A Python script to merge Sanger sequences

Cen Chen
Clinical Center for Genomic Research, Union Hospital, Huazhong University of Science and Technology
Xiaofang Huang
Fujian Agriculture and Forestry University
Bingguo Lu
Fujian Normal University
Chuyun Bi
Fujian Agriculture and Forestry University
Lili Zhao
Chinese Center for Disease Control and Prevention
Yunzhuo Hu
Fujian Agriculture and Forestry University
Xuanyang Chen
Fujian Agriculture and Forestry University
Shiqiang Lin (linshiqiang@fafu.edu.cn)
Fujian Agriculture and Forestry University https://orcid.org/0000-0001-7181-4261
Kai Huang
Clinical Center for Human Genomic Research, Union Hospital, Huazhong University of Science and Technology

Software article

Keywords: Sanger sequencing, Merge, Python script

Posted Date: November 18th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-107687/v1

Abstract

Merging Sanger sequences is frequently needed during the gene cloning process. In this study, we provide a Python script that is able to merge multiple overlapping Sanger sequences. Our results demonstrate that the script can produce the merged sequence from the input Sanger sequences in a single run. The script offers a simple and free method for merging Sanger sequences and is useful for gene cloning.

Introduction

Gene cloning is customarily required to study the gene functions in vivo and in vitro. For the most part, the gene of concern is ligated to a vector with a canonical method of restriction enzyme cutting and ligation, or the new-fashioned technique of seamless ligation [1]. To ensure that the gene sequence is correct within the constructed plasmid, the Sanger sequencing is utilized to sequence the gene and the results are aligned and checked [2].

For those genes with lengths of only several hundreds, it is possible to go through each one of the whole sequences with one single Sanger sequencing reaction and the results can be directly aligned to the correct gene files. However, in other cases, the lengths of the genes exceed one thousand, which is beyond the scope of one single Sanger sequencing reaction. In order to get the sequence of the full-length gene, it is necessary to carry out DNA walking sequencing using a new primer based on the previously sequenced region. To sequence a large gene, several reactions might be required in both forward and reverse directions. These results have to be aligned to the correct gene file for confirmation.

It is preferred to merging all the walking results prior to alignment with the correct gene file, rather than aligning each walking result with the target gene file separately. To merge the multiple walking results, there are several commercial software such as the DNASTAR [3], DNAMAN [4] and Vector NTI [5], which are powerful, however, expensive. In terms of free software, there is an on-line tool that is able to merge overlapping long sequence fragments based on the program merger in the EMBOSS suite [6, 7]. However, the web tool relies on web access and server status. Here, we introduce a Python script that is capable of merging multiple overlapping Sanger sequencing files by employing the alignment module of Biopython [8]. To visualize the key parameters of the merging process, we print the output of alignment and show the critical junction within the merged sequence. The script is useful for merging the Sanger sequences and increasing the efficiency of gene cloning.

Materials And Methods

Running environments and input files

To run the script Merge_sanger_v2.py, the macOS Catalina 10.15.3 or higher (Apple Inc., CA, USA) is needed. Python 3.7.3, Biopython 1.7.4, and EMBOSS 6.6.0 are necessary for running the script in this study [7-9]. Here, The example input files (S1, see Supplementary Materials) are the sequencing result

LS, which was amplified from PX458 [10], shown in Fig. 1. All

input files are in the seq format. Our script requires that the first four characters of the file name consist of three numbers (000, 001, 002, 003,…) plus one capital letter F or R (F for forward Sanger sequencing and R for reverse Sanger sequencing). This step is done by the user by adding four characters to the original filename provided by the sequencing company, according to the positions of the Sanger sequencing result, with a smaller number in the upstream and larger number in the downstream along the sense strand of the full-length sequence.

Running script

To run the script, open the macOS terminal, cd to the directory containing the script (S2, see Supplementary Materials) and Sanger sequences and input the following bash command.

1) Make a new directory and copy the sequencing result files of the above section (Running environments and input files) to the newly made directory. Also, copy the script ‘Merge_Sanger_v2.py’ to the directory.

2) Open a new terminal and cd the directory. Run the following command.

python3.7 Merge_Sanger_v2.py 001F_F103749_F714961c_CS13_CS13.W1F
$B11369.seq002F_F103773_F714961e_CS13_CS13.W2F\(B11420$.seq
003F_F103434_F714961g_CS13_CS13.W3F
$B11495.seq004R_F103773_F714961f_CS13_CS13.W2R\(B11421$.seq
005R_F103749_F714961d_CS13_CS13.W1R\(B11370.seq 006R_F104120_F714961a_CS13_T7ter.seq

The Sanger sequencing result files must be input from lower to higher according to the first three numbers of the file name. It is convenient to input the first three numbers and then use the tab key to fill the rest of the file name automatically. Our script is able to merge dozens of Sanger sequencing result files, which is far enough for typical lab needs.

3) After running finishes, there is a folder named ‘merged_sequence’, which contains the merged sequence file with the name ‘merged.seq’.

Algorithm

Our algorithm's starting point is to take advantage of the overlapping regions within the tandem Sanger sequences, which are commonly no less than 200 bps (Fig. 2). The alignment module of the Biopython is called to conduct the needle alignment of the two consecutive Sanger results. The script then parses the output file of needle alignment to seek for the first line with full consensus, which is 50 bps, to obtain the critical coordinates of the corresponding Sanger sequences. These critical coordinates are then used to extract base pairs from each Sanger sequence for merging to the full-length sequence.

Results

As can be seen in the directory after a script running, the merged Sanger sequence ‘merged.seq’ is stored in a new folder named ‘merged_sequence’ (Fig. 3).

Besides, there are outputs in the screen showing the needle alignment and the critical junction of the consensus region between the two adjacent Sanger sequences (Fig 4), which clarifies the working process of merging. Further, the merged sequence is shown on the screen. Our results showed that the running process of the script is fast, and the output is clear.

Discussion

Merging Sanger sequences is routinely required during the gene cloning. The EMBOSS command merger is capable of generating a merged sequenced based on the global alignment of Needleman and Wunsch [7]. In the case of merging two overlapping Sanger sequences, the global alignment might bring in mismatches due to the decrease of signal-to-noise ratio during the later stage of the Sanger sequencing reaction and gel running. The EMBOSS merger deals with the disputed bases according to the local sequence quality score, which is rather complicated and often introduces the unwanted bases.

To address this issue, we look for the first full consensus line within the output of needle alignment and pick up the fragments to join the merged sequence according to the variations of the signal-to-noise ratio of the Sanger sequencing reaction, which circumvents the dispute settlement for the mismatch bases.

Conclusion

In sum, we provide a simple and direct method to merge the Sanger sequences via Python programming, which can be run at a local computer and satisfies the need during the gene cloning process.

Declarations

Compliance with Ethical Standards

Conflict of Interest The authors declare no conflict of interest.

References

1. Okegawa, Y. and K. Motohashi, *A simple and ultra-low cost homemade seamless ligation cloning extract (SLiCE) as an alternative to a commercially available seamless DNA cloning kit.* Biochem Biophys Rep, 2015. **4**: p. 148-151.
2. Zimmermann, J., et al., *Automated Sanger dideoxy sequencing reaction protocol.* FEBS Lett, 1988. **233**(2): p. 432-6.
3. *https://www.dnastar.com/.*
4. *https://dnaman.software.informer.com/.*

5. *https://www.thermofisher.com/cn/zh/home/life-science/cloning/vector-nti-software.html.*
6. Bell, T.G. and A. Kramvis, *Fragment merger: an online tool to merge overlapping long sequence fragments.* Viruses, 2013. **5**(3): p. 824-33.
7. Rice, P., I. Longden, and A. Bleasby, *EMBOSS: the European Molecular Biology Open Software Suite.* Trends Genet, 2000. **16**(6): p. 276-7.
8. Cock, P.J., et al., *Biopython: freely available Python tools for computational molecular biology and bioinformatics.* Bioinformatics, 2009. **25**(11): p. 1422-3.
9. *Python. https://www.python.org/. accessed 14 May 2020.*
10. Ran, F.A., et al., *Genome engineering using the CRISPR-Cas9 system.* Nat Protoc, 2013. **8**(11): p. 2281-2308.

Supplementary Materials

The example input files and the Python script Merge_Sanger_v2.py can be found in Supplementary Materials.

Figures

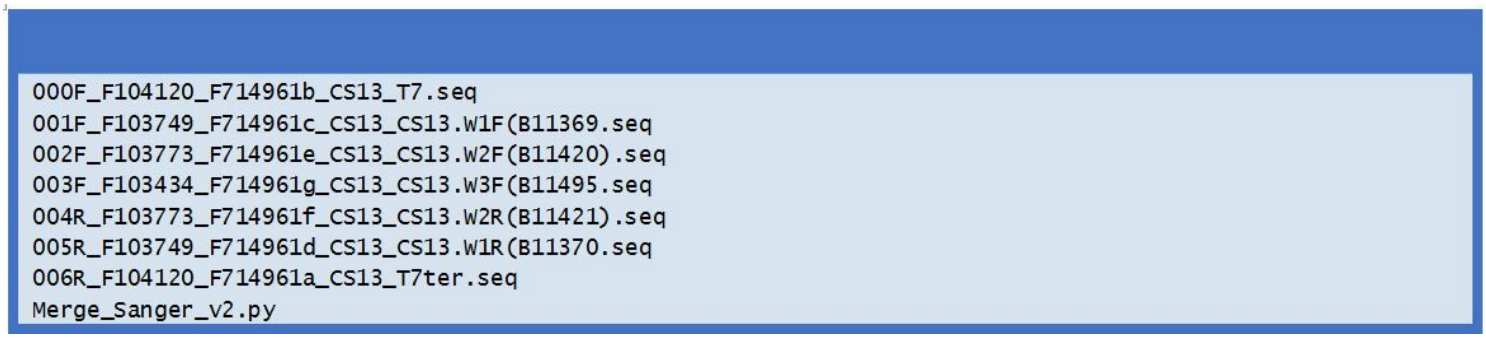

Figure 1

The test files used for running the script

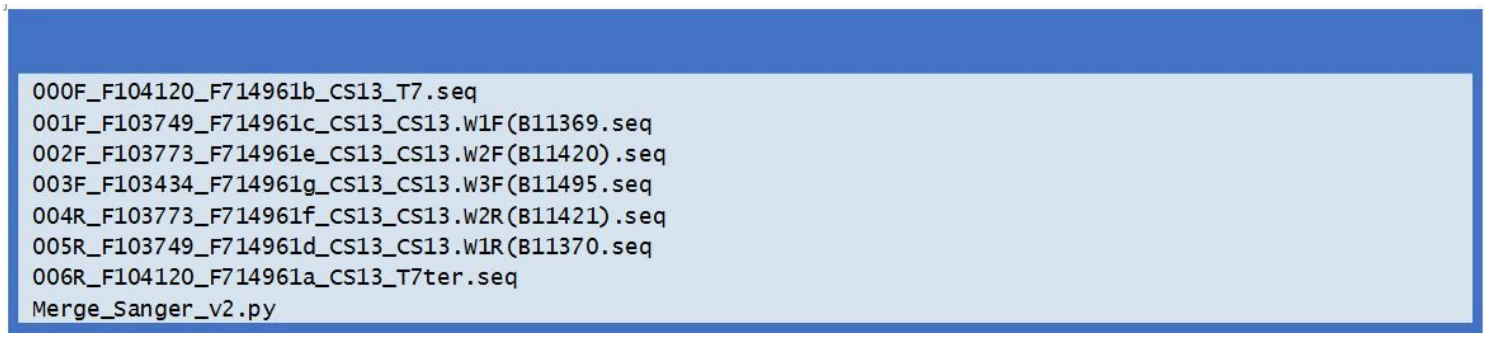

Figure 1

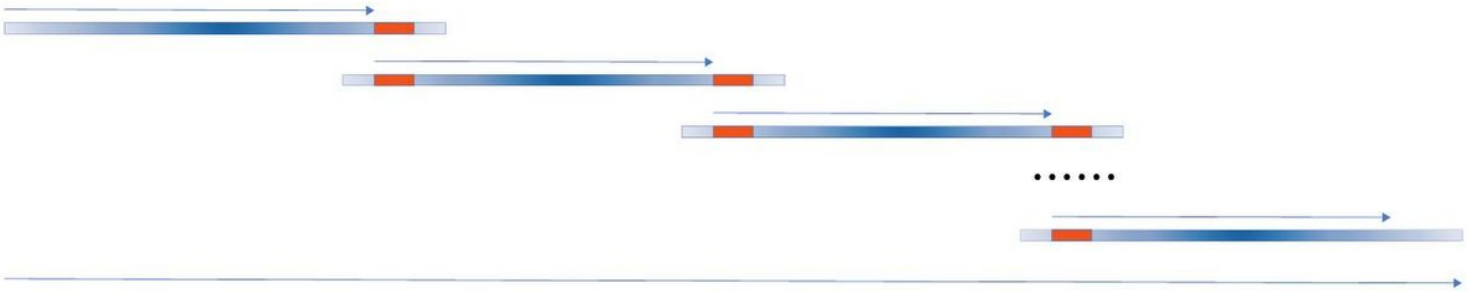

Figure 2

A diagrammatic view of merging multiple Sanger sequences. The thick blue lines show the multiple (no less than two) Sanger sequences to be merged, with the color depth indicating the sequencing quality, which is high in the middle part and low in both ends. The thick orange lines represent the first line of full consensus (50 bps long) appearing in the needle output file for the two consecutive Sanger sequences. The thin blue arrow above the thick blue line indicates the part to be extracted for joining to the full-length sequence, which is shown by the long thin blue arrow at the bottom. All the Sanger sequences are preprocessed to 5’-3’ direction, aligned and merged, thus the direction of all thin arrows is 5’-3’

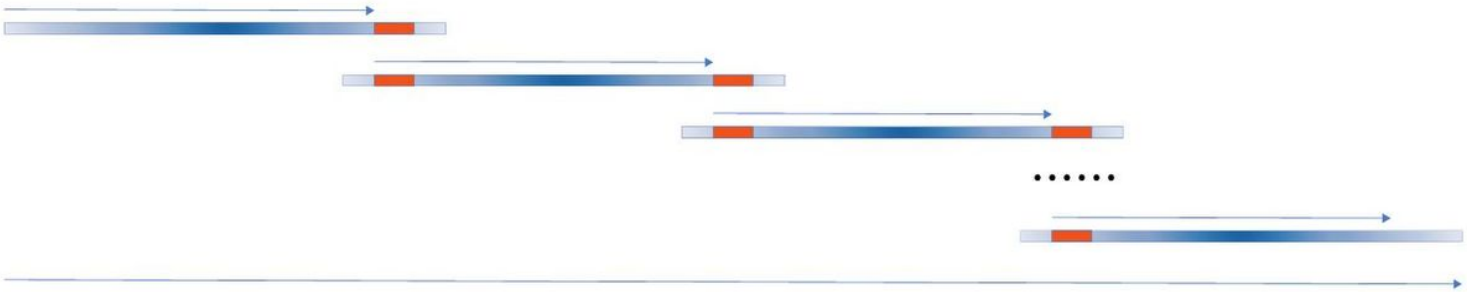

Figure 2

A diagrammatic view of merging multiple Sanger sequences. The thick blue lines show the multiple (no less than two) Sanger sequences to be merged, with the color depth indicating the sequencing quality, which is high in the middle part and low in both ends. The thick orange lines represent the first line of full consensus (50 bps long) appearing in the needle output file for the two consecutive Sanger sequences. The thin blue arrow above the thick blue line indicates the part to be extracted for joining to the full-length sequence, which is shown by the long thin blue arrow at the bottom. All the Sanger sequences are preprocessed to 5’-3’ direction, aligned and merged, thus the direction of all thin arrows is 5’-3’

```
.
├── 000F_F104120_F714961b_CS13_T7.seq
├── 001F_F103749_F714961c_CS13_CS13.W1F(B11369.seq
├── 002F_F103773_F714961e_CS13_CS13.W2F(B11420).seq
├── 003F_F103434_F714961g_CS13_CS13.W3F(B11495.seq
├── 004R_F103773_F714961f_CS13_CS13.W2R(B11421).seq
├── 005R_F103749_F714961d_CS13_CS13.W1R(B11370.seq
├── 006R_F104120_F714961a_CS13_T7ter.seq
├── Merge_Sanger_v2.py
└── merged_sequence
    └── merged.seq
```

Figure 3

The directory showing the output file after merging the Sanger sequences


```
.
├── 000F_F104120_F714961b_CS13_T7.seq
├── 001F_F103749_F714961c_CS13_CS13.W1F(B11369.seq
├── 002F_F103773_F714961e_CS13_CS13.W2F(B11420).seq
├── 003F_F103434_F714961g_CS13_CS13.W3F(B11495.seq
├── 004R_F103773_F714961f_CS13_CS13.W2R(B11421).seq
├── 005R_F103749_F714961d_CS13_CS13.W1R(B11370.seq
├── 006R_F104120_F714961a_CS13_T7ter.seq
├── Merge_Sanger_v2.py
└── merged_sequence
    └── merged.seq
```

Figure 3

The directory showing the output file after merging the Sanger sequences

Figure 4

The screenshots of the script run showing the partial output of the needle alignment (left panel) and the first consensus line and the merged DNA sequence (right panel).

Figure 4

The screenshots of the script run showing the partial output of the needle alignment (left panel) and the first consensus line and the merged DNA sequence (right panel).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- 000FF104120F714961bCS13T7.seq
- 000FF104120F714961bCS13T7.seq
- 001FF103749F714961cCS13CS13.W1FB11369.seq
- 001FF103749F714961cCS13CS13.W1FB11369.seq
- 002FF103773F714961eCS13CS13.W2FB11420.seq
- 002FF103773F714961eCS13CS13.W2FB11420.seq
- 003FF103434F714961gCS13CS13.W3FB11495.seq
- 003FF103434F714961gCS13CS13.W3FB11495.seq
- 004RF103773F714961fCS13CS13.W2RB11421.seq
- 004RF103773F714961fCS13CS13.W2RB11421.seq
- RB11370.seq

- 005RF103749F714961dCS13CS13.W1RB11370.seq
- 006RF104120F714961aCS13T7ter.seq
- 006RF104120F714961aCS13T7ter.seq
- MergeSangerv2.py
- MergeSangerv2.py