HelitronScanner 1.1 user manual

HelitronScanner is a two-layered local combinational variable (LCV) tool for generalized *Helitron* identification that is considerably superior to previous *Helitron* identification programs based on DNA sequence or structure.

SYSTEM REQUIREMENTS

HelitronScanner requires JRE 1.6+. A full list of Oracle Certified Configuration for Java SE 6 can be found at http://www.oracle.com/technetwork/java/javase/system-configurations-135212.html. To determine if the java run time is available in your system, please open a system terminal, and type 'java -version' (without quotes). If you have not installed Java yet, please go to the official Java website (http://java.com/en/download/) and download Java.

INSTALLATION

HelitronScanner was developed on the Java Virtual Machine (JVM) platform. Users can easily download the zip file and unzip it to a local directory. No compilation is needed thanks to JVM's cross platform compatibility.

QUICK START

Simply download and unzip the software archive to a local directory. Open a system terminal and change to the unzipped directory. Then type the following to see a list of available commands.

```
java -jar HelitronScanner.jar
```

COMMAND LINE ENTRY

In order to identify putative *Helitrons* from input genomic sequences in Fasta format, HelitronScanner works in several steps, each achieved by a separate command, as described in COMMANDS section.

```
java -jar HelitronScanner.jar <Command> [options]
```

Please note that if the current working directory is not where the HelitronScanner.jar file is, you need to provide the full or relative path for it.

For example,

```
java -jar /Users/xxx/HelitronScanner.jar ...
```

COMMANDS

For any command, you can provide a **-help** option following the command name to get details of this command. For example,

```
java -jar HelitronScanner.jar scanHead -h
```

Options are listed in brackets, with usually a full name for clarity (e.g. -lcv_filepath) and a shortened one (e.g. -lf) for simplicity.

scanHead/scanTail

These two commands scan for *Helitron* 5' (scanHead) and 3' (scanTail) termini, respectively, and share the command options.

LCVs extracted from a training set of published *Helitrons* are the key features to determine *Helitron* termini based on matching scores. The LCV file specified by -lcv_filepath option should be a text file, each line representing a valid LCV in java regular expression syntax. Any line starting with a # sign will be ignored.

```
   * Options are mandatory

          [-lcv_filepath, -lf] LCV file path        | default: [Standard console input]
*         [-genome, -g]       Genome data to scan for Helitrons  [must be a valid file path]
*         [-buffer_size, -bs]Genome slice size (use negative or zero for non-buffer, i.e. treat every whole
chromosome)
          [-output, -o]       Output file for match scores          | default: [Standard console output]
          [-threshold, -th]   Threshold for the minimum match score      | default: [1]
          [-threads_LCV, -tl]            Number of threads used for LCV regex match            | default: [1]
          [-overlap, -ol]     Slice overlap size  | default: [50]
          [--rc, --rc_mode]  Scan for the complementary strand instead. | default: [false]
          [-h, -help, /h, /help, --help]           Show help information      | default: [false]
```

Here, the default threshold is 1, which allows any matches to LCVs. The location scores can be further filtered in **pairends** command, with default threshold of 5 for *Helitron's* both ends.

Only the forward strand of DNA sequences will be scanned. Explicitly provide the --rc option to scan the reverse strand.

To avoid excessive memory usage, use -buffer_size 1000000 (1 Mb slice at a time) for whole genome scans. Otherwise, use -buffer_size 0 instead.

The output location scores of these two commands are usually saved to local disks for the next step.

pairends

This command takes location scores from both *Helitron* ends and pairs them to create coordinates/scores for putative *Helitrons*.
Default thresholds of 5 for each *Helitron* end are used in the manuscript, but they can be adjusted separately to suit user needs of sensitivity and specificity. Higher thresholds lead to lower false positive rates, while increasing the probability of missing low-scoring real *Helitrons*.

Helitrons are normally between 200 bp and 20 kb, as defined by the -helitron_len_range option. Users can specify a broader or narrower range in the form of *min_len:max_len*.
If the score files are generated in reverse complementary mode, the --rc option has to be provided accordingly, as well.

The output file of this command stores coordinates and scores of putative *Helitrons*, which is fed to the **draw** command to get *Helitron* sequences from the original genomic sequences.

```
        * Options are mandatory

*       [-head_score, -hs]          Score file from 5' end.       [must be a valid file path]
*       [-tail_score, -ts]   Score file from 3' end.     [must be a valid file path]
        [-output, -o]        Output file for paired scores         | default: [Standard console output]
        [-head_threshold, -ht]      Threshold for the minimum match score at 5' end     | default: [5]
        [-tail_threshold, -tt]      Threshold for the minimum match score at 3' end     | default: [5]
        [-helitron_len_range, -hlr] Helitron length range      | default: [200:20000]
        [--rc, --rc_mode]    Reverse complementary mode      | default: [false]
        [-h, -help, /h, /help, --help]         Show help information     | default: [false]
```

draw

This command takes the pair-end score file ([-pscore, -p] option) generated by **pairends** command as input and draw *Helitrons* or their related regions from the original input file ([-genome, -g] option). The original genomic sequences from input file must be the exact same ones that generate the pair-end scores; otherwise the results are meaningless.

Sometimes it is convenient to get the flanking regions of *Helitrons* for further investigations. The **draw** command provides three options for such purposes. Use -pure_helitron option to get just *Helitron* sequences, -extended_helitron option to get *Helitrons* with their flanking regions, and -joint_flanking option to get flanking regions only. These 3 options can be used at the same time or in desired combinations, but at least one option must be specified.
The lengths of flanking region at both ends are defined by -ext5 and -ext3 options, respectively. They will not affect the -pure_helitron mode.

```
     * Options are mandatory

        [-pscore, -p]        Pair-end score file path     | default: [Standard console input]
*       [-genome, -g]        Genome file path [must be a valid file path]
*       [-output, -o]        Template for output file paths (xxx). At most 3 files will be created, xxx.hel.fa,
xxx.ext.hel.fa and xxx.flanking.fa for --pure, --ext and --flanking options respectively.
        [-ext5]    Extension length at 5' end  | default: [0]
        [-ext3]    Extension length at 3' end  | default: [0]
        [-pure_helitron, --pure]    Draw Helitrons only        | default: [false]
        [-extended_helitron, --ext]         Draw Helitrons with their flanking regions    | default: [false]
        [-joint_flanking, --flanking]       Draw joint flanking sequences only | default: [false]
        [-h, -help, /h, /help, --help]      Show help information     | default: [false]
```

Thanks for using HelitronScanner. If you find it useful for your work, please cite:

> **Wenwei Xiong[1], Limei He[2], Jinsheng Lai[3], Hugo K. Dooner[2,4], Chunguang Du[1] ***
> **HelitronScanner uncovers a large overlooked cache of *Helitron* transposons in many plant genomes (2014)**

Please find the latest version of HelitronScanner at http://bo.csam.montclair.edu/du/software/helitronscanner.