SeqTrim: a Validation and Trimming Tool for All Purpose Sequence Reads

Juan Falgueras[1], Antonio J. Lara[2], Francisco R. Cantón[2], Guillermo Pérez-Trabado[3], and M. Gonzalo Claros[4]

[1]Lenguajes y Ciencias de la Computación, ETSI Informática, Campus Universitario de Teatinos, s/n. E-29071, Málaga, Spain
[2]Biología Molecular y Bioquímica, Universidad de Málaga, Campus Universitario de Teatinos, s/n. E-29071, Málaga, Spain
[3]Arquitectura de Computadores, ETSI Informática, Campus de Teatinos, E-29071 Málaga, Spain
[4]Departamento de Biología Molecular y Bioquímica. Facultad de Ciencias Universidad de Málaga. 29071 Málaga, Spain. Tel: +34 95 213 72 84. Fax: +34 95 213 20 41
claros@uma.es

Abstract. Bioinformatics tools are required to produce reliable, high quality data devoid of unwanted sequences in the preprocessing stage of current sequencing and EST projects. In this paper we describe SeqTrim, an algorithm designed to extract the insert sequence from any sequence read devoid of any foreign, contaminant or unwanted sequence, whatever the experimental process was. SeqTrim is easy to install and able to identify the sequence insert by removing low quality sequences, cloning vector, poly A or T tails, adaptors, and sequences that can be considered contaminants. It is easy to use and can be used as stand-alone application or as web page. The default parameters of the algorithm are best suited for most cases but a configuration file can be provided along with input sequences. SeqTrim admits several input and output formats (with and without quality values), which enables its inclusion in already or newly defined sequence processing workflows. SeqTrim is under continuous refinement due to collaboration between biologists and computer scientists which has succeed in correct dealing with most sequence cases and opens the possibility to include new capabilities to manage new kinds of bad sequences.

Introduction

Sequencing projects and EST data have proven to be an important resource for gene discovery and mapping, and promise to be invaluable for the annotation of the eukaryotic and prokaryotic genomes by providing sequence information to identify novel genes, gene location and even intron-exon boundaries. They are helped by the availability of high throughput automated sequencing, which has enabled the exponential growth rate of these experimentally determined sequences, although not always with the desired quality. However, this exponential growth requires efforts to be made in increasing the quality and reliability of sequences incorporated into databases, since there is a high percent of nucleotides in the databases corresponding

to contaminant sequences [1]. Hence, bioinformatic tools are required to produce reliable, high quality data devoid of unwanted sequences with efficient preprocessing methods.

Preprocessing includes filtering of low-quality sequences, identification of sequence features, removal of contaminant sequences (from vector to any other artifacts) and trimming the undesired segments. Though there are some bioinformatic tools that serve to accomplish individual parts of the preprocessing (e.g. TrimSeq, TrimEST, VectorStrip, VecScreen, ESTPrep, crossmatch), currently used programs that deal with the complete preprocessing are, Lucy [2], SeqClean (`http://www.tigr.org/tdb/tgi/software`) or PreGap4 [3]. However, many of these programs are hard to configure, environment specific, or focused on specific needs (like only pre-process ESTs), and it is not always easy to connect them with further processing tools for annotation or assembling, for example. Moreover, using such programs typically requires a change in implementation and design of either the program or the protocols within the laboratory itself.

This paper presents a new program, SeqTrim, which has been designed to extract the insert sequence from any sequence read devoid of any foreign, contaminant or unwanted sequence, whatever the experimental process was. SeqTrim is easy to install and able to identify the sequence insert by removing low quality sequences, removing cloning vector, removing any special feature (like poly A or poly T tails if present, restriction sites used for cloning, terminal transferase tails if present, sequence adaptors, etc.). Configuration parameters can be customizable transiently or permanently by the user. SeqTrim is provided as a command line tool or as a web tool (`http://castanea.ac.uma.es/genuma/seqtrim`). Although it works as a standalone application, it is intended to work in a web server as a part of an automated workflow that, starting with raw sequences in a database finishes in the design of contigs and the annotation of them.

Overview

Although there are many DNA sequence comparison and analysis algorithms in the bioinformatics literature, getting them to work correctly may be very hard, as is the case of Lucy [2, 8], phrap or consed [9]. In other cases, getting them to process high throughput data requires an extra programming effort to consider unlikely special cases that appear when handling large amounts of sequences or the data quality are very low [2]. Due to inherently unpredictable nature of biological data, there may be some distance between the theoretical design of a bioinformatic solution and the successful implementation of the solution in a reliable, working program that can deal with large amounts of data. Since the only way to shorten such distance is to collaborate between computer scientists and biologists, SeqTrim emerged from such collaboration since 1999 in the University of Málaga (Spain). Along these years, the algorithm has been trained with real data obtained from previous, different researches ([10]; R Bautista, FR Cantón, C Avila and MG Claros, unpublished results). In other words, it has been under continuous development, testing and production, which make it well suited for the needs of large-scale sequence preprocessing, providing a

time- and cost-effective solution. The use of SeqTrim has significantly reduced the time and complexity involved in numerous gene discovery projects at the University of Málaga.

Installation of SeqTrim does no require special knowledge since there is nothing to compile and the only previous requirements are: (i) presence of blast, (ii) presence of bioperl libraries and, (iii) presence of phred. SeqTrim itself only needs uncompression and moving the 'seqtrim' directory to `/usr/local` or create a soft link in the `/usr/local` directory to your preferred location of the seqtrim folder. This transparency is also extended to the configuration: SeqTrim is provided with databases that will be used for vector and contaminant detection, an editable list of restriction enzymes, adaptors and configuration parameters (see above). The configuration file contains the default parameters that can serve to run SeqTrim with results of many sequencing projects. In the seek of customization, a wide range of parameters are available even if we know that most of them will never be changed. Default parameter modification will be permanent only if it is done directly within the `seqtrim.conf` file; otherwise, default values are used for each run.

Sequence preprocessors are not expected to be used alone but in a pipeline with other programs [11, 12, 13, 14]. Sometimes, constructing a pipeline is not easy mainly due to input/output formats or any other program peculiarities. This has been considered in SeqTrim, since it transparently accepts several kinds of data formats, and outputs different formats too. Input sequences are accepted as follows: (i) one or more fasta file(s), each one containing one or more sequences, (ii) a fasta file and the corresponding quality value file in fasta format, (iii) a phd format file, (iv) a set of chromatograms, and (v) a zip file containing a set of chromatograms. Outputs are generic fasta files, or fasta files compatible with phred/phrap workflows. Hence, SeqTrim can be included in a workflow to remove low quality sequences, cloning vector, and any other special feature (like poly A or poly T tails if present, restriction sites used for cloning, terminal transferase tails if present, sequence adaptors, etc.) to obtain a clean insert sequence.

Implementation

The algorithm underlying SeqTrim was programmed in Perl 5.6 using BioPerl libraries, and can be executed as a command line tool (ask for code to `jfalgueras@uma.es`) or as a web tool (`http://castanea.ac.uma.es/genuma/seqtrim`). SeqTrim makes use of the external programs phred [4, 5] and blast [6] that must be installed in the same machine that SeqTrim. It will work in any unix/linux release, including OSX, but not on Windows. SeqTrim uses a set of files that are gathered in the '`/usr/local/seqtrim`' directory. In it, the file `seqtrim.conf` contains all configurable default parameters for analysis. The user can also create a copy into the home directory (`~/seqtrim`) to modify it permanently according to the needs. These parameters can also be changed for a single run via command-line options or the web interface (Fig. 1). The '`/usr/local/seqtrim`' directory also contains an editable file called '`RE_sites.txt`' that contains the restriction sites that are

recognized by SeqTrim, and a third editable file named '`adaptorSeqs.txt`' which contains a list of adaptor sequences that will appear in the popup menu for adaptors in the web interface. The content of every file in the `~/seqtrim` directory can be edited by the user (for local customizations) or the system administrator can modify the `/usr/local/seqtrim` files for permanent changes.

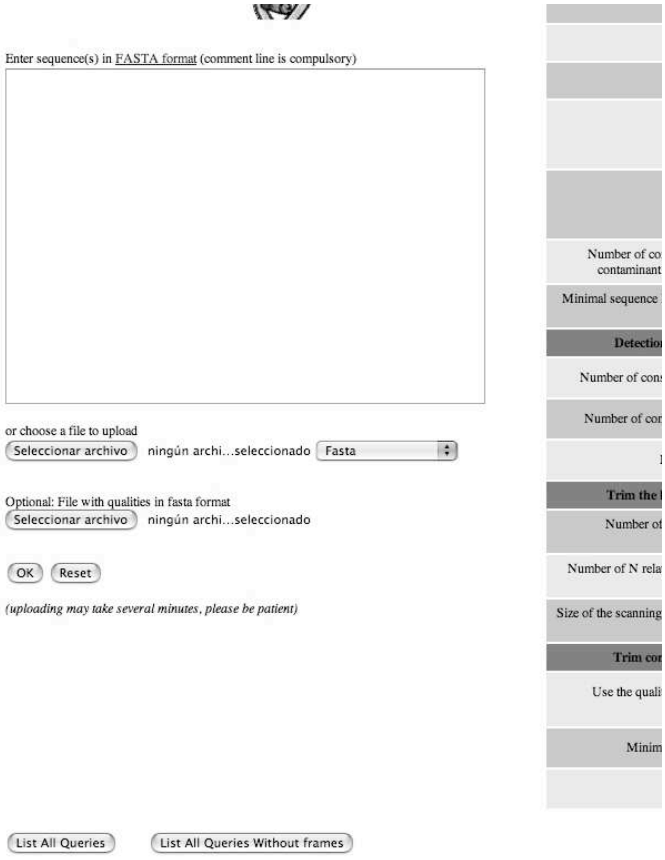

Fig. 1. Web interface of SeqTrim that shows the default parameters

SeqTrim was designed to provide a final trimmed sequence without any unsuitable nucleotides throughout a series of main steps (Fig. 2). A sequence read can contain bases of very low quality, which can hinder detection of further trimming features. Hence the first trimming step will take the base-call quality assessment of each base into consideration (and/or the presence of N as nucleotide indetermination) to trim the original sequence in order to obtain the largest sequences with sufficiently high quality. If the input sequences are chromatograms, the external program phred is called automatically and in a transparently way in order to obtain the quality value for each individual base of the raw sequences and then it calculates the largest sequence with the highest quality. If the input sequences are in a text fasta format, SeqTrim

contains a function to extract the largest subsequence without too many N (indetermination). Any of them provide a "first trim sequence" which corresponds to the reliable sequence obtained from a read. Parameters for determining the sequence quality can be changed.

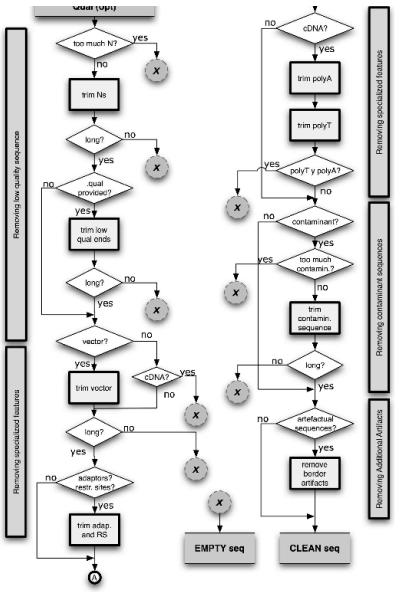

Fig. 2. Detailed data-flow diagram of steps involved in SeqTrim while processing a sequence

The second step relates to cloning vector detection comparing the NCBI's UniVec and the EMBL's emvec vector/adaptor libraries against each "first trim sequence" using blast so that users do not need to specify the cloning vector used for their experiment. Finding vector with blast seems to be better than locating it as a function of restriction site presence since this targets are subject of read or experimental error, even in high quality sequences, and they are very short to assure a good prediction when they are searched with error tolerance. However, since users are asked for the upstream and downstream restriction sites even if both are the same, SeqTrim tries to locate the upstream and downstream restriction sites without ambiguity as a reference to discard the cloning vector when no vector sequence was identified in the database.

The next step is to locate special features that are common to all sequences as they (i) provide false sequences, (ii) mislead assembling or clustering algorithms that can be further used with these sequences, (iii) mislead researchers that use these contaminated sequences. The special features to locate are adaptors, poly A tails (only for ESTs) and poly T tails (only for ESTs, which indicates that the sequence is in the reverse orientation). Due to the procedures used in cDNA library construction, the length of the poly A or T tail should contain at least 20 successive identical nucleotides unless they are at the end of the sequence. To manage such cases, more than one A at the 3' end of a sequence are always removed.

At this moment, the trimmed sequence can be thought to be correct, but there are a lot of possible experimental contaminations that can make several sequences to be useless. Contamination of the input sequences can come from many sources, such as Escherichia coli genomic DNA, the cloning vector, cell plasmids, organelle DNA, viruses, yeast (for tissue culture), human (due to handling), etc. Detection of such contaminants has been achieved launching blast with the trimmed sequences against a database – built from fasta sequence files – that contains sequences than can be considered potential contamination. SeqTrim is provided with the genomes of E. coli, S. cerevisiae, lambda phage and several mitochondria. The vector database is re-screened again. Length, position and identity of the contaminant sequence is provided for user information. Finally, the sequence ends are regarded for presence of any N or artefactual terminal transferase adds.

Rejection criteria for the output sequences include insufficient length (less than 100 bp by default, but this is also modifiable by users), low quality sequence (or excessive amount of Ns) throughout the sequence, absence of insert between identified cloning vector boundaries, detection of two inserts (only in EST) by the presence of two poly A or poly T tails or the concomitant presence of poly A and poly T tails, or considering the complete insert as a contaminant sequence. Warnings are given when cloning vector is not found at the 5' end.

The command line output is a fasta file with the trimmed part from original sequences. Users can determine if the output file will contain or not user readable information for each sequence concerning the trimming events. A coloured output of each sequence can also be seen on the screen, a direct customizing interface being obtained with the SeqTrim web server (Fig. 3), where users can call current and previous executions, save the results according to what is shown on the screen (trimmed sequences, discarded sequences or both), and see the detailed information about each sequence. On the other hand, command line is more adequate for automatic batch processing or workflows. Instead of trimming the original sequence, SeqTrim can mask the unwanted sequences in order to make the output compatible with other programs like the sequence assemblers cap3 [7] or phrap.

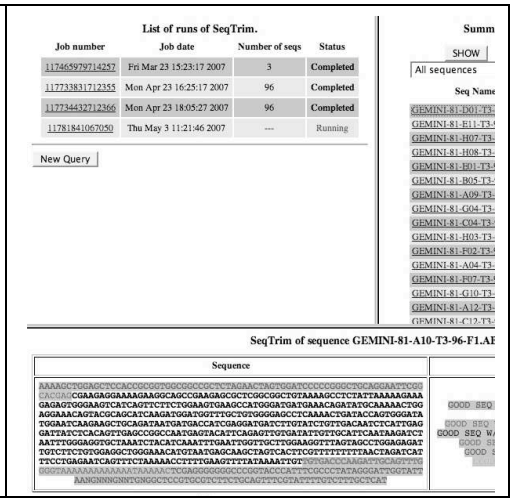

Fig. 3. Example of sequence trimming using different sets of sequences. The upper-left "List of runs" frame displays the status of all executions of SeqTrim made by the user ordered by date and time of run. The upper-right "Summary" frame displays the list of the sequences analyzed in a single session with the original length, and the positions and length of the trimmed part. The bottom frame shows the original sequence coloured according to the different parts that have been removed. Legend on the right explains what colours mean.

Discussion

SeqTrim is well suited for the need of large-scale sequence processing. The flexibility regarding input and output formats contrasts with other sequence preprocessors that admit only one single type of sequence file: SeqClean or ESTPrep accept fasta sequences while Lucy or pregap4 accept chromatograms. Concerning the output, the possibility of saving the final sequences as trimmed sequences or as masked sequences enables the possibility to easily include SeqTrim in other previous workflows like pred/crossmatch/repeatmasker/phrap, EGAssembler [12], or EST2UNI `http://www.melogen.upv.es/genomica/web_estpipe/`). With respect to parameter modification, SeqTrim offers more customization than SeqClean or ESTPrep, nearly pregap4, but do not present the more than forty parameters that can be modified in Lucy [8]. Databases can also be modified directly by the user, simply adding sequences in fasta format, or removing files from appropriate folders in the `/usr/local/seqtrim/DB` directory. Before each execution, SeqTrim verifies if there are changes in the databases and notifies it to the user, reformatting the database index when needed. Concerning the interface, other sequence preprocessors were created only as command line programs (SeqClean, ESTPrep, TrimSeq, phred/crossmatch), only as web pages (VecScreen, EMBVec Query) or as command line and a GUI interface (Lucy, pregap4). SeqTrim is the first one that provides a command line and web interface, although the creation of EGAssembler has recently provided an additional web interface to SeqClean [8]. Like Lucy, SeqTrim uses colors for the output, but in contrast to the rest of sequence preprocessor programs, SeqTrim colors trimmed regions in each sequence, the clean sequence is shown as black text (Fig. 3). This facilitates result interpretation as well as comparison between cleaned and original sequences since they are on the same string instead of two synchronized scrolling panels as occurs in Lucy. The 'Info' cell containing the legend explaining colors also deploys the trimmed range that passed each trimming step.

SeqTrim is under continuous refinement due to collaboration between biologists and computer scientists that have succeed in dealing with most sequence cases and open the possibility to include new capabilities to manage new unexpected sequence behaviors. At present, several changes are planned, like enabling users to select the contaminant databases to use (instead using all of them like now), or to establish a different order of action than described.

Acknowledgements

This work is supported bye the Spanish MEC grants AGL-2006-07360/FOR and BIO2006-06216 as well as the Junta de Andalucía grant AGR-663 and foundings to the research groups CVI-114, TIC-160, and TIC-113.

References

1. Coker JS, Davies E (2004) Identifying adaptor contamination when mining DNA sequence data. Biotechniques 37, 194, 196, 198
2. Chou HH, Holmes MH (2001) DNA sequence quality trimming and vector removal. Bioinformatics 17:1093–1104
3. Bonfield JK, Smith K, Staden R (1995) A new DNA sequence assembly program. Nucleic Acids Res 23:4992–4999
4. Ewing B, Green P (1998) Base-calling of automated sequencer traces using phred. II. Error probabilities. Genome Res 8:186–194
5. Ewing B, Hillier L, Wendl MC, Green P (1998) Base-calling of automated sequencer traces using phred. I. Accuracy assessment. Genome Res 8:175–185
6. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. J Mol Biol 215:403–410
7. Huang X, Madan A (1999) CAP3: A DNA sequence assembly program. Genome research 9:868–877
8. Li S, Chou HH (2004) LUCY2: an interactive DNA sequence quality trimming and vector removal tool. Bioinformatics 20:2865–2866
9. Gordon D, Abajian C, Green P (1998) Consed: a graphical tool for sequence finishing. Genome Res 8:195–202
10. Cantón F, Le Provost G, García V, Barré A, Frigerio JM, Paiva J, Fevereiro P, Ávila C, Mouret JF, de Daruvar A, Cánovas F, Plomion C (2003) Transcriptome analysis of wood formation in maritime pine. In Sustainable Forestry, Wood products and Biotechnology, S Espinel, Y Barredo, E Ritter, eds (Vitoria-Gasteiz: DFA-AFA Press)
11. Liang F, Holt I, Pertea G, Karamycheva S, Salzberg S, Quackenbush J (2000) An optimized protocol for analysis of EST sequences. Nucleic acids research 28:3657–3665
12. Masoudi-Nejad A, Tonomura K, Kawashima S, Moriya Y, Suzuki M, Itoh M, Kanehisa M, Endo T, Goto S (2006) EGassembler: online bioinformatics service for large-scale processing, clustering and assembling ESTs and genomic DNA fragments. Nucleic Acids Res 34:W459–462
13. Miller RT, Christoffels AG, Gopalakrishnan C, Burke J, Ptitsyn AA, Broveak TR, Hide WA (1999) A comprehensive approach to clustering of expressed human gene sequence: the sequence tag alignment and consensus knowledge base. Gemone Res 9:1143–1155
14. Scheetz TE, Trivedi N, Roberts CA, Kucaba T, Berger B, Robinson NL, Birkett CL, Gavin AJ, O'Leary B, Braun TA, Bonaldo MF, Robinson JP, Sheffield VC, Soares MB, Casavant TL (2003) ESTprep: preprocessing cDNA sequence reads. Bioinformatics 19:1318–1324