NAME

vsearch — dereplicate, filter, sort, search, compare and clusterize amplicons from metagenomic projects

SYNOPSIS

vsearch [*options*] *filename*

DESCRIPTION

Environmental or clinical molecular studies generate large volumes of amplicons (e.g. SSU-rRNA sequences) that need to be filtered, dereplicated, searched, clustered or compared to sequences from other studies. The aim of **vsearch** is to offer a all-in-one open source tool to perform these tasks, using optimized algorithm implementations and harvesting the full potential of modern computers to guarantee the fastest and more accurate possible processing.

Nucleotidic sequence comparisons is at the core of **vsearch**. To speed up comparisons, **vsearch** implements *k*-mer filtering, and an extremely fast Needleman-Wunsch algorithm making use of the Streaming SIMD Extensions (SSE2) of modern x86-64 CPUs. If SSE2 instructions are not available, **vsearch** exits with an error message.

vsearch input is a fasta file containing one or several nucleotidic sequences. For each sequence, the sequence identifier is defined as the string comprised between the ">" symbol and the first space or the end of the line, whichever comes first. Additionally, if the line ends with the pattern ";size=*integer*;", **vsearch** will interpret *integer* as the abundance of the sequence (in a dereplicated fasta file for instance). The nucleotidic sequence is defined as a string of [acgt] or [acgu] symbols (case insensitive), starting after the end of the identifier line and ending before the next identifier line or the file end; **vsearch** exits with an error message if any other symbol is present in the sequence. Optionally, **vsearch** can be compiled to accepted compressed fasta files as input (gz and bzip2 formats).

Options

vsearch recognizes a large number of command-line options. For an easier navigation, options are grouped by theme (dereplication, filtering, sorting, searching, comparison, clustering). We start with general options that apply to all themes.

General options:

--help display a short help and exit.

--version output version information and exit.

--fasta_width *positive integer*
fasta files produced by **vsearch** are wrapped (sequences written on lines of *integer* nucleotides, 80 by default). Set that value to 0 to eliminate the wrapping.

--maxseqlength *positive integer*
all **vsearch** operations will discard sequences of length equal or greater than *integer* (50,000 nucleotides by default).

--minseqlength *positive integer*
all **vsearch** operations will discard sequences of length smaller than *integer* (1 nucleotide by default for sorting or shuffling, 32 nucleotides for dereplication or searching).

--notrunclabels
do not truncate sequence labels at first space, use the full header.

--strand *plus*|*both*
when searching or dereplicating, check the *plus* strand only (default) or check *both* strands.

--threads *positive integer*
number of computation threads to use. The number of threads should be lesser or equal to the number of available CPU cores. Default number of threads is 1, use 0 to launch a number of threads equal to the number of CPU cores.

--uc *filename*
when searching or dereplicating, output results in *filename* using a uclust-like format.

--uc_allhits
when searching or dereplicating, and when using the uclust-like format option, show all hits, not just top hit.

Clustering options:

--uc *filename*
use the fast clustering algorithm and write the results to *filename*.

--cluster_smallmem *filename*
use a slower clustering algorithm (consumes less memory) and write the results to *filename*.

--centroids *filename*
output cluster centroid sequences to *filename* file.

--clusters *string*
output each cluster to a separate fasta file using *string* as prefix for the filenames.

--consout *filename*
output cluster consensus sequences to *filename* file.

--construncate
when using the consout option, do not ignore terminal gaps in the multiple sequence alignment when building the consensus sequence.

--msaout *filename*
output multiple sequence alignments to *filename*.

--usersort
conserve the initial input order of sequences, do not sort sequences by decreasing length.

Dereplication, masking, shuffling and sorting options:

--derep_fulllength *filename*
merge strictly identical sequences contained in *filename*. Redundant sequences receive the header of the sequence of their group, and the number of occurrences (abundance) is indicated at the end of the fasta header using the pattern ";size=X;".

--maskfasta *filename* mask sequences contained in *filename*.

--shuffle *filename* pseudo-randomly shuffle the order of sequences contained in *filename*.

--sortbylength *filename* sort by decreasing length the sequences contained in *filename*.

--sortbysize *filename* sort by decreasing abundance the sequences contained in *filename*.

--maxsize *positive integer* when using --sortbysize, discard sequences with an abundance value greater than *integer*.

--minsize *positive integer* when using --sortbysize, discard sequences with an abundance value smaller than *integer*.

--minuniquesize *positive integer* when dereplicating, discard sequences with an abundance value smaller than *integer*.

--output *filename* when dereplicating, sorting or shuffling, write the results to *filename*.

--relabel *string* when sorting, relabel sequence headers using *string* as suffix.

--seed *positive integer* when shuffling, use *integer* as seed. Set to 0 to use a pseudo-random seed.

--sizein read abundance annotation from input

--sizeout add abundance annotation to output

--topn positive integer when dereplicating, sorting or shuffling, output just the top *integer* sequences.

Searching options: (in progress)

EXAMPLES

(in progress)

LIMITATIONS

vsearch does not perform chimera detection..

AUTHORS

Implementation by Torbjørn Rognes, documentation by Frédéric Mahé, .

REPORTING BUGS

Submit suggestions and bug-reports at <https://github.com/torognes/vsearch/issues>, send a pull request on <https://github.com/torognes/vsearch>, or compose a friendly or curmudgeont e-mail to Torbjørn Rognes <torognes@ifi.uio.no>.

AVAILABILITY

The software is available from <https://github.com/torognes/vsearch>

COPYRIGHT

SEE ALSO

swipe, an extremely fast Smith-Waterman database search tool by Torbjørn Rognes (available from <https://github.com/torognes/swipe>).

VERSION HISTORY

New features and important modifications of **vsearch** (short lived or minor bug releases are not mentioned):

v1.0 released November 1st, 2014
First public release