NAME

lucy − Assembly Sequence Cleanup Program

SYNOPSIS

lucy [-pass_along min_value max_value med_value]
[-range area1 area2 area3] [-alignment area1 area2 area3]
[-vector vector_sequence_file splice_site_file]
[-cdna [minimum_span maximum_error initial_search_range]] [-keep]
[-size vector_tag_size] [-threshold vector_cutoff]
[-minimum good_sequence_length] [-debug [filename]]
[-output sequence_filename quality_filename]
[-error max_avg_error max_error_at_ends]
[-window window_size max_avg_error
[window_size max_avg_error ...]]
[-bracket window_size max_avg_error]
[-quiet] [-inform_me] [-xtra cpu_threads]
sequence_file quality_file [2nd_sequence_file]

DESCRIPTION

Lucy is a utility that prepares raw DNA sequence fragments for sequence assembly, possibly using the *TIGR Assembler.* Raw DNA sequence fragments are obtained from DNA sequencing machines, such as those from the Applied Biosystems Inc. (ABI). *Lucy* accepts three input data files: the **sequences file,** the **quality file,** and (optionally) a **second sequence file** for comparison purposes. All three files should be in the plain FASTA format.

The first sequence file and its accompanying quality file are obtained from other utility programs, such as *phred(1),* which reads the sequencing machine chromatograph outputs and generates sequence base calls in a sequence file together with a quality assessment file for each base call made. The optional second sequence file usually comes directly from the sequencing machine itself and are used to reassure/enhance the quality of the sequences.

Lucy makes no assumption about the order of sequences in the three input files. As long as all necessary information can be found, DNA sequences and quality sequences can be in different order. A sequence without its quality assessment companion or vice versa will be reported as an error. The second sequence file is allowed to have missing sequences that appear only in the first sequence file. Sequences that appear only in the second sequence file will be reported and ignored by *lucy.*

The operations of *Lucy* are divided into 7 phases:

Phase 1:

Count the number of input sequences in the first sequence file, and create internal data structures accordingly. *Lucy* allocates memory dynamically, so there is no preset limit on the size of input files. The size of input files is only limited by your available computer memory.

Note that lucy's static memory requirement grows very slowly with the number of input sequences, at roughly 60 bytes plus the sequence name storage for each input sequence. This is because lucy does not store any sequence data in the memory after processing them. Therefore, by all practical considerations lucy can handle any number of input sequences. The dynamic memory requirement of lucy is proportional to the longest sequence in the input, not the number of sequences, but its actual size varies from processing phases to phases.

Phase 2:

Read all sequence information, including name, length, and positions. To save memory, *lucy* does not load all sequences data into main memory at once. Instead, it uses direct file addressing to access each sequence only when it is needed. Therefore, it is very important that the content of the input files stays unchanged during runtime.

Phase 3:

Read the quality information for sequences, and compute good quality regions, i.e., regions within sequences that have higher quality values and can be trusted to be correct. *Lucy* determines a "clean" range which has an average probability of error (per base) that is no greater than the probability specified by the -error option (or the default, if -error is not used). Note that secondary sequence extension (next step) is performed after quality trimming. Because of this, it is possible that the final "clear" range (after vector trimming) will have a probability of error which is greater than the specified value.

Phase 4:

Read the second sequence file, compare its sequences to the first sequence file, and extend good quality regions if they both agree. If the second sequence file is not provided, this step is skipped. Note that this sequence comparison phase will not in anyway shorten the good quality region determined by the previous phase; it will only extend it if possible. It is very important that the second sequence file does *not* come from the same base calling software as the first sequence file, and is base-called with *different* algorithms. Otherwise, if the two sequence files are identical, *lucy* will extend sequences all the way to both ends, and completely ruin the purpose of quality trimming done in the previous phase.

Usually, the first sequence file comes from *phred(1)* with the companion quality file, and the second sequence file comes directly from the original ABI base calling software with the sequencing machines.

Phase 5:

Chop off splice sites from the sequences. In this phase, *lucy* tries to compare all input sequences against splice site sequences in a splice site file which defines the vector sequences near the insertion point on the vector. If splice site sequences are found on any input sequence, they will be excluded from the good quality region so that the sequence assembly program will not mistakenly take them into account when trying to reassemble the sequences. Note that *lucy* assumes all input sequences are read from the same direction and matching the direction of the splice site sequences. Therefore, the forward and reverse read sequences of a clone should not be mixed together in a single input file. If such a mixture of forward and reverse read sequences is unavoidable, *lucy* can be run twice to check in both directions, once with the forward splice site sequences, the other time with the reverse splice site sequences. See the description of option **-vector** below for more details.

By popular demand, a poly-A/T trimming feature has been built into *lucy.* It is designated Phase 5a and is an optional step. See the options **-cdna** and **-keep** below for details of their usage.

Phase 6:

Remove vector insert sequences. In this phase, all input sequences are checked against a full length vector sequence in a vector file, and sequences that are vector inserts themselves will be detected and removed. *Lucy* uses a quick fragment match method to check for vector sequences. Both the target vector sequence and the input sequences are converted into fragments (range from 8 to 16 bases long, default is 10), and matching fragments are detected quickly. Vector sequences are detected when they contain more matches to vector fragments in their good quality region (already excluded of splice site sequences done previously) than a normal, non-vector sequence can possibly match by chance. The default cutoff threshold is 20%. A sequence which contains over 20% match to the vector will be considered a vector insert and discarded.

Phase 7:

Produce output sequences for fragment assembly. In the final phase, *Lucy* produces two output files, the cleaned sequence file with markers for good quality regions, and a companion quality file. Optionally, *lucy* can also generate a cleavage information file (i.e. the good quality region information) which can be used to update database.

Each sequence in the output sequence file begins with a header that includes its name, three pass along

clone length values to the fragment assembly program, and a left and right marker denoting the begin and end of the **good** quality, vector free region. The following is an example of *lucy* output:

```
>GCCAA03TF 1500 3000 2000 43 490
AGCCAAGTTTGCAGCCCTGCAGGTCGACTCTAGAGGATCCCCAGGATGATCAGCCACATT
GGGACTGAGACACGGCCCAAACTCCTACGGGAGGCAGCAGTGGGGAATCTTGCGCAATGG
GCGAAAGCCTGACGCAGCCATGCCGCGTGAATGATGAAGGTCTTAGGATTGTAAAATTCT
TTCACCGGGGACGATAATGACGGTACCCGGAGAAGAAGCCCCGGCTAACTTCGTGCCAGC
   ...
```

OPTIONS

Note that *lucy* checks only the first letter of each option, so all options below can be represented by just typing the first letter, e.g. -p for -pass_along.

−pass_along min_value max_value med_value

The three pass along values of minimum, maximum and medium clone lengths are given using this option. *Lucy* does not interpret these values; they are used by some sequence assembly programs, such as *TIGR Assembler.* These values are directly copied over to the output sequence file. The default values are 0, 0, and 0.

−range area1 area2 area3

This option is used in combination with the following option **-alignment.** It defines the three splice site checking areas which may need different strengths of splice site alignment. The quality of the base calls is usually poor at the beginning of a sequence but gradually improves when moving into the sequence read. Therefore, when looking for splice sites, stronger and stronger alignment measurements are needed to cope with the quality change. The default range values are 40, 60 and 100, i.e., *lucy* will check for splice sites at the first 200 DNA bases. If splice site is not found at the first 200 bases, the next 100 (=area3) bases will be checked, with a total checking length of 300. Once a splice site is found, the rest of the sequence after the splice site is searched for the other end of the splice site, if any, to guard against short inserts.

−alignment area1 area2 area3

This option is used in combination with the previous option. It defines the three different alignment strengths for the three areas. An alignment within each area must be equal or longer than these values before it is considered a match of the splice site. Default values are 8, 12 and 16 for the first 40, 60 and 100 bases, respectively.

−vector vector_sequence_file splice_site_file

This option provides the complete vector sequence file and a partial splice site sequence file. *Lucy* expects to see one single (probably long) sequence of the vector that is used to do cloning in the vector file, and **two** splice site sequences **before** and **after** the insertion point on the vector in the splice site file. The splice site sequences are usually 150 bases in length, with a 50 bases overlay right around the vector insertion point. Their actual lengths are not very critical. For example, the followings are the PUC18 splice site sequences that can be used by *lucy:*

```
>PUCsplice.for.begin
gattaagttgggtaacgccagggttttcccagtcacgacgttgtaaaacg
acggccagtgccaagcttgcatgcctgcaggtcgactctagaggatcccc
gggtaccgagctcgaattcgtaatcatggtcatagctgtttcctgtgtga
>PUCsplice.for.end
acggccagtgccaagcttgcatgcctgcaggtcgactctagaggatcccc
gggtaccgagctcgaattcgtaatcatggtcatagctgtttcctgtgtga
aattgttatccgctcacaattccacacaacatacgagccggaagcataaa
```

With two splice site sequences as above, *lucy* assumes all sequences from the input files are read in the same direction as the splice site sequences. If that is not true and the input consists of sequences from both forward and reverse reads of a clone, there are two options. One can either separate the sequences into forward and reverse read sets and run them through *lucy* with correct

splice site sequences. One can also run *lucy* with a combined splice site file with **both** the forward and reverse splice site sequence pairs. That is, if *lucy* sees four splice site sequences, it will assume that a **bidirectional** splice site trimming has been ordered. For example, the following reverse PUC18 splice site sequences can be appended to the forward splice sequences above to instruct *lucy* to do bidirectional trimmings:

>PUCsplice.rev.begin
tttatgcttccggctcgtatgttgtgtggaattgtgagcggataacaatt
tcacacaggaaacagctatgaccatgattacgaattcgagctcggtaccc
ggggatcctctagagtcgacctgcaggcatgcaagcttggcactggccgt
>PUCsplice.rev.end
tcacacaggaaacagctatgaccatgattacgaattcgagctcggtaccc
ggggatcctctagagtcgacctgcaggcatgcaagcttggcactggccgt
cgttttacaacgtcgtgactgggaaaaccctggcgttacccaacttaatc

Bidirectional trimmings will run about two times slower because each sequence is compared against two sets of splice site sequences when only one set is actually needed. It is possible that random alignments with the other (unneeded) set will result in somehow a shortened good quality region of a sequence. However, bidirectional trimmings can guarantee that there are no vector fragments in the good quality region even when the assumed direction of some sequence reads is wrong.

During **phase 6** contaminant removal, *lucy* will automatically reverse complement the full length vector sequence and check for both its forward and reverse inserts, so only one vector sequence is needed in the vector file. *Lucy* can also get the vector and splice site file names from the environment variables **VECTOR_FILE** and **SPLICE_FILE,** if they are not given at the command line.

−cdna [minimum_span maximum_error initial_search_range]

Since the release of *lucy*, many people have requested that a poly-A/T trimming feature be built into *lucy* for the convenience of people doing cDNA sequencing. This option is added for that purpose. By default *lucy* will not do this step, unless this option is given. This option can be given alone without any parameter, in that case the default values will be used, or it can be given with **all** three parameters. The **minimum_span** defines the minimum length of *continuous* poly-A or T before *lucy* believes that it has found them. The **maximum_error** denotes the maximum number of errors allowed before a new continuous poly-A/T region stops. If mismatch error count goes beyond maximum_error, then *lucy* believes that it has reached the end of the poly-A/T tail/head. Note that each new continuous poly-A/T region that is more than minimum_span long will reset the error counter to zero, therefore an interleaving number of maximum_error followed by minimum_span can keep the poly-A/T region expanding. The last parameter **initial_search_range** denotes the range from the ends of a sequence within which a minimum_span number of continuous poly-A/T's have to be found, otherwise *lucy* believes that it cannot find poly-A/T regions for the sequence. Note that the ends of the sequence are related to the clear region **after** vector splice sites trimming, not to the actual physical ends of the sequence. The default values of the three parameter are minimum_span=10, maximum_error=3 and initial_search_range=50. **Warning:** these three parameters have to be set carefully in order to avoid throwing good sequences away in the middle of a sequence, or to let poly-A/T regions slip by at the ends of a sequence.

−keep When **-cdna** option is turned on, *lucy* will trim all poly-A/T fragments it finds. This is good for EST clustering purposes, where you don't want the poly-A/T fragments to stay. However, if you want to see the EST sequence in its entirety to know its direction, it is not helpful to trim poly-A/T away. The **-keep** option, when used in combination with the **-cdna** option, will preserve the poly-A/T tails/heads at ends of each EST sequence to keep them as tags indicating the direction of the EST sequence.

−size vector_tag_size

This option is used in combination with the following option. It defines the size of fragments for checking vector using the fragment matching algorithm. The default value is 10 bases. The range of acceptable values are 8 to 16.

−threshold vector_cutoff

The option is used in combination with the previous option. It defines the threshold of similarity between a sequence and the vector for it to be considered a vector insert. Since splice sites are not included in vector screening, any sequence which has a higher than normal similarity to the vector sequence will be considered a vector itself and discarded. The default value of cutoff is 20% of the good quality region.

−minimum good_sequence_length

After all kinds of checking, comparing and trimming, the good region of a sequence must still be long enough than the minimum length for it to be considered useful to the sequence assembly program. We do not want our sequence assembly program to be bothered by many small, trashy fragments. The default minimum good sequence length is 100 bases.

−debug [filename]

This option, if given, tells *lucy* to produce a sequence cleavage information file for reference or for updating the database. The default file name is "lucy.debug", which can be overridden by the given file name.

−output sequence_filename quality_filename

This option defines the output sequence and quality file names. If not given, the default file names *lucy* uses are "lucy.seq" and "lucy.qul".

−error max_avg_error max_error_at_ends

There are three main steps in the quality trimming performed by *lucy.* The first involves removing low-quality bases from each end of the sequence, using the criteria specified by the **-bracket** option. The second involves finding regions of the sequence where the probability of error meets all of the criteria specified by the **-window** option. After these regions are found, the third step is to trim each of them to the largest region having an average probability of error no greater than the max_avg_error specified by the **-error** option. Finally, the largest region meeting all of the criteria is chosen as the final "clean" range.

Two parameters are specified with this option: max_avg_error is the maximum acceptable average probability of error over the final clean range. max_error_at_ends is the maximum probability of error that is allowed for the 2 bases at each end of the final clean range. The defaults are 0.025 and 0.02, respectively, if **-error** is not specified.

Note: A base's estimated probability of error is calculated from the quality value that is assigned by the base caller. The quality value (Q) is defined as:

Q = -10 * log10(Probability of error)

−window window_size max_avg_error [window_size max_avg_error ...]

This option affects the quality trimming of the sequence (see the description of the **-error** option, above). It specifies one or more window sizes, and a maximum allowable average probability of error for each of those window sizes. If more than one window size is specified, they must be specified in decreasing order by window size. The maximum number of windows that may be specified is 20.

Lucy uses a sliding window algorithm to find regions of the sequence within which the average probability of error, within any window of the specified size, is no greater than the specified max_avg_error. Regions which meet all of the specified window criteria are then trimmed again using the criteria specified by the **-error** option, and the final "clean" range is the largest region that meets all of the criteria.

If the **-window** option is not specified, then *lucy* uses 2 windows by default, of 50 and 10 bases. The default maximum allowable probabilities of error in the two windows are listed below:

50-base window: 0.08
10-base window: 0.3

−bracket window_size max_avg_error

This option controls the initial quality trimming step, which is the removal of low-quality bases from both ends of the sequence. *lucy* looks for the first and last window of size window_size having an average probability of error no greater than max_avg_error. The subsequence which extends from the first base of the first such window to the last base of the last window is then examined further to find the clean range. Bases which precede the first window or follow the last window are excluded from the clean range (so the two terminal windows bracket the clean range).

The defaults for window_size and max_avg_error are 10 and 0.02.

−quiet Tells *lucy* to shut up and only report serious errors it finds. :)

−inform_me

Asks *lucy* to report sequences by names that have been thrown out due to low quality values, or salvaged due to comparison to the 2nd sequence file.

−xtra cpu_threads

If you have multiple CPUs in your computer, you can dramatically increase lucy's speed by allowing lucy to run multiple execution *threads* concurrently. For example, if you have a dual-CPU computer, you can give the option **-xtra 2** to cut lucy's execution time roughly in half. By default, lucy will run just one thread if this option is not given. The maximum number of allowable threads is 32. Note that this option is only available with the multi-threaded lucy version 1.16p. There is also a 1.16s version that does not do multi-threading.

ENVIRONMENT

The environment variable **VECTOR_FILE** defines the vector file name, and the environment variable **SPLICE_FILE** defines the splice site file name. Both variables are used when the user does not specify them using the **-vector** option.

SEE ALSO

TIGR_Assembler(1), grim(1), everm.sp(1), phred(1), TraceTuner(1), and ethyl.pl(1).

BUGS

No known bugs for the program at this moment. Some of the manual pages mentioned above do not exist.

CAVEATS

The "no bugs" claim above can never be true. This is a new program built mostly from scratch, and there must be bugs somewhere, somehow. Please direct all bug reports to the authors.

ACRONYM

Lucy stands for **Less Useful Chunks Yank,** an awkward combination of words in order to make it a member of the family with phred, the base caller, ethyl, the old scripting system lucy replaced, and ricky, the database linking and communication driving software of lucy for use in TIGR.

AUTHOR

Lucy was written by Hui-Hsien Chou and Michael Holmes at The Institute for Genomic Research, with help and suggestions from Granger Sutton, Anna Glodek, John Scott, and Terry Shea. Michael Holmes is currently responsible for *lucy.* Please direct any suggestions, bug reports, etc. to mholmes@tigr.org.