NAME

nexalign - aligns sequences to databases

SYNOPSIS

nexalign *[options]* -p *query database1 database2 ...* -o *output*

DESCRIPTION

Nexalign aligns sequence reads from a single *query* file to multiple *database* files and writes the alignments to *output*.

Nexalign is designed to work efficiently with millions of short reads generated by e.g. Roche's 454, Illumina's Solexa or ABI's SOliD sequencing technology. Nexalign reports all alignments with up to one insertion/deletion or with up to 4 mismatches. In addition to the alignments themselves, nexalign can output the mapped or unmapped tags in fasta format facilitating the construction of complex mapping pipelines.

The *query* file can be in either standard fasta or color-space fasta format (XXX.csfasta). All *database* files have to be in standard fasta format. If no *query* file is specified nexalign tries to read sequences from standard input.

If sequence names in the *query* file are followed by *'_X'* nexalign assumes that *'X'* is the number of times the sequence was observed. Otherwise it assign a count of 1 to all sequences. This default behavior can be changed using the **-expression-char** option.

ALIGNMENT OPTIONS

-e Reports exact alignments.

-m <*x*>
Reports alignments with up to *x* mismatches.

-indel Reports alignments with one insertion or deletion.

Note: Each alignment option turns off the other two alignments options. I.e. -m *x* will produce mismatched alignments but not exact alignments. To obtain both exact and mismatched alignments both options have to be used: " -e -m 1".

-solid Align SOliD color space sequences.

SEQUENCE SELECTION OPTIONS

-min-len <*x*>
Align tags equal or longer than *'X'* (default: 0).

-max-len <*x*>
Align tags equal or shorter than *'X'* (default: 500).

-chop_at <*x*>
Align only the first <*x*> bases of every tag. This is applied after the -min-len / -max-len checks.

-frag <*x*>
Fragments the input tags into <*x*> -mers. The sequence counts are divided up equally among the fragments. This option is applied after all of the above.

OUTPUT OPTIONS

-o <*f*> Writes alignments to file <*f*>.

-unmapped <*f*>
Writes all unmapped sequences to file <*f*>.

-mapped <*f*>
Writes all mapped sequences to file <*f*>. A sequence is considered mapped if it aligns to less than 100 places in all input *databases*.

-multi <*f*>
Writes all ambiguously mapped sequences to file <*f*>. A sequence is considered to be ambiguously mapped if it aligns to more than 100 places in all input *databases*.

-u <*x*> Specifies the maximum number of alignments reported per *query* sequence. This option will not stop reporting alignments at a certain point but rather only report alignments for sequences that map less than <*x*> times. Additionally, this option is used to modify the cutoffs for options **-mapped** and **-multi.** For example, "-u 1 -mapped <*f*> " will write alignments for sequences mapping only once and print the corresponding sequences to file <*f*>. Exception: Using "-u 0 -mapped <*f*> " will not result in an empty <*f*> as expected, but will instead print all mapping sequences to <*f*>. *Default:* 100.

REPORTING OPTIONS

-stats <*f*>
Prints overall mapping statistics to file <*f*>. If the R statistical package is available pdf and jpeg files are also created. Setting <f> to STDOUT will print the statistics to the standard output.

-count <*f*>
Prints a summary of overall mapping statistics for each sequence to file <*f*>.

-split <*f*>
Prints a summary of mapping statistics for each sequence and each *database* to file <*f*>.

ANNOTATION OPTIONS

-ann <*f*>
Prints an overview of the percentage of sequences mapping to specific *databases* to file <*f*>. If the R statistical package is available pdf and jpeg files are also created. Setting <f> to STDOUT will print the statistics to the standard output.

-max_no_ann <*x*>
This option modifies the behavior of "-ann". <*x*> specifies the maximum number of annotation databases a sequence can map to before being considered ambiguously annotated. *Default:* 2.

-types <*f*>
When working with large annotation databases it is desirable to split them up into several smaller fasta files due to potential memory limitations. For nexalign all the database files belonging to the same annotation category should contain a TAG in the filename in the format "_TAG". The file <*f*> contains a list of all TAGs telling nexalign which annotation files belong together.

CLUSTERING OPTIONS

-c <*f*> Clusters mapping locations in file <*f*> into tag clusters. The file needs to be in standard nexalign output format.

-cluster-window <*x*>
Adds a window of <*x*> to each mapping position for clustering purposes.

-gff <*f*>
Creates a gff file containing the clusters.

-bed <*f*>
Reformats clusters into bedgraph formatted files for both the negative and positive strand.

-description <*x*>
Add a UCSC viewable description to the bedgraph files.

Note: Using options **-gff** or **-bed** will cause nexalign to automatically cluster the mapping output.

OTHER OPTIONS

-h Prints a brief help.

-q Quiet mode - nothing is written to STDERR.

-w Creates indices for each *database* file. These indices are written to the same directory as the *database* files. To create indices for SOliD data use '-w' in combination with '-solid' (see below).

-time Reports the running time of nexalign.

-threads <*x*>
Number of threads used (default: 16).

-expression-char <*c*>
Use <*c*> instead of underscore to indicate the counts of tags in the input.

-uniq Merges identical input sequences into single query sequences. The names of the individual sequences are lost. Fasta files produced by nexalign will contain the sequence itself as the name followed by '_*X*' where *X* reflects how often the sequence was observed in the input.

Note: If the hash_size (see below) is smaller that the number of sequences in the *query* file, nexalign will only merge identical sequences within smaller chunks of the *query*. In this case it is recommended to either increase the hash_size or convert the *query* file into a file including the '_*X*' naming convention.

-hash_size <*x*>
Defines the number of unique query sequences nexalign can hold in memory. If smaller than the number of input sequences <*y*> , all databases have to be searched <*y*> / <*x*> times. On the other hand, defining <*x*> much larger than <*y*> increases execution time since nexalign always allocates a datastructure of size <*x*>. *Default:* 10000000.

EXAMPLES

Hierarchical Mapping:

nexalign -e -p query.fa /hg18/*.fa -o exact_mappings.csv -unmapped STDOUT | nexalign -m 1 /hg18/*.fa -o 1_mismatch_mappings.csv -unmapped STDOUT | nexalign -m 2 /hg18/*.fa -o 2_mismatch_mappings.csv

Maps query sequences to the human genome exactly, takes the remaining unmapped sequences and maps them first with 1, then with 2 mismatches.

Annotation:

nexalign -e -p query.fa exon.fa intron.fa promoter.fa -ann annotation.csv -u 0 -max_no_ann 1

Displays the proportion of query sequences mapping to exons, introns, promoters. Since max_no_ann is set to one all sequences mapping to more than one category will be displayed as ambiguously annotated.

Nexalign Queries :

nexalign -p query.fa ribosomal.fa -unmapped STDOUT | nexalign miRNA_hairpin.fa -mapped STDOUT | nexalign /hg18/*.fa -u 1 -mapped map.fa > mappings.csv

Extracts all sequences from "query.fa" that do not map to ribosomal sequences, but map uniquely to miRNA hairpin sequences.

COPYRIGHT

AUTHOR

Timo Lassmann
Researcher LSA Bioinformatics Team
Omics Science Center (OSC),
Riken Yokohama Institute
1-7-22 Suehiro-chu, Tsurumi-ku, Yokohama
230-0045 Kanagawa, Japan

email: timolassmann@gmail.com
http://genome.gsc.riken.jp/osc/english/members/Timo_Lassmann.html