MergeSanger V3.1 Program Test Report

1 INTRODUCTION

1.1 Aim

The purpose of this test is to test for sequencing errors.

1.2 Scope

The scope of this test is the merge function in the GUI mode of the program. The merge function of CLI mode is the same as that of GUI mode.

2 ENVIRONMENTS

2.1 Hardware

CPU: 13th Gen Intel (R) Core (TM) i5-13420H 2.10 GHz
RAM: 16.0 GB

2.2 Software

Windows 11 Home Edition
Python 3.12
biopython 1.83
numpy 2.1.0
MergeSangerV3.1

3 TEST CASES

3.1 Design of test cases

The design of test cases is shown in the Figure below. For a typical Sanger sequencing reaction (may be more than 1000 bases), the signal quality goes from low to high, and to low again. There are generally at least 700 bases with fine signal. In very rare cases, however, errors might occur within these 700 bases, due to continuously identical bases ('AAAA' for example) within some genes, which causes a signal overlap that is difficult for the peak reading program to discern the number of bases. Also, there might be some transient experimental disturbances during the sequencing reaction that give rise to sequencing errors. However, the Sanger sequencing is a mature and robust technique and the probability of sequencing error (within the fine signal region) is very low. In the event of suspecting sequencing errors, one may re-sequence the gene regions, or redo the gene cloning. Here we manually mutate bases in different regions of the sequences and test how the V3.1 program processes these errors.

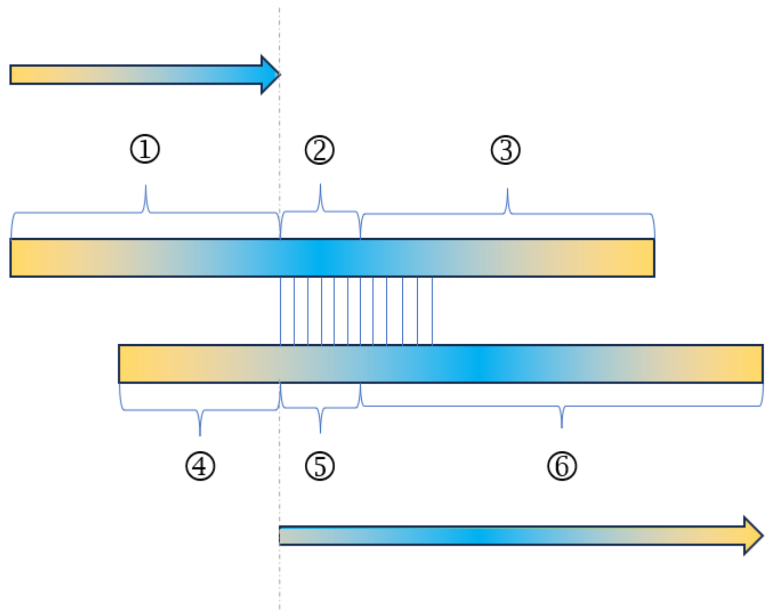


Figure The design of test cases. The rectangle above the 'F104120_F714961b_CS13_T7.seq' (seq1) and the rectangle below represents 'F103749_F714961c_CS13_CS13.W1F(B11369.seq' (seq2). The '2' (in circle) and '5' (in circle) indicate the region of first time 50 bases consecutive match between seq1 and seq2. The other four regions, namely, 1, 3, 4, and 6, are also for artificially mutating bases to simulate sequencing errors. The arrows show the range of merged sequence if there are only two sequences, i.e., seq1 and seq2. The gray vertical line shows where the first time 50 bases consecutive match starts. The colors stand for the signal quality of sequencing data, yellow for low and blue for high.

Test method（Test case 0 - Test case 6）

(1) Merge the 7 gene sequences to obtain the merged sequence merged.seq.

(2) Each time one base in region 1,2,3,4,5, or 6 is mutated (simulating sequencing error) to get the merged sequence (merged1.seq, merged2.seq, ,,, merged6.seq).

(3) The needle of the EMBOSS is used to align merged.seq and merged1.seq-merged6.seq, respectively, to see the difference between the aligned sequences.

Test method（Test case 7）

(1) Manually modify multiple bases of the first sequence so that sequence 1 and sequence 2 cannot find a consecutive 50 match.

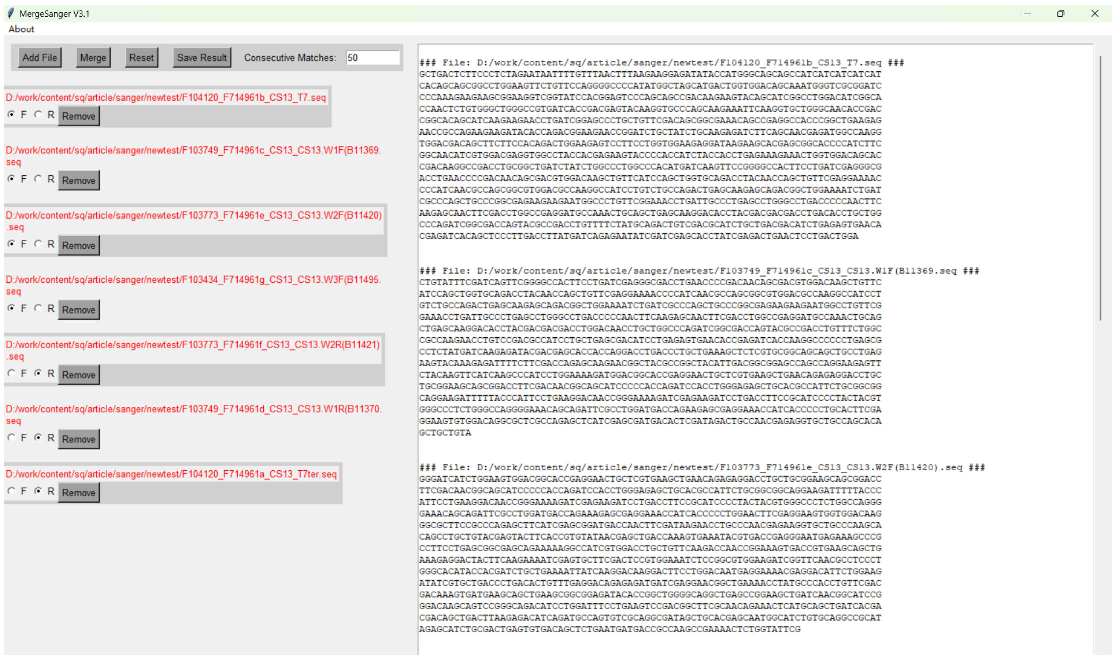
(2) Observe the results of the program.

3.2 Test cases

3.2.0 Test case 0

Test case description: Merge 7 gene sequences (sequence names) to obtain the merged sequence merged.seq.

(1) Add the 7 gene sequences through the Add File and select the F/R attribute.



Consecutive Matches: 50

D:/work/content/sq/article/sanger/newtest/F104120_F714961b_CS13_T7.seq [✓ F] R

D:/work/content/sq/article/sanger/newtest/F103749_F714961c_CS13_CS13.W1F(B11369.seq [✓ F] R

D:/work/content/sq/article/sanger/newtest/F103773_F714961e_CS13_CS13.W2F(B11420).seq [✓ F] R

D:/work/content/sq/article/sanger/newtest/F103434_F714961g_CS13_CS13.W3F(B11495.seq [✓ F] R

D:/work/content/sq/article/sanger/newtest/F103773_F714961f_CS13_CS13.W2R(B11421).seq F [✓ R]

D:/work/content/sq/article/sanger/newtest/F103749_F714961d_CS13_CS13.W1R(B11370.seq F [✓ R]

D:/work/content/sq/article/sanger/newtest/F104120_F714961a_CS13_T7ter.seq F [✓ R]

```
### File: D:/work/content/sq/article/sanger/newtest/F104120_F714961b_CS13_T7.seq ###
GCTGACTCTTCCCTCTAGAATAATTTTGTTTAACTTTAAGAAGGAGATATACCATGGGCAGCAGCCATCATCATCATCAT
CACAGCAGCGGCCTGGAAGTTCTGTTCCAGGGGCCCATATGGCTAGCATGACTGGTGGACAGCAAATGGGTCGCGGATC
CCCAAAGAAGAAGCGGAAGGTCGGTATCCACGGAGTCCCAGCAGCCGACAAGAAGTACAGCATCGGCCTGGACATCGGCA
CCAACTCTGTGGGCTGGGCCGTGATCACCGACGAGTACAAGGTGCCCAGCAAGAAATTCAAGGTGCTGGGCAACACCGAC
CGGCACAGCATCAAGAAGAACCTGATCGGAGCCCTGCTGTTCGACAGCGGCGAAACAGCCGAGGCCACCCGGCTGAAGAG
AACCGCCAGAAGAAGATACACCAGACGGAAGAACCGGATCTGCTATCTGCAAGAGATCTTCAGCAACGAGATGGCCAAGG
TGGACGACAGCTTCTTCCACAGACTGGAAGAGTCCTTCCTGGTGGAAGAGGATAAGAAGCACGAGCGGCACCCCATCTTC
GGCAACATCGTGGACGAGGTGGCCTACCACGAGAAGTACCCCACCATCTACCACCTGAGAAAGAAACTGGTGGACAGCAC
CGACAAGGCCGACCTGCGGCTGATCTATCTGGCCCTGGCCCACATGATCAAGTTCCGGGGCCACTTCCTGATCGAGGGCG
ACCTGAACCCCGACAACAGCGACGTGGACAAGCTGTTCATCCAGCTGGTGCAGACCTACAACCAGCTGTTCGAGGAAAAC
CCCATCAACGCCAGCGGCGTGGACGCCAAGGCCATCCTGTCTGCCAGACTGAGCAAGAGCAGACGGCTGGAAAATCTGAT
CGCCCAGCTGCCCGGCGAGAAGAAGAATGGCCCTGTTCGGAAACCTGATTGCCCTGAGCCTGGGCCTGACCCCCAACTTC
AAGAGCAACTTCGACCTGGCCGAGGATGCCAAACTGCAGCTGAGCAAGGACACCTACGACGACGACCTGACACCTGCTGG
CCCAGATCGGCGACCAGTACGCCGACCTGTTTTCTATGCAGACTGTCGACGCATCTGCTGACGACATCTGAGAGTGAACA
CGAGATCACAGCTCCCTTGACCTTATGATCAGAGAATATCGATCGAGCACCTATCGAGACTGAACTCCTGACTGGA


### File: D:/work/content/sq/article/sanger/newtest/F103749_F714961c_CS13_CS13.W1F(B11369.seq ###
CTGTATTTCGATCAGTTCGGGGCCACTTCCTGATCGAGGGCGACCTGAACCCCGACAACAGCGACGTGGACAAGCTGTTC
ATCCAGCTGGTGCAGACCTACAACCAGCTGTTCGAGGAAAACCCCATCAACGCCAGCGGCGTGGACGCCAAGGCCATCCT
GTCTGCCAGACTGAGCAAGAGCAGACGGCTGGAAAATCTGATCGCCCAGCTGCCCGGCGAGAAGAAGAATGGCCTGTTCG
GAAACCTGATTGCCCTGAGCCTGGGCCTGACCCCCAACTTCAAGAGCAACTTCGACCTGGCCGAGGATGCCAAACTGCAG
CTGAGCAAGGACACCTACGACGACGACCTGGACAACCTGCTGGCCCAGATCGGCGACCAGTACGCCGACCTGTTTCTGGC
CGCCAAGAACCTGTCCGACGCCATCCTGCTGAGCGACATCCTGAGAGTGAACACCGAGATCACCAAGGCCCCCCTGAGCG
CCTCTATGATCAAGAGATACGACGAGCACCACCAGGACCTGACCCTGCTGAAAGCTCTCGTGCGGCAGCAGCTGCCTGAG
AAGTACAAAGAGATTTTCTTCGACCAGAGCAAGAACGGCTACGCCGGCTACATTGACGGCGGAGCCAGCCAGGAAGAGTT
CTACAAGTTCATCAAGCCCATCCTGGAAAAGATGGACGGCACCGAGGAACTGCTCGTGAAGCTGAACAGAGAGGACCTGC
TGCGGAAGCAGCGGACCTTCGACAACGGCAGCATCCCCCACCAGATCCACCTGGGAGAGCTGCACGCCATTCTGCGGCGG
CAGGAAGATTTTTACCCATTCCTGAAGGACAACCGGGAAAAGATCGAGAAGATCCTGACCTTCCGCATCCCCTACTACGT
GGGCCCTCTGGGCCAGGGGAAACAGCAGATTCGCCTGGATGACCAGAAGAGCGAGGAAACCATCACCCCCTGCACTTCGA
GGAAGTGTGGACAGGCGCTCGCCAGAGCTCATCGAGCGATGACACTCGATAGACTGCCAACGAGAGGTGCTGCCAGCACA
GCTGCTGTA


### File: D:/work/content/sq/article/sanger/newtest/F103773_F714961e_CS13_CS13.W2F(B11420).seq ###
GGGATCATCTGGAAGTGGACGGCACCGAGGAACTGCTCGTGAAGCTGAACAGAGAGGACCTGCTGCGGAAGCAGCGGACC
TTCGACAACGGCAGCATCCCCCACCAGATCCACCTGGGAGAGCTGCACGCCATTCTGCGGCGGCAGGAAGATTTTTACCC
ATTCCTGAAGGACAACCGGGAAAAGATCGAGAAGATCCTGACCTTCCGCATCCCCTACTACGTGGGCCCTCTGGCCAGGG
GAAACAGCAGATTCGCCTGGATGACCAGAAAGAGCGAGGAAACCATCACCCCCTGGAACTTCGAGGAAGTGGTGGACAAG
GGCGCTTCCGCCCAGAGCTTCATCGAGCGGATGACCAACTTCGATAAGAACCTGCCCAACGAGAAGGTGCTGCCCAAGCA
CAGCCTGCTGTACGAGTACTTCACCGTGTATAACGAGCTGACCAAAGTGAAATACGTGACCGAGGGAATGAGAAAGCCCG
CCTTCCTGAGCGGCGAGCAGAAAAAGGCCATCGTGGACCTGCTGTTCAAGACCAACCGGAAAGTGACCGTGAAGCAGCTG
AAAGAGGACTACTTCAAGAAAATCGAGTGCTTCGACTCCGTGGAAATCTCCGGCGTGGAAGATCGGTTCAACGCCTCCCT
GGGCACATACCACGATCTGCTGAAAATTATCAAGGACAAGGACTTCCTGGACAATGAGGAAAACGAGGACATTCTGGAAG
ATATCGTGCTGACCCTGACACTGTTTGAGGACAGAGAGATGATCGAGGAACGGCTGAAAACCTATGCCCACCTGTTCGAC
GACAAAGTGATGAAGCAGCTGAAGCGGCGGAGATACACCGGCTGGGGCAGGCTGAGCCGGAAGCTGATCAACGGCATCCG
GGACAAGCAGTCCGGGCAGACATCCTGGATTTCCTGAAGTCCGACGGCTTCGCAACAGAAACTCATGCAGCTGATCACGA
CGACAGCTGACTTAAGAGACATCAGATGCCAGTGTCGCAGGCGATAGCTGCACGAGCAATGGCATCTGTGCAGGCCGCAT
AGAGCATCTGCGACTGAGTGTGACAGCTCTGAATGATGACCGCCAAGCCGAAAACTCTGGTATTCG
```

(2) Click Merge to merge

(3) Result

In directory ‘test_case/case0’:

directory ‘sequences’: test sequences

_merged.log

merged.seq

Please open them with TextEdit and set font as ‘Courier New’ for viewing.

3.2.1 Test case 1

(1) Mutate one base in region 1

Mutate A to G in position 598 in the first sequence, F104120_F714961b_CS13_T7.seq. The resultant file is designated F104120_F714961b_CS13_T7_A598G.seq. Re-run the MergeSangerV3.1 program.

(2) Expected result

The merged sequence has one base difference from the merged.seq of 3.2.0.

(3) Analysis

The first sequence is mutated before the first 50 matches, and the position of the first 50 matches is not changed, resulting in a difference of one base (i.e., where the mutation occurred).

(4) Result

in directory ‘test_case/case1’:

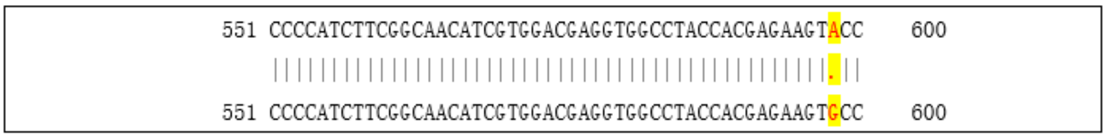
directory ‘sequences’: test sequences

_merged.log

merged1.seq

merged.seq

merged.needle: align merged.seq and merged1.seq using the needle



```
551 CCCCATCTTCGGCAACATCGTGGACGAGGTGGCCTACCACGAGAAGTACC     600
    |||||||||||||||||||||||||||||||||||||||||||||||.||
551 CCCCATCTTCGGCAACATCGTGGACGAGGTGGCCTACCACGAGAAGTGCC     600
```

As can be seen from the merged.needle file, merged1.seq and merged.seq have a different base at position 598.

(5) Test status

Pass

3.2.2 Test case 2

(1) Mutate one base in region 2

Mutate A to C in position 703 in the first sequence, F104120_F714961b_CS13_T7.seq. The resultant file is designated F104120_F714961b_CS13_T7_A703C.seq. Re-run the MergeSangerV3.1 program.

(2) Expected result

The merged sequence has one base difference from the merged.seq of 3.2.0.

(3) Analysis

A mutation occurs in the middle of the first 50 matches of the first sequence, resulting in a change in the position of the first 50 matches, and a difference of one base (i.e., where the mutation occurred).

(4) Result

in directory ‘test_case/case2’:

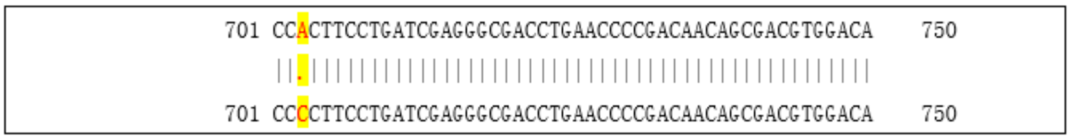
directory ‘sequences’: test sequences

_merged.log

merged2.seq

merged.seq

merged.needle: align merged.seq and merged2.seq using the needle



```
701 CCACTTCCTGATCGAGGGCGACCTGAACCCCGACAACAGCGACGTGGACA    750
    ||.|||||||||||||||||||||||||||||||||||||||||||||||
701 CCCCTTCCTGATCGAGGGCGACCTGAACCCCGACAACAGCGACGTGGACA    750
```

As can be seen from the merged.needle file, merged1.seq and merged.seq have a different base at position 703.

(5) Test status

Pass

3.2.3 Test case 3

(1) Mutate one base in region 3

Mutate C to T in position 1175 in the first sequence, F104120_F714961b_CS13_T7.seq. The resultant file is designated F104120_F714961b_CS13_T7_C1175T.seq. Re-run the MergeSangerV3.1 program.

(2) Expected result

The merged result merged3.seq is the same as merged.seq from 3.2.0.

(3) Analysis

The mutation occurs after the first 50 matches of the first sequence, the position of the first 50 matches does not change, and the mutated base is not merged into the final sequence, so merged3.seq and merged.seq are identical.

(4) Result

in directory ‘test_case/case3’:

directory ‘sequences’: test sequences

_merged.log

merged3.seq

merged.seq

merged.needle: align merged.seq and merged3.seq using the needle

It can be seen from merged.needle that the merged3.seq and merged.seq have identical sequences.

(5) Test status

Pass

3.2.4 Test case 4

(1) Mutate one base in region 4

Mutate T to A in position 2 in the second sequence, F103749_F714961c_CS13_CS13.W1F(B11369.seq. The resultant file is designated F103749_F714961c_CS13_CS13.W1F(B11369_T2A.seq. Re-run the MergeSangerV3.1 program.

(2) Expected result

The merged result merged4.seq is the same as merged.seq from 3.2.0.

(3) Analysis

The mutation precedes the first 50 matches of the second sequence, the position of the first 50 matches is not changed, and the mutated base is not merged into the final sequence, so merged4.seq and merged.seq are identical.

(4) Result

In directory ‘test_case/case4’:

directory ‘sequences’: test sequences

_merged.log

merged4.seq

merged.seq

merged.needle: align merged.seq and merged4.seq using the needle

It can be seen from merged.needle that the merged4.seq and merged.seq have identical sequences.

(5) Test status

Pass

3.2.5 Test case 5

(1) Mutate one base in region 5

Mutate A to C in position 25 in the second sequence, F103749_F714961c_CS13_CS13.W1F(B11369.seq. The resultant file is designated F103749_F714961c_CS13_CS13.W1F(B11369_A25C.seq. Re-run the MergeSangerV3.1 program.

(2) Expected result

The merged result merged5.seq is the same as merged.seq from 3.2.0.

(3) Analysis

The first 50 matches of the second sequence (region 5) have a mutation in the middle, and the position of the first 50 match is changed, and the mutated base is not merged into the final sequence, so merged5.seq and merged.seq are identical.

(4) Result

in directory ‘test_case/case5’:

directory ‘sequences’: test sequences

_merged.log

merged5.seq

merged.seq

merged.needle: align merged.seq and merged5.seq using the needle

It can be seen from merged.needle that the merged5.seq and merged.seq have identical sequences.

(5) Test status

Pass

3.2.6 Test case 6

(1) Mutate one base in region 6

Mutate A to G in position 1019 in the second sequence, F103749_F714961c_CS13_CS13.W1F(B11369.seq. The resultant file is designated F103749_F714961c_CS13_CS13.W1F(B11369_A1019G.seq. Re-run the MergeSangerV3.1 program.

Note: The location of the region 6 mutation can be divided into three situations, before, in the middle of, and after the consecutive 50 matches between sequence file 2 and sequence file 3. These situations are quite similar to the previous cases, i.e., case 1, case2, and case3. The last situation is taken here as an example.

(2) Expected result

The merged result merged6.seq is the same as merged.seq from 3.2.0.

(3) Analysis

The mutation occurs after the first 50 match between seq2 and seq3 (at the end of seq2), the position of the first 50 match between seq2 and seq3 does not change, and the mutated base is not merged into the final sequence, so merged6.seq and merged.seq are identical.

(4) Result

in directory ‘test_case/case6’:

directory ‘sequences’: test sequences

_merged.log

merged6.seq

merged.seq

merged.needle: align merged.seq and merged6.seq using the needle

It can be seen from merged.needle that the merged6.seq and merged.seq have identical sequences.

(5) Test status

Pass

3.2.7 Test case 7

(1) Modify multiple bases for sequence 1

Modify multiple bases of the first sequence, F104120_F714961b_CS13_T7.seq, so that the first and second sequences do not find a continuous 50 match. Run the MergeSanger V3.1 program.

(2) Expected result

The merging fails and a suggestion is given.

(3) Analysis

When seq1 and seq2 cannot find a continuous 50 match, the merging cannot be successful. You can observe the pairwise alignments log to determine whether the data is a problem, (the accompanying ab1 files shall be used to see if the sequencing quality is too low) and if it is, you may need to redo the experiment (re-sequencing the gene region, perhaps using a different sequencing primer).

When the merging failure occurs in the middle, for example, seq3 and seq4 do not find a continuous match 50, then you can use the program to merge seq1-seq3 to get mergedA.seq, and then use the program to merge seq4-seq7 to obtain mergedB.seq. Finally, the Consecutive Matches parameter in the GUI is adjusted according to the number of consecutive matches in seq3 and seq4 to merge mergedA.seq and mergedB.seq to obtain the final merged sequence.

(4) Result

In directory ‘test_case/case7’:
directory ‘sequences’
_merged.log
There is no merged.seq file

(5) Test status

Pass

4 Analysis of test results

The individual bases of the regions 1-6 were manually modified to simulate sequencing errors, and the test results were as expected. Manually modifying multiple bases of sequence 1 simulated splicing failures, and the test results are as expected.

5 Analysis of consecutive matches

From the merged.log in case0 we can see that
Sequences 1 and 2 are actually matched for 200+ bases consecutively when they first match 50
Sequences 2 and 3 are actually matched for 200+ bases consecutively when they first match 50
Sequences 3 and 4 are actually matched 200+ bases consecutively when they first match 50
Sequences 4 and 5 are actually matched 130+ bases consecutively when they first match 50
Sequences 5 and 6 are actually matched 75+ bases consecutively when they first match 50
Sequences 6 and 7 are actually matched consecutively 290+ bases when they first match 50

It can be seen that the default ‘Consecutive Matches’ in GUI set 50 is reasonable. In fact, the sequencing company or core facility use the sequencing primers of plasmid (T7-for and T7-ter) to perform the first forward and the first reversing sequencing reactions, respectively. There is an enough space between the place where T7-for anneals and where the gene resides in the plasmid, so that the sequencing signal is fine when it comes to the gene. The situation of the T7-rev is the same. Then, the sequencing company or core facility will conduct walking sequencing (forward walking and reverse walking) to determine the rest regions of the gene. The primer of each walking sequencing is designed according to the previously obtained sequence. There is an enough space between where the walking primer anneals and where the next region is to be sequenced. Hence, we get the consecutive matches listed above, which ensures that our program works. The program looks for these consecutive matches using alignment and collects the regions (see the arrows in the Figure in page 1 of this document) with fine sequencing quality to get the final merged sequence.

6 Conclusion

MergeSanger V3.1 satisfies the situation of sequencing error.