the number of nucleotides (to the left and to the right) that will be searched for TSDs and/or motifs around 5' and 3' boundary
`-overlaps`	specify no|best|all
Alignment options	
`-xdrop`	specify xdrop value for seed extension
`-mat`	specify matchscore for seed extension
`-mis`	specify mismatchscore for seed extension
`-ins`	specify insertionscore for seed extension
`-del`	specify deletionscore for seed extension
Miscellaneous options	
`-v`	verbose mode
`-longoutput`	additional motif/TSD output (requires `-mintsd` or `-motif`)
`-help`	show all options

The option `-index` is mandatory.

`-range` x y
: Specify the range in the input sequence(s) to be searched for LTR retrotransposon candidates. That is, if $x = 1000$ and $y = 10000$ then candidates are only reported if they start after position 1000 and end before position 10000 in their respective sequence coordinates.

  If $x = y = 0$ (default), then the whole sequences are searched.

2.3 Output options

Results are reported in tabular fashion on stdout and can easily be written to a file using the notation > *resultfile* as in the following:

`gt ltrharvest -index` *indexname* [*options*] > *resultfile*

In addition, the following options write extra information to files

`-out` *outputfile*
: Specify the name of a file where the predictions will be written to. Each prediction will be represented by an individual FASTA entry.

`-outinner` *outputfile*
: Specify the name of a file where the inner regions (the prediction sequences without the flanking LTR sequences) will be written to. Each prediction will be represented by an individual FASTA entry.

`-gff3` *outputfile*
: Specify the name of a file where the predictions will be written to. Each prediction will be represented by an individual GFF3 [2] entry.

2.4 Filter options

The filter options provide the opportunity to exclude predictions with unwanted sequence, length or distance features. If a particular option is not set by the user, a default value for this options will be set, except for the options `-mintsd`, `-maxtsd` and `-motif`. Thus, if none of the filter options is set by the user, a prediction of LTR retrotransposons will be conducted without searching for target site duplications (TSD) or a particular LTR start-end motif.

`-seed` L_{ex}
: Specify the minimum length for the exact maximal repeats. Only those repeats with the specified minimum length are analyzed in the process of finding candidate pairs. Exact maximal repeats of length below that threshold are not taken further into account. L_{ex} has to be a positive integer. If this option is not selected by the user, then L_{ex} is set to 30 by default.

`-minlenltr` L_{min}
: Specify the minimum length of each LTR. L_{min} has to be specified as a positive integer. If this option is not selected by the user, then L_{min} is set to 100 by default.

`-maxlenltr` L_{max}

Specify the maximum length of each LTR. L_{max} has to be specified as a positive integer. If this option is not selected by the user, then L_{max} is set to 1000 by default.

`-mindistltr` D_{min}

Specify the minimum distance of LTR starting positions. D_{min} has to be specified as a positive integer. If this option is not selected by the user, then D_{min} is set to 1000 by default.

`-maxdistltr` D_{max}

Specify the maximum distance of LTR starting positions. D_{max} has to be specified as a positive integer. If this option is not selected by the user, then D_{max} is set to 15000 by default.

`-similar` *similaritythreshold*

Specify the minimum similarity value between the two LTRs. The argument *similaritythreshold* has to be chosen from the range [0,100] and means a percentage. If this option is not selected by the user, the default value is set to 85%.

`-mintsd` *TSDminlen*

If this option is selected, a search for target site duplications (TSDs) will be conducted with a minimum TSD length of *TSDminlen*. If this option is not selected by the user, a search for TSDs with minimum TSD length 4 will be conducted. If this option is set but no maximum TSD length is specified by the option `-maxtsd`, then the maximum TSD length is set to 20 by default.

`-maxtsd` *TSDmaxlen*

This option requires the option `-mintsd`. If this option is selected, a search for target site duplications (TSDs) will be conducted with a maximum TSD length of *TSDmaxlen*.

`-motif` *expr*

Specify 2 nucleotides for the starting motif and 2 nucleotides for the ending motif at the beginning and the ending of each LTR, respectively. Only palindromic motif sequences - where the motif sequence is equal to its complementary sequence read backwards - are allowed, e.g. *tgca*. Type the nucleotides without any space separating them. If this option is not selected by the user, then candidate pairs will not be checked, if they contain a motif. If this options is set but no allowed number of mismatches is specified by the option `-motifmis`, then a search for the exact motif will be conducted.

`-motifmis` *n*

This option requires the option `-motif`. Specify the allowed number of mismatches by the argument *n*. If this option is not set, then a search for the exact motif will be conducted. The non-negative integer *n* has to be chosen from the range $[0, 3]$.

`-vic` *l*

Specify the number of nucleotide positions *l* to the left and to the right (the vicinity), that will be searched for TSDs and/or one motif around the 5' and 3' predicted boundary of a LTR retrotransposon. This option has only an effect, if option `-mintsd` and/or option `-motif` is switched on. If this option is not selected by the user, the default value of l is 60.

`-overlaps` *no*|*best*|*all*

Specify the output with regard to nested and/or overlapping LTR retrotransposon predictions. If the argument *no* is selected, then neither nested nor overlapping predictions will be reported in the output. If the argument *best* is selected, then, in the case of two or more nested or overlapping predictions, solely the LTR retrotransposon prediction with the highest similarity between its LTRs will be reported. If the argument *all* is selected, then all LTR retrotransposon predictions will be reported whether there are nested and/or overlapping predictions or not. If this option is not selected by the user, the option with argument *best* is set by default.

2.5 Alignment options

An X-drop extension process permits the search for degenerated LTRs. The alignment options provide the opportunity to control this extension of the maximal repeat seeds. If a particular alignment option is not set by the user, a default value for this options will be set.

`-xdrop` *X*

Specify the xdrop value *X* for extending a seed repeat in both directions allowing for matches, mismatches, insertions, and deletions. The argument *X* must be a positive integer or 0. The X-drop extension process stops as soon as the extension involving matches, mismatches, insertions, and deletions has a score smaller than $T - X$ where T denotes the largest score seen so far. If this option is not selected by the user, then *X* is set to 5 by default.

`-mat` *score*

Specify the positive match score for the X-drop extension process. If the option is not selected by the user, the default value is 2.

`-mis` *score*

Specify the negative mismatch score for the X-drop extension process. If this option is not selected by the user, the default value is -2.

`-ins` *score*

Specify the negative insertion score for the X-drop extension process. If this option is not selected by the user, the default value is -3.

`-del` *score*

Specify the negative deletion score for the X-drop extension process. If this option is not selected by the user, the default value is -3.

2.6 Miscellaneous options

`-v`

This option enables the verbose mode. This means, that some more information about the processing will be printed to `stdout` during the run. This includes a long list of switched on or switched off options.

`-longoutput`

This option additionally prints information about the detected TSD and/or the motif to `stdout`, if a search for TSD and/or for the motif has been selected by the user. This option requires the option `-mintsd` and/or `-motif`.

`-help`

LTRharvest will show a summary of all options on `stdout` and terminate with exit code 0.

3 Examples

In this section, example applications of *LTRharvest* are presented. In Subsection 3.1, examples for using different options of *LTRharvest* are given. Subsection 3.2 then gives an example for the prediction of LTR retrotransposons on the entire *S. cerevisiae* genome. In Subsection 3.3, an example for a clustering process of the *LTRharvest* output is shown. Please note that this step is not part of *LTRharvest* and is carried out by *Vmatch* [7].

3.1 Using different options of *LTRharvest*

As target sequence file we use some FASTA file `chr02.19970727.fsa.gz` containing the *S. cerevisiae* genome sequence, chromosome 2. Note that the file is a compressed file in gzip format (because of the ending `.gz`). This format can be handled by the program *Suffixerator*.

First, we create the enhanced suffix array. We invoke `gt suffixerator` with options `-tis`, `-suf`, `-lcp`, `-des`, `-ssp` and `-sds` since *LTRharvest* needs the corresponding tables. Furthermore, we specify `-dna`, as we process DNA-sequences.

```
$ gt suffixerator -db chr02.19970727.fsa.gz -indexname chr02.19970727.fsa -tis
-suf -lcp -des -ssp -sds -dna
# dna=yes
# indexname="chr02.19970727.fsa"
.
.
.
# TIME overall 1.37
```

Now we can use the index for *LTRharvest*. The first example call will just use the default parameters for the filter and alignment options without searching for TSD or an LTR start-end motif.

```
$ gt ltrharvest -index chr02.19970727.fsa
# args=-index chr02.19970727.fsa
# predictions are reported in the following way
# s(ret) e(ret) l(ret) s(lLTR) e(lLTR) l(lLTR) s(rLTR) e(rLTR) l(rLTR)
sim(LTRs) seq-nr
# where:
# s = starting position
# e = ending position
# l = length
# ret = LTR-retrotransposon
```

```
# lLTR = left LTR
# rLTR = right LTR
# sim = similarity
# seq-nr = sequence number
259532  265448  5917  259532  259863  332  265117  265448  332  99.40  0
427672  430021  2350  427672  428170  499  429522  430021  500  91.60  0
29632  35597  5966  29632  29969  338  35259  35597  339  98.82  0
220989  226919  5931  220989  221339  351  226574  226919  346  97.15  0
```

Each comment line starts with the comment symbol #. Each non-comment line denotes a LTR retrotransposon prediction with starting and ending positions of the whole LTR retrotransposon, the left LTR instance and the right LTR instance, respectively. Furthermore, for each of these elements, the corresponding element length is reported as well as a percentage similarity of the two LTRs. The last integer of each line denotes the number of the input sequence, the LTR retrotransposon prediction occurs in. The input sequence numbers are counted from 0.

Invoking *LTRharvest* with the optional argument `-v` gives more information about enabled and disabled options as well as additional information about the enhanced suffix array index and time/space consumption. Moreover, specifying options `-out` and `-outinner` the run results in two multiple FASTA files.

```
$ gt ltrharvest -index chr02.19970727.fsa -v -out pred-chr02.fsa
-outinner pred-inner-chr02.fsa
# args=-index chr02.19970727.fsa -v -out pred-chr02.fsa -outinner
pred-inner-chr02.fsa
# user defined options and values:
#   verbosemode: On
#   indexname: chr02.19970727.fsa
#   outputfile: pred-chr02.fsa
#   outputfile inner region: pred-inner-chr02.fsa
#   xdropbelowscore: 5
#   similaritythreshold: 85.00
#   minseedlength: 30
#   matchscore: 2
#   mismatchscore: -2
#   insertionscore: -3
#   deletionscore: -3
#   minLTRlength: 100
#   maxLTRlength: 1000
#   minLTRdistance: 1000
#   maxLTRdistance: 15000
#   overlaps: best
#   minTSDlength: 0
#   maxTSDlength: 20
#   palindromic motif:
#   motifmismatchesallowed: 4
#   vicinity: 60 nt
# predictions are reported in the following way
# s(ret) e(ret) l(ret) s(lLTR) e(lLTR) l(lLTR) s(rLTR) e(rLTR) l(rLTR) sim(LTRs)
seq-nr
# where:
# s = starting position
# e = ending position
# l = length
```

```
# ret = LTR-retrotransposon
# lLTR = left LTR
# rLTR = right LTR
# sim = similarity
# seq-nr = sequence number
259532  265448  5917  259532  259863  332  265117  265448  332  99.40  0
427672  430021  2350  427672  428170  499  429522  430021  500  91.60  0
29632  35597  5966  29632  29969  338  35259  35597  339  98.82  0
220989  226919  5931  220989  221339  351  226574  226919  346  97.15  0
```

Searching additionally for TSD and a LTR start-end motif we use the options `-mintsd`, `-maxtsd`, `-motif` and `-motifmis`.

```
$ gt ltrharvest -index chr02.19970727.fsa -mintsd 5 -maxtsd 20 -motif tgca
-motifmis 0
# args=-index chr02.19970727.fsa -mintsd 5 -maxtsd 20 -motif tgca -motifmis 0
# predictions are reported in the following way
# s(ret) e(ret) l(ret) s(lLTR) e(lLTR) l(lLTR) s(rLTR) e(rLTR) l(rLTR) sim(LTRs)
seq-nr
# where:
# s = starting position
# e = ending position
# l = length
# ret = LTR-retrotransposon
# lLTR = left LTR
# rLTR = right LTR
# sim = similarity
# seq-nr = sequence number
259532  265448  5917  259532  259863  332  265117  265448  332  99.40  0
29632  35590  5959  29632  29963  332  35259  35590  332  99.70  0
220996  226911  5916  220996  221329  334  226575  226911  337  97.33  0
```

Finally, if we are interested in the sequence and the length of the TSD as well as the sequence of the motif, we select the option `-longoutput`. This also is an example for specifying all filter and alignment options by the user.

```
$ gt ltrharvest -index chr02.19970727.fsa -seed 30 -xdrop 5 -mat 2 -mis -2 -ins -3
-del -3 -minlenltr 100 -maxlenltr 1000 -mindistltr 1000 -maxdistltr 15000
-similar 90.0 -overlaps all -mintsd 5 -maxtsd 20 -motif tgca -motifmis 0 -vic 60
-longoutput
# args=-index chr02.19970727.fsa -seed 30 -xdrop 5 -mat 2 -mis -2 -ins -3 -del -3
-minlenltr 100 -maxlenltr 1000 -mindistltr 1000 -maxdistltr 15000 -similar 90.0
-overlaps all -mintsd 5 -maxtsd 20 -motif tgca -motifmis 0 -vic 60 -longoutput
# predictions are reported in the following way
# s(ret) e(ret) l(ret) s(lLTR) e(lLTR) l(lLTR) TSD l(TSD) m(lLTR) s(rLTR)
e(rLTR) l(rLTR) TSD l(TSD) m(rLTR) sim(LTRs) seq-nr
# where:
# s = starting position
# e = ending position
# l = length
# m = motif
# ret = LTR-retrotransposon
# lLTR = left LTR
```

```
# rLTR = right LTR
# TSD = target site duplication
# sim = similarity
# seq-nr = sequence number
259532  265448  5917  259532  259863  332  gtaat  5  tg..ca  265117  265448  332
gtaat  5  tg..ca  99.40  0
29632  35590  5959  29632  29963  332  ataat  5  tg..ca  35259  35590  332
ataat  5  tg..ca  99.70  0
220996  226911  5916  220996  221329  334  ggaat  5  tg..ca  226575  226911 337
ggaat  5  tg..ca  97.33  0
```

3.2 Predictions on the entire *S.cerevisiae* genome

As target sequences file we use some multiple FASTA file `chrAll.19971001.fsa.gz`, which contains all 16 chromosomal sequences of the release before Oct. 1st, 1997, probably used by Kim et al. in their comprehensive survey of retrotransposons [6].

First, we create the enhanced suffix array. We invoke `gt suffixerator` with options `-tis`, `-suf`, `-lcp`, `-des`, since *LTRharvest* needs the corresponding tables. Furthermore, we specify option `-dna`, as we are processing DNA sequences.

```
$ gt suffixerator -db chrAll.19971001.fsa.gz -indexname chrAll.19971001.fsa -tis
-suf -lcp -des -sds -dna
# dna=yes
# indexname="chrAll.19971001.fsa"
.
.
.
# TIME overall 108.35
```

Now we can use the index with *LTRharvest*. In addition to the filter and alignment options, we choose option `-out` for printing the predicted LTR retrotransposon sequences to a file.

```
$ gt ltrharvest -index chrAll.19971001.fsa -seed 100 -minlenltr 100 -maxlenltr 1000
-mindistltr 1000 -maxdistltr 15000 -xdrop 5 -mat 2 -mis -2 -ins -3 -del -3
-similar 90.0 -overlaps best -mintsd 5 -maxtsd 20 -motif tgca -motifmis 0
-vic 60 -longoutput
-out pred-chrAll.fsa
# args=-index chrAll.19971001.fsa -seed 100 -minlenltr 100 -maxlenltr 1000
-mindistltr 1000 -maxdistltr 15000 -xdrop 5 -mat 2 -mis -2 -ins -3 -del -3
-similar 90.0 -overlaps best -mintsd 5 -maxtsd 20 -motif tgca -motifmis 0
-vic 60 -longoutput -out pred-chrAll.fsa
# predictions are reported in the following way
# s(ret) e(ret) l(ret) s(lLTR) e(lLTR) l(lLTR) TSD l(TSD) m(lLTR) s(rLTR) e(rLTR)
l(rLTR) TSD l(TSD) m(rLTR) sim(LTRs) seq-nr
# where:
160239  166163  5925  160239  160575  337  ggttc  5  tg..ca  165827  166163  337
ggttc  5  tg..ca  100.00  0
29632  35590  5959  29632  29963  332  ataat  5  tg..ca  35259  35590  332
ataat  5  tg..ca  99.70  1
.
.
```

```
.
844407  850335  5929  844407  844744  338  gaaat  5  tg..ca  849998  850335  338
gaaat  5  tg..ca  100.00  15
56452  62375  5924  56452  56788  337  gttat  5  tg..ca  62039  62375  337
gttat  5  tg..ca  100.00  15
```

3.3 Sequence clustering of *LTRharvest* output (optional)

In addition to the prediction process done by *LTRharvest*, a cluster analysis on the predicted sequences is recommended. Here, we choose the single linkage cluster analysis program from the *Vmatch* software tool [7] (which is not part of the *GenomeTools* binary `gt`) in order to show how this task can be accomplished. An index needs to be constructed from the predicted sequences by the program `mkvtree` which is part of *Vmatch*.

```
$ mkvtree -db pred-chrAll.fsa -dna -pl -allout -v
reading file "pred-chrAll.fsa"
...
.
.
.
overall space peak: main=2.61 MB (10.01 bytes/symbol), secondary=0.53 MB
```

Now we can use `vmatch` (which is part of *Vmatch*) to perform a clustering of the predicted sequences. A reasonable parameter set is given in Tab. 2, page 11. See the *Vmatch manual* [7] for an explanation of the options. The following command computes clusters of all (database) sequences in `pred-chrAll.fsa`. Each cluster has a unique cluster number, followed by the sequence number contained in the cluster.

```
$ vmatch -dbcluster 95 7 Cluster-pred-chrAll -p -d -seedlength 50
-l 1101 -exdrop 9 pred-chrAll.fsa
# args=-dbcluster 95 7 Cluster-pred-chrAll -p -d -seedlength 50
-l 1101 -exdrop 9 pred-chrAll.fsa
# 3 clusters
# 45 elements out of 46 (97.83%) are in clusters
# 1 elements out of 46 (2.17%) are singlets
# 1 cluster of size 2
# 1 cluster of size 3
# 1 cluster of size 40
0:  32 41 42 36 39 13 40 43 37 30 44 38 18 21 17 34 35 31 12 20
33 23 26 27 6 28 3 29 25 1 24 9 19 2 7 22 4 10 5 0
1:  8 15 11
2:  14 45
```

References

[1] `http://www.zbh.uni-hamburg.de/ltrharvest`

[2] GFF3, tab delimited file format for genome annotation. `http://www.sequenceontology.org/gff3.shtml`.

Table 2: A default parameter set for *Vmatch*'s single linkage clustering program.

Parameter name	Value	Comment
dbcluster	95 ($\lfloor(\frac{D_{min}}{D_{max}} \times 100)\rfloor + 1$)	Match covers at least 95% of the smaller sequence and (*formula value*)% of the larger sequence
d		Compute direct matches
p		Compute palindromic matches
seedlength	50	Minimal length of the exact repeats
l	$D_{min} + L_{min}$	Minimal length of matches
exdrop	9	Xdrop score when extending a seed in both directions

[3] M.I. Abouelhoda, S. Kurtz, and E. Ohlebusch. Replacing suffix trees with enhanced suffix arrays. *Journal of Discrete Algorithms*, 2:53–86, 2004.

[4] D. Ellinghaus, S. Kurtz, and U. Willhoeft. *LTRharvest*, an efficient and flexible software for de novo detection of LTR retrotransposons. *BMC Bioinformatics*, 9:18, 2008.

[5] G. Gremme. The GENOMETOOLS genome analysis system. `http://genometools.org`.

[6] J.M. Kim, S. Vanguri, J.D. Boeke, A. Gabriel, and D.F. Voytas. Transposable elements and genome organisation: a comprehensive survey of retrotransposons revealed by the complete *Saccharomyces cerevisiae* genome sequence. *Genome Research*, 8(5):464–478, 1998.

[7] S. Kurtz. The VMATCH large scale sequence analysis software. `http://www.vmatch.de`.