SPECTRUM PUBLISHED: June, 2018

Auto Sequencer: A DNA Sequence Alignment and Assembly Tool

Abhi Aggarwal, Landon Zarowny, Robert E. Campbell

Department of Chemistry, University of Alberta, Edmonton, Alberta
Corresponding author: abhi2@ualberta.ca

ABSTRACT

The process of determining the exact order of nucleotides in DNA is a crucial component of a wide variety of research applications known as DNA sequencing. Over the last fifty years, several DNA sequencing technologies have been well characterized through their nature and the kind of output they provide. Even with significant advances in DNA sequencing technology, sequencing and assembly of large pieces of DNA remains a complex task. It requires sequencing small reads of DNA at a time, and performing DNA sequence assembly to merge the individual pieces into a single contiguous sequence. DNA sequence assembly, albeit tedious and time consuming, is a process in which short DNA sequence fragments are merged into longer fragments in the attempt to reconstruct the original DNA sequence. This is usually achieved by manually identifying sequence overlaps between two reads before aligning them into one contiguous sequence. Then, with the aid of online tools or software, this contiguous sequence is translated into protein sequence. While this process may only take a few minutes, the complexity of sequence translation and assembly can be driven by two major challenges: finding the most reasonable overlap in sequences that may contain repeats or low quality regions, and outputting both nucleotide and protein sequence in an easy to use, comprehensive output. To facilitate this process, we introduce an all-in-one tool: Auto Sequencer. This user-friendly tool can combine and translate raw DNA sequence files by finding the most reasonable overlap between them displaying outputs in flexible formats.

Background

Deoxyribonucleic acid (DNA) is the hereditary material found inside of all cells. It is made up of four chemical building blocks called nucleotides: adenine (A), thymine (T), guanine (G), and cytosine (C). These nucleotide bases on one strand of DNA form base pairs through hydrogen bonds with a second strand of DNA to form the familiar double helix structure[1] of DNA. However, the combinations of base pairs that can be formed are strictly limited: adenine only forms

Citation: Aggarwal, A; Zarowny, L; Campbell, R.E. (2018) Auto Sequencer: A DNA Sequence Alignment and Assembly Tool. *Spectrum* Issue No. 1

doi: 10.29173/spectrum35

Received: Nov. 30, 2017
Accepted: Feb. 25, 2018
Published: June 1, 2018

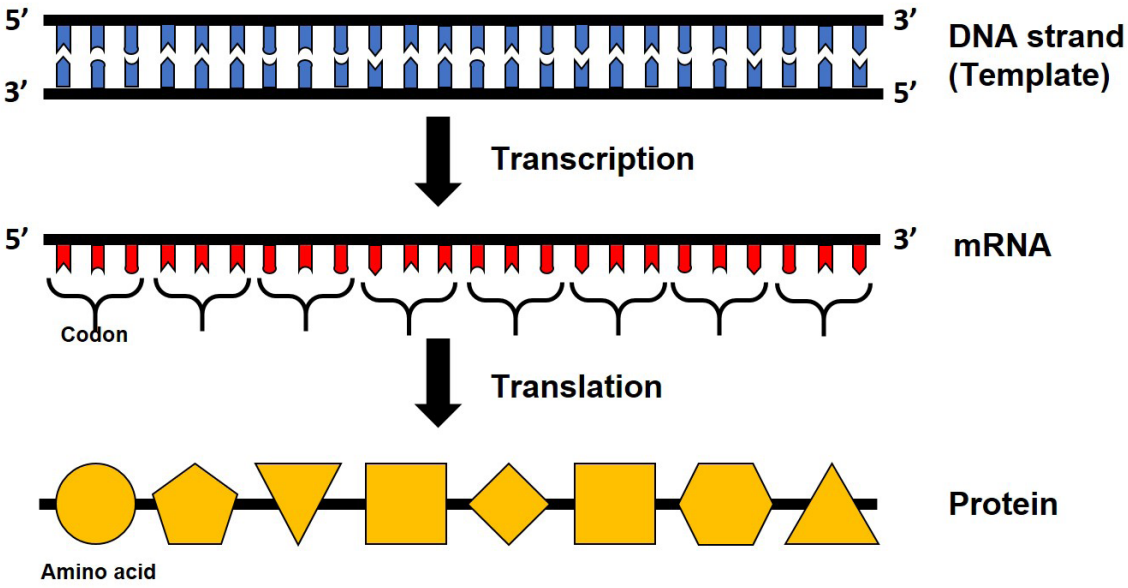

Figure 1. An overview of the two stages of protein production: Transcription and Translation. DNA sequence information is converted to mRNA in a process called transcription. Each group of three bases in mRNA constitutes a codon, and each codon specifies a particular amino acid. The mRNA sequence is thus used as a template to assemble the chain of amino acids that form a protein, via a process called translation.

a base pair with thymine and guanine only forms a base pair with cytosine. These strands of DNA run in opposite directions: the starting end (5') of one strand is paired with the tail end (3') of the second strand (Figure 1). Accordingly, knowing the sequence of bases on one strand of DNA can allow us to determine the sequence of bases on the other strand of DNA. The order of these bases, or the sequence of DNA, determines what biological instructions are contained in a strand of DNA. For example, a certain strand of DNA may contain the instructions that an organism needs to develop, survive and reproduce. To carry out these functions, DNA sequences are transcribed into mRNA, which are then translated into amino acids, the building blocks of proteins. These proteins are the complex molecules that do most of the work in our bodies. Ultimately, the precise order of DNA sequence links to a specific order of amino acids, which encodes for a specific protein. This flow of genetic information from DNA to RNA to protein is known as The Central Dogma of Biology[2] (Figure 1). Consequently, getting the information about the order of bases in the DNA sequence is important in order to understand which protein that DNA encodes for and what function it may have in the cell. To read the DNA sequence, two-time Nobel Laureate Frederic Sanger created an ingenious method of reading a DNA molecule, commonly known as Sanger Sequencing[3]. In this method of sequencing, the DNA of interest serves as a template for DNA synthesis. A DNA primer, a short strand of DNA, binds to the 3' of the template DNA to be read. The DNA primer serves as a starting point for DNA synthesis by a DNA polymerase. Fluorescently labelled nucleotide analogues, differing in their chemical structure from normal nucleotides, terminate strand synthesis at many different positions in the reaction. This technique produces DNA fragments with fluorescently labelled nucleotides that terminate at specific bases. Separation of the fragments using electrophoresis produces specific sizes that can then be used to determine the nucleotide at each position. Sequences produced using the Sanger method can then be aligned using overlapping segments to produce ever larger full sequences from genes to chromosomes.

doi: 10.29173/spectrum35

Introduction

The process of determining the sequence of nucleotides in DNA is a fundamental biological technique. As such, many DNA sequencing technologies[4] have been developed, but the process itself is a seemingly complex one. Since these technologies have a limit to how many bases of a DNA sequence that can be sequenced in one experiment, larger DNA sequences must be sequenced as smaller sections. This resulting sequence information must then be digitally combined to provide the overall sequence information. For example, Sanger sequencing produces approximately 1000 base pairs of sequencing information per sequencing reaction[5]. To sequence a gene longer than 1000 base pairs, researchers often perform two sequencing reactions: one at the front end of the gene in the forward direction, and the other at the opposite end of the gene in the reverse direction (Figure 2). To get the complete nucleotide (and ultimately protein) sequence, these forward and reverse reads must be assembled into one complete sequence by finding the overlap in between. Depending on the size of the genome and the sequencing technology used, this process can be very tedious, time consuming, and repetitive.

To facilitate the combining of forward and reverse sequencing information, we have developed the Auto Sequencer software. Auto Sequencer is an open-source, free-to-use tool, that can align two DNA sequences by finding the overlap between different fragments automatically, with consistency and high accuracy. It contains a user-friendly graphical interface that allows the user to choose their output display in three different formats: nucleotide sequence, protein sequence, or both. Auto Sequencer aims to increase productivity and efficiency by saving time, reducing human error, and allowing users with little experience to accomplish their sequencing needs. Because Sanger sequencing, which is limited to approximately 1000 base pair read per reaction, is still considered the gold standard DNA sequencing technique[6] and is widely available to researchers, the Auto Sequencer software is likely to be of great utility to a wide range of researchers.

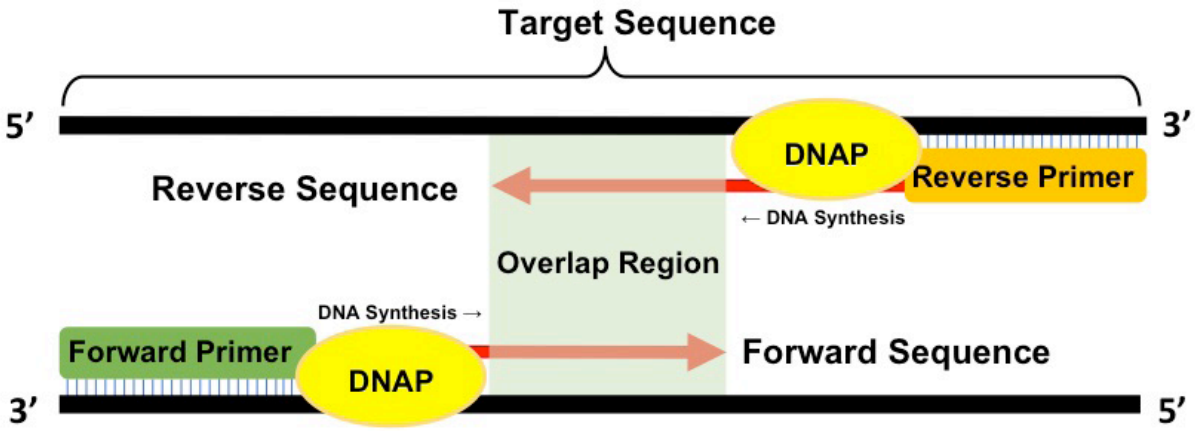

Figure 2. Schematic showing the use of forward and reverse primers. Forward and reverse primers anneal to their respective complementary site at each end of the desired target gene. These primers serve as a starting point for DNA synthesis by the DNA polymerase (DNAP). As DNA synthesis continues from the forward and reverse primers, forward and reverse sequences are obtained, respectively. The overlap region between the forward and reverse sequence is used to assemble the complete DNA sequence.

doi: 10.29173/spectrum35

Procedure

Auto Sequencer uses a graphical user interface (Figure 3). To use Auto Sequencer, the user needs raw nucleotide sequence text file(s) that they wish to translate and assemble. If the user wishes to translate a forward nucleotide sequence only, then they must tick 'Enable Forward Sequence' to activate the text box underneath, while keeping reverse sequence textbox deactivated (Figure 3a). To translate reverse sequence only, 'Enable Reverse Sequence' must be checked, while 'Enable Forward Sequence' is disabled. Note that this procedure will only translate your sequence but not carry out the assembly algorithm. To translate and assemble forward and reverse sequencing files, both checkboxes must be ticked (Figure 3b). Once a selection has been made, the sequencing reads can be dragged-and-dropped or copy-pasted into the activated textboxes, ensuring that the reads are in in 5' 3' direction. Then, the user can proceed to translation and assembly algorithm by clicking the 'Translate' button (Figure 3c).

Clicking the 'Translate' button will open another window containing three forward sequence frames and/or three reverse sequence frames, depending on the sequence reads used (Figure 4). To find out which frame(s) will undergo translation, the frame containing the longest stretch of codons before arriving a stop codon is identified; this frame that contains the largest open reading frame (ORF) will be bolded and used for the assembly process (Figure 4a,b). If only one sequence read is being used for translation, then this sequence read will be displayed in the main window. If both forward and reverse sequence reads are used, the bolded frames (containing the largest ORF), will be assembled into one sequence by finding the largest overlap between the two reads. If more than one overlap is found, the priority will be given to the larger match. The size of this overlapping amino acid sequence will be displayed in the bottom left corner and can be used to infer the quality of the assembly process (Figure 4c). In this window, the user also has the choice to display these frames as translated proteins or raw nucleotides by choosing the appropriate radio button at the bottom (Figure 4d). At this point, the user may close the frames window and navigate back to the sequence input window to access their translated and assembled sequence. Note that by closing the frames window, the user will not be able to access the individual frames again. To save the individual frames for

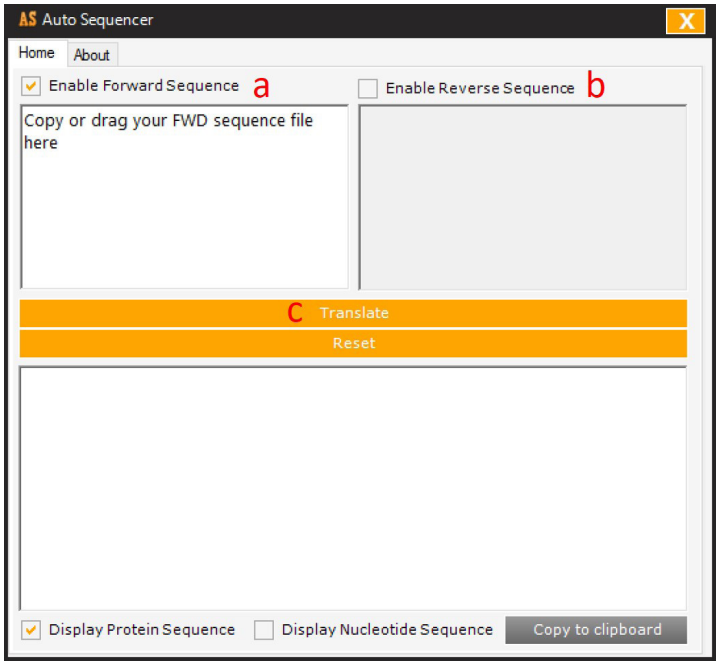

Figure 3. Graphical User Interface. Photo of the main screen of the program. a) To translate a forward sequence read, this checkbox must be ticked to activate the drop box for the forward sequence. b) If translating a reverse sequence read, then this checkbox must be ticked to activate its respective drop box. Both checkboxes must be ticked to translate and assemble a forward and reverse sequence read. c) Once sequence reads have been dropped in or copy/pasted, user can proceed to translation and assembly algorithm by clicking the translate button.

doi: 10.29173/spectrum35

future comparison, the user can click the right-angled arrow in the top right corner: this will save all six frames in the program's memory and allow for direct, manual comparison even when a different sequence is being manipulated (Figure 4e). In the main user interface, where the translated and assembled sequence will appear, users have several options to display the sequence to fit their needs (Figure 5). Users can check 'Display Nucleotide Sequence' to display the untranslated, assembled sequence (Figure 5i); 'Display Protein Sequence' to display the translated, assembled sequence (Figure 5ii); or check both options to display translated and untranslated sequence on

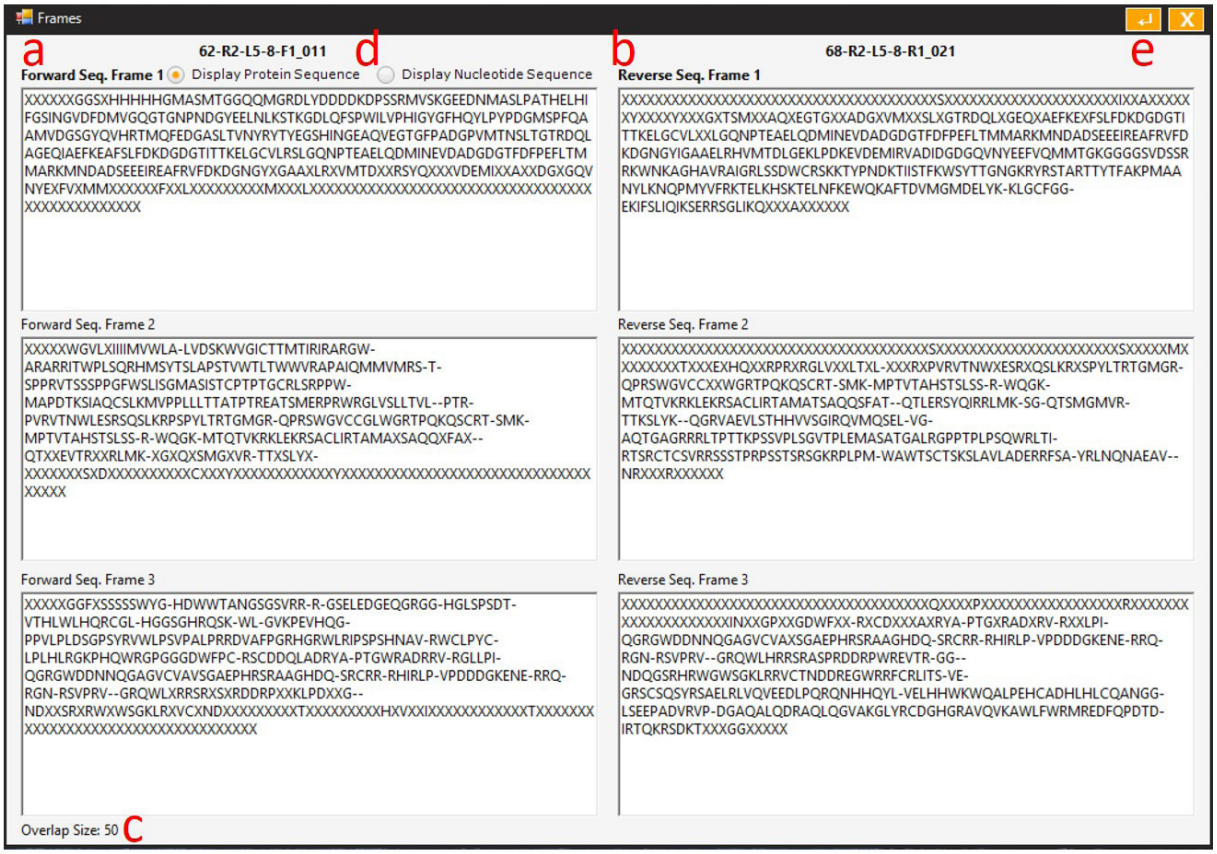

Figure 4. Example of the frames window using a sample forward and reverse sequence reads. a) The forward sequence frame containing the largest open reading frame (ORF) is bolded. b) The reverse sequence frame containing the largest ORF is bolded. c) The size of the maximum overlap of amino acids between two frames is indicated for user's discretion. d) The user can view the sequence frames in protein sequence or nucleotide sequence by choosing the appropriate button. e) Clicking this button will save all individual frames in the program's memory, and allow for a side-by-side comparison with other sequencing frames.

the same window (Figure 5iii). The nucleotide sequences that cannot be translated to amino acids are denoted in red, bolded "X" to allow easy corrections. Once the resulting sequence is displayed as desired, users can click 'Copy to Clipboard', making it available to paste the results into other applications while keeping the format of the text intact (Figure 5b). At this point, users can close the window or click the 'Reset' button to continue the translation and assembly of other sequencing files (Figure 5c).

the same window (Figure 5iii). The nucleotide sequences that cannot be translated to amino acids are denoted in red, bolded "X" to allow easy corrections. Once the resulting sequence is displayed as desired, users can click 'Copy to Clipboard', making it available to paste the results into other applications while keeping

doi: 10.29173/spectrum35

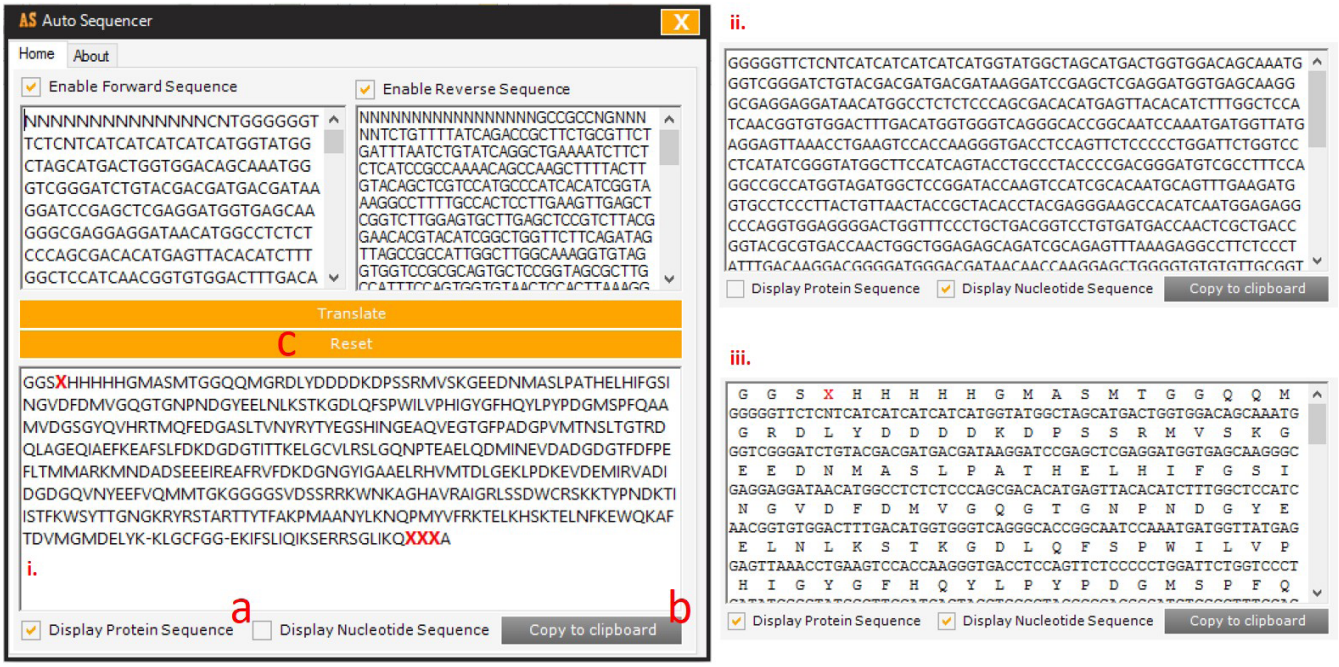

Figure 5. Example of the main user interface containing outputs from the translated and assembled forward and sequence reads. a) The output can be viewed in 3 different formats. Selecting the protein sequence box only will display the translated, assembled protein sequence (i), selecting nucleotide sequence only will display the untranslated, assembled nucleotide sequence (ii), and selecting both checkboxes will display both translated protein sequence and untranslated nucleotide sequence on the same window (iii). b) Once the desired output format is displayed in the window, users can click 'Copy to clipboard' to easily copy the output to their clipboard, making it available to be pasted into other applications. c) To translate and assemble other sequence reads, users can click 'Reset' to reset the program to default.

the format of the text intact (Figure 5b). At this point, users can close the window or click the 'Reset' button to continue the translation and assembly of other sequencing files (Figure 5c).

Methodology

When the 'Translate' button is clicked, Auto Sequencer goes through a series of algorithms to achieve the final translated, assembled sequence (Figure 6). Firstly, the program detects the input of forward and/or reverse reads by looking at which of the two textboxes are activated using their respective checkboxes. If forward sequence is present, it is assigned to a 'forward' variable as is; the reverse sequence is taken as a reverse complement before it is assigned to its specific 'reverse' variable. Then, each sequence read is split into 3-letter codons starting at the first, second, and third letter of the sequence, generating three unique combinations of the nucleotide sequences. Next, using a standard DNA codon table, each combination of codons is translated into protein sequence, resulting in three potential ORFs (Open Reading Frame) encoded within each of the three translation frames for each forward and reverse sequence.

To determine which sequence frame codes for a protein, the program's algorithm looks for the frame that gives the longest amino acid sequence before a stop codon is encountered. Longest ORFs are often used to identify candidate protein-coding regions in a DNA sequence. Since there are 64 codons and three of these code for stop codons, we would expect a stop codon to appear on average every 21 amino acids in an incorrect frame[7]. Consequently, the two incorrect frames will contain a higher number of stop codons when compared to the correct frame, containing the

doi: 10.29173/spectrum35

lowest number of stop codons. The algorithm counts the number of stop codons in each frame, denoted by a dash, and bolds the frame with the lowest number of stop codons. If the second sequence read is also inputted by the user, then the process is repeated and the two chosen frames are used for the assembly process.

The assembly process, where the program attempts to align and merge the amino acid fragments from the correct forward and reverse frames involves taking the two frames, looking for areas in which they overlap with each other and then gradually piecing together the pieces. Auto Sequencer uses a heuristic approach that merges the two frames together at the point of maximum overlap (Figure 7). Since there are 20 amino acids but only 4 nucleotide bases, running this algorithm

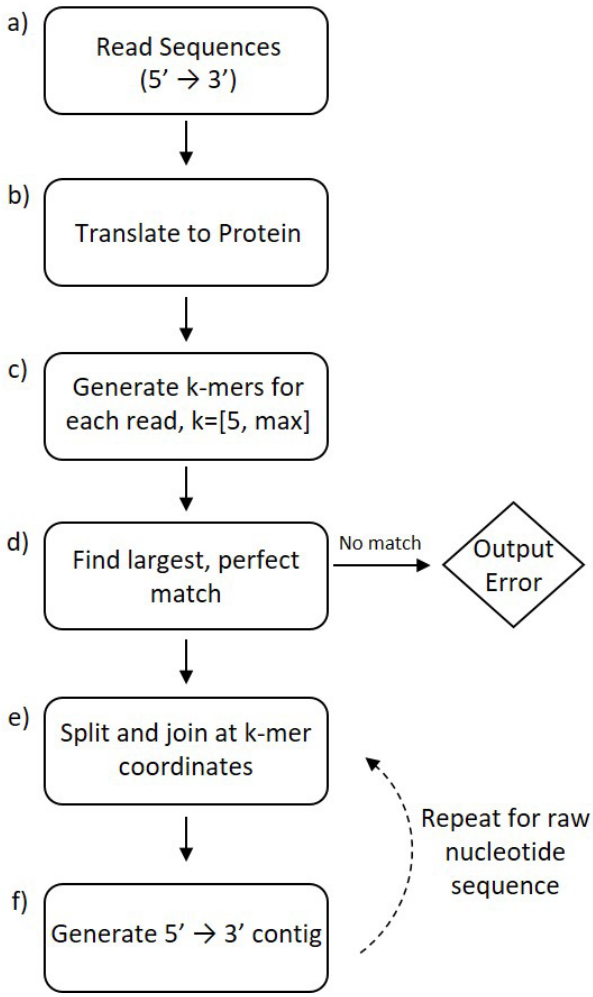

Figure 6. Pseudocode for Auto Sequencer algorithm. a) Sequences are read in 5' 3' direction; reverse complement of the reverse sequence is taken before it is read. b) Sequences are translated to protein sequence using standard DNA codon table. c) For each input read of length l, k-mers of length [5, l] are generated by sliding a window of length k over the read. d) The largest perfect match is found between k-mers generated from the forward sequence and k-mers generated from the reverse sequence. This is used as the 'overlap point'; if no match is found, it indicates low quality sequence or lack of overlap region between the two reads and an error message is outputted. e) Forward sequence is split at coordinate [l-k]; k = length of match, reverse sequence is split at coordinate [k]. The split forward sequence is aligned with split reverse sequence to generate a final sequence. f) The process is repeated for nucleotide sequence where l*3 and k*3 positions are used to align the two sequences.

doi: 10.29173/spectrum35

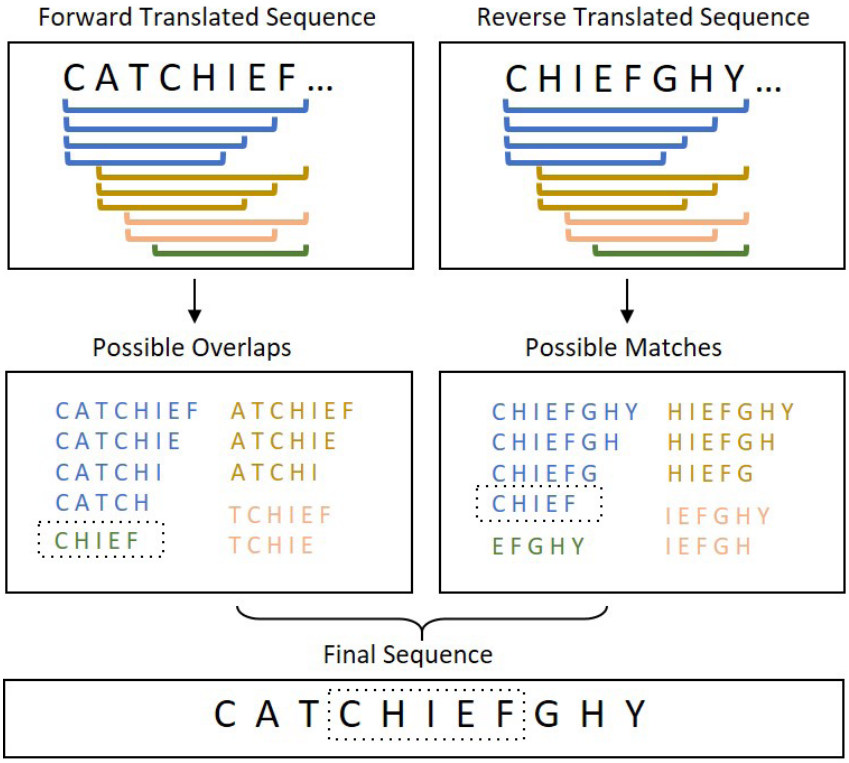

Figure 7. Example showing the assembly of final sequence from forward and reverse protein sequences using overlap sequence algorithm. A database of possible overlaps and matches is created by splitting the forward translated sequence and reverse translated sequence into k-mers, where the size of k = [5,max]. Once a match has been found, coordinates of the matched sequence are used to clump the sequences into larger alignments. If >1 matches are found, priority is given to the largest match.

at the protein level should minimize the chances of finding overlaps due to chance[8]. We also expect this process to be relatively quicker as it will require a search for a smaller sized amino acid overlap in comparison to a nucleotide sequence overlap.

To efficiently find this maximum overlap point, the algorithm begins by trimming off low quality sequences on the starting end of the forward frame and the ending of the reverse frame, denoted by 'X'. Then, using the trimmed forward frame, a database of potential overlap points, k-mers, are generated by creating combinations of consecutive strings, ranging from a minimum size of five amino acids to a maximum size of the frame. This is achieved by looping from the size of the frame, down to five, and extending the overlap substring until the entire frame has been covered. The minimum limit, five amino acids, has been chosen to ensure a reasonable overlap, such that a match is not found due to chance[9]. Once this database is generated, each k-mer is cross-referenced with the reverse frame, starting from the largest k-mer to the smallest, for a perfect match. Upon match, the search is stopped and the size of the current match (maximum sized overlap) is denoted in the bottom left corner of the frames window for user's discretion. The starting and ending position, along with the size of this overlap point, is stored in integer variables to be used later in the assembly of nucleotide sequences. Finally, the overlap substring is removed from the forward frame before the two frames are merged together to generate the assembled protein sequence. The assembled protein sequence is stored in a new variable to be used later in the display window. To replicate the assembly for the raw nucleotide sequence, the algorithm begins by trimming off

doi: 10.29173/spectrum35

an equivalent number of codons from the ends of the two frames. For example, if initially 10 low quality amino acids were trimmed from the start of the protein sequence frame, then 30 DNA bases (3 bases per codon) are trimmed from the start of the nucleotide sequence. Next, using the stored variables from above, the positions and size of the overlap portions are also manipulated to represent their true position in the nucleotide sequence. Using these positions, the overlap portion is removed from the forward sequence frame and the resulting frames are merged together to assemble the nucleotide sequence. The assembled nucleotide sequence is stored in a new variable to be used later in the display window.

In the display window, the user can alternate between the nucleotide sequence, protein sequence or display both at the same time. When the nucleotide sequence checkbox is ticked, the assembled nucleotide sequence is displayed in the window and all the un-sequenced regions, denoted by 'N' are highlighted in red. When the protein sequence checkbox is ticked, the assembled protein sequence is displayed in the window and all the un-translated regions, denoted by 'X' are highlighted in red. When both nucleotide and protein sequence checkboxes are ticked, both sequences are displayed on alternating lines. To ensure that the format stays consistent (three nucleotides for every amino acid), a space has been added before and after each amino acid. Furthermore, a monospaced font[10], Courier New, has been used to ensure that the width of each character stays consistent and does not alter the spacing between nucleotide codons and its respective amino acid.

Discussion

Auto Sequencer has several advantages to available online tools. It is a straightforward program that has been optimized to do one job effectively and efficiently. The tool is a stand-alone application, very small in size (800 Kb), that can sit on your desktop and can be used without access to the internet. All users need to operate the tool are their DNA sequence files. Auto Sequencer's interface allows the flexibility of sequence files to be in any format and can be drag and dropped or copy-pasted into their respective boxes. There is also no limitation to the size of the sequences, given that the computer's random-access memory (RAM) can uphold the algorithm. The tool will automatically translate the sequences, find the correct sequencing frames, determine the best overlap point between each frame and output the result for viewing.

Just like any other assembly tool, Auto Sequencer's algorithm also contains some limitations that must be addressed. The first limitation is a sequence read with long repeats that can prevent the program from detecting the maximum sized overlap between two frames. Any sequence assembler cannot overcome this limitation solely based on the information contained in the sequence reads – some additional information about the sequence or the repeats must be provided, otherwise manual human labor will be required. For example, as seen in Figure 8, a repetitive element Hn consisting of six amino acids is repeated throughout the forward and reverse sequence. Given that the overlap between the two frames is only five amino acids long, relatively smaller to the repetitive element, no assembler can truly distinguish between the repeat and the overlap without additional information. In this case, the resulting assembly will contain the region of the repetitive element that will align the forward and reverse sequence reads together, resulting in an incorrect assembly. To address this issue, assembly tools must be able to identify repeats and low quality sequences prior to the assembly process, to avoid incorrect genome reconstructions due to over collapsing repeating elements. Then, the tool needs to remove the repeats from its possible overlap database before continuing the assembly process. However, detecting such repeats can be difficult and often requires additional information provided by the user.

doi: 10.29173/spectrum35

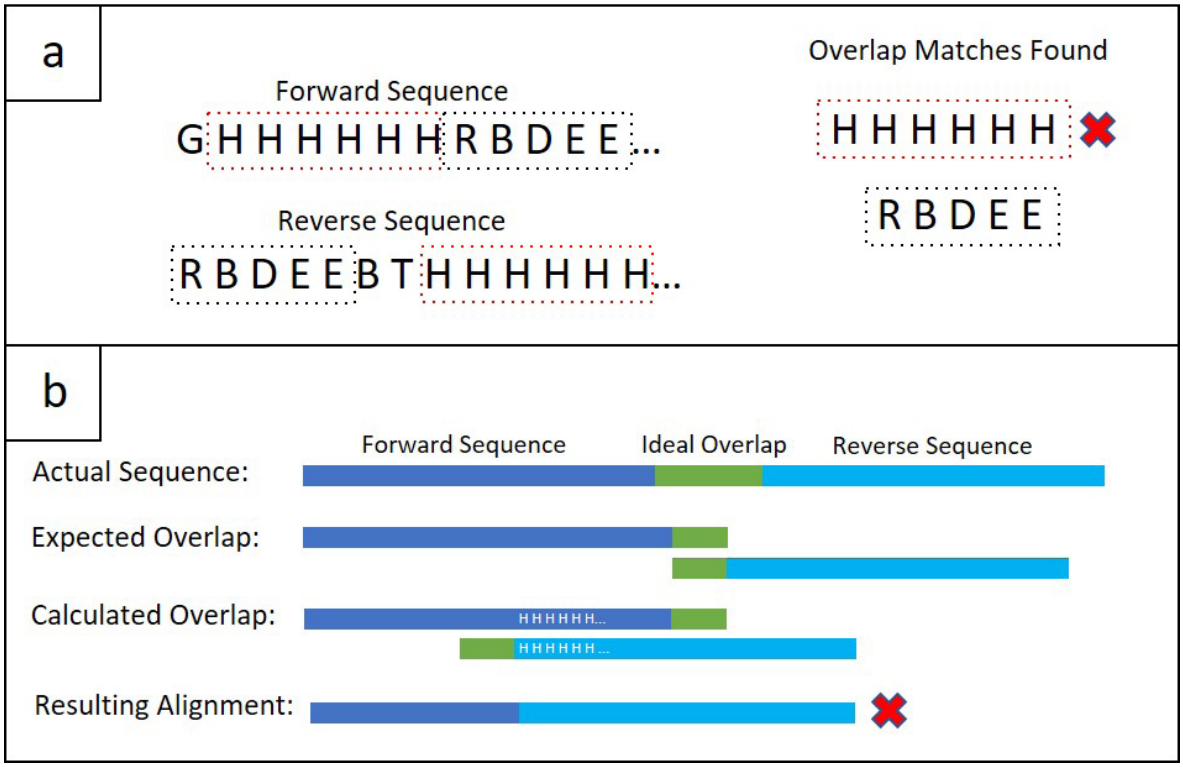

Figure 8: Example of an incorrect assembly due to repeats. a) Low quality sequence or repetitive elements (H^n in this case) common throughout the forward and reverse protein sequence lead to larger overlap matches and aligns the two sequences at the wrong position. The first potential overlap match found is 'RBDEE' which is overridden by a bigger match found after: 'HHHHHH'. This leads to splitting at the end of Hn sequence, as oppose to RBDEE, which results in an incorrect alignment. b) Visual representation of incorrect alignment. The forward and reverse sequences are aligned together with the green overlap region. Since there is a low quality sequence or duplicated regions, the calculated overlap length exceeds the green overlap, and is chosen to be the overlap point for splitting and aligning the two sequences. This causes a missassembly despite the fact that there was a correct overlap match found.

Further complicating assembly, the presence of gaps, missense mutations and nonsense mutations in the overlap region will fail the algorithm in detecting the overlap between both reads. The failure to identify the overlap will disallow the program from aligning the two frames together, resulting in an error. Complex mutations or gaps in the frame could also decrease the size of the ORF, failing the algorithm in recognizing the correct, protein-coding frame. These examples highlight several issues that Auto Sequencer cannot address without additional information. The resolution of these problems entails an additional assembly phase, involving a large amount of human intervention. In the future versions of Auto Sequencer, we will attempt to overcome this problem by allowing users to enter in a reference sequence that can be used, as an example, to align the two frames together with much more accuracy.

Conclusion

The goal of sequencing a piece of DNA is the complete nucleotide and protein sequence constructed into one read. Since many sequencing technologies are unable to read an entire genome in one go, a sequencing translation and assembly tool can reduce the human labour involved in the final step of DNA sequencing. Auto Sequencer has been created to automatically translate sequencing reads, two at a time, and assemble them together by finding the longest common overlap between both

doi: 10.29173/spectrum35

reads. It can efficiently translate and assemble reads together, even in the presence of minimal sequencing errors. DNA sequence assembly, especially when you have multiple sets, can be very laborious and time consuming. By introducing this program to researchers and scientists, we hope to dramatically save their valuable time, increase efficiency by reducing human errors, and decrease the need for human intervention.

Acknowledgments

We thank Dr. Yi Shen and Dr. Matthew Wiens for insight and expertise that greatly assisted the development of Auto Sequencer. We would also like to show our gratitude to our colleagues from Campbell Lab who tested the tool and reported any bugs, suggestions and ways of improvement. We are also immensely grateful to Undergraduate Research Initiative (URI) for providing funding and encouraging undergraduate research projects.

Download

Auto Sequencer can be downloaded from Campbell Lab website at: http://campbellweb.chem.ualberta.ca/links/auto-sequencer/

doi: 10.29173/spectrum35

References

1. Watson JD, Crick FH. 1953. Molecular structure of nucleic Acids: a structure for deoxyribose nucleic acid. Nature. 171(4356):737-738. http://dx.doi.org/10.1038/171737a0.
2. Crick FH. 1970. Central dogma of molecular biology. Nature. 227(5258):561-563.
3. Sanger F, Nicklen S, Coulson AR. 1977. DNA sequencing with chain-terminating inhibitors. Proc Natl Acad Sci USA. 74(12):5463-7.
4. Pareek CS, Smoczynski R, Tretyn A. 2011. Sequencing technologies and genome sequencing. J Appl Genet. 52(4):413-35. http://dx.doi.org/10.1007/s13353-011-0057-x.
5. Karger BL, Guttman A. 2009. DNA sequencing by CE. Electrophoresis. 30(S1):196-202. http://dx.doi.org/10.1002/elps.200900218.
6. Perrier L, Heinz D, Baffert S, Zou Z, Durand ZI, Rouleau E, Wang Q, Haddad V, Bringuier P, Merlio J, et al. 2015. Cost of genome analysis: the sanger sequencing method. Value Health. 18(7):A353. http://dx.doi.org/10.1016/j.jval.2015.09.654.
7. Deonier RC, Tavaré S, Waterman M. 2005. Computational genome analysis. Springer-Verlag New York. http://dx.doi.org/10.1007/0-387-28807-4.
8. Pearson WR. 2013. An introduction to sequence similarity ("homology") searching. Current Protocols in Bioinformatics. 3(10). http://dx.doi.org/10.1002/0471250953.bi0301s42.
9. Kent WJ. 2002. BLAT – the BLAST-like alignment tool. Genome Research. 12(4):656-664. http://dx.doi.org/10.1101/gr.229202.
10. Rosendorf T. 2009. The typographic desk reference. New Castle, DE: Oak Knoll Press.

doi: 10.29173/spectrum35