

# Package ‘HextractorR’

May 7, 2026

**Type** Package

**Title** Integrated Tool for Hairpin Extraction of RNA Sequences

**Version** 1.4

**Author** Cristian Yones

**Maintainer** Cristian Yones <cyones@sinc.unl.edu.ar>

**Description** Simple and integrated tool that automatically extracts and folds all hairpin sequences from raw genome-wide data. It predicts the secondary structure of several overlapped segments, with longer length than the mean length of sequences of interest for the species under processing, ensuring that no one is lost nor inappropriately cut.

**License** Apache License 2.0

**Encoding** UTF-8

**LazyData** true

**Imports** seqinr, parallel, doParallel, foreach,

**NeedsCompilation** no

**RoxygenNote** 6.1.1

**Repository** CRAN

**Date/Publication** 2019-08-06 16:20:02 UTC

## Contents

HextractorR .....	2
<b>Index</b>	<b>4</b>

---

HextractoR                      *HextractoR: Integrated Tool for Hairpin Extraction of RNA Sequences*

---

## Description

To preprocess a genome, you need a file containing the raw genome in fasta format. To run HExtractor, simply call the main function. This function creates 2 files in the "out" folder and automatically names them.

## Usage

```
HextractoR(input_file, min_valid_nucleotides = 500, window_size = 160,
           window_step = 30, only_sloop = T, min_length = 60, min_bp = 16,
           trim_sequences = T, margin_bp = 6, blast_evalue = 1,
           identity_threshold = 90, nthreads = 4, nworks = 4,
           filter_files = { })
```

## Arguments

<code>input_file</code>	filename of the fasta file to process
<code>min_valid_nucleotides</code>	Each input sequence must have this quantity of valid nucleotides (not 'N') to be processed.
<code>window_size</code>	Number of bases in the windows.
<code>window_step</code>	Window step. This number defines indirectly the overlap: <code>window_overlap=window_size-window_step</code>
<code>only_sloop</code>	Only extract single loop sequence.
<code>min_length</code>	Minimum sequence length. Shorter sequences are discarded.
<code>min_bp</code>	Minimum number of base-pairs that must form a sequence.
<code>trim_sequences</code>	Use some heuristics to trim the hairpins.
<code>margin_bp</code>	When the sequence is trimmed, at least <code>min_bp+margin_bp</code> base-pairs are left.
<code>blast_evalue</code>	e-value used in blast to match the extracted sequences with the sequences from the filter files.
<code>identity_threshold</code>	Identity threshold used to match sequences with the sequences from the filter files.
<code>nthreads</code>	Allows using more than one thread in the execution.
<code>nworks</code>	Split each sequence in <code>nworks</code> to use less RAM memory.
<code>filter_files</code>	Fasta files with known sequences to separate the output stems.

## Value

A list with the path of the output files and the result of the processing of each sequence (if it was succesful or failed)

**Examples**

```
# Small example without filter files
library(HextractoR)
# First we get the path of the example FASTA file
fpath <- system.file("Example_tiny.fasta", package="HextractoR")
# To run HextractoR, simply call the main function
HextractoR(input_file = fpath)
# Other example with filter files and bigger input file
fpath1 <- system.file("Example_human.fasta", package="HextractoR")
fpath2 <- system.file("Example_pre-miRNA.fasta", package="HextractoR")
HextractoR(input_file = fpath1, filter_files = {fpath2})
# This function creates 2 files in the working directory and automatically
# names them.
```

# Index

HextractoR, [2](#)