## **Supporting Information**

### Supplementary materials and methods

### Additional information on bioinformatic analysis of small RNA sequences.

The initial parsing of raw small RNA sequence tags using the Python script "CLASSIFY". To identify the high quality small RNA reads from the raw sequencing data file provided by Illumina, we created a Python script named "CLASSIFY" (see below) to extract the small RNAs sequences with both identifiable 5' and 3' adapter sequences. Sequences tags with a non-matching 5'- or 3'-adapter, with adaptor selfligation, or with a GFP size marker-specific sequences were removed from further analysis (Fig. 1B). The resulting small RNA tags were then sorted into the four "source libraries" based on the 2-nt sequence index in the 5'-adaptor (Fig. 1B). Small RNAs that are shorter than 18-nt or longer than 25-nt were discarded from each of the four libraries (Fig. 1B).

Sorting of small RNAs of viral and host origin. For each of the four source libraries, small RNAs with a sequence perfectly matching the TMV-Cg genome in either sense or antisense orientation were identified and sorted into a separated bin designated as TMV-Cg-derived small RNAs (Fig. 1C). This was done using the Python script "MAP" (see below). Similarly, small RNAs with a sequence perfectly matching the *Arabidopsis thaliana* nuclear and organellar genomes (version TAIR 8), respectively, were extracted sequentially into distinct bins using the "MAP" algorithm. Of note, from the pool of 18-~25-nt small RNAs, the number of small RNAs matching the nuclear genome were 788,710 (76.9%; Col-0 mock), 628,627 (71.53%, Col-0 TMV-Cg), 616,147(73.9%, *rdr1-*1 TMV-Cg), and 575,365 (70.38%, *rdr6-15* TMV-Cg), respectively. Those matching the

1

organellar genomes were 44,472 (4.3%; Col-0 mock), 30,164 (3.4%, Col-0 TMV-Cg), 44,819 (5.4%, *rdr1*-1 TMV-Cg), and 38,281 (4.7%, *rdr6*-15 TMV-Cg), respectively. Small RNAs that match neither the host nor viral genomic sequence were designated as trash and discarded from further analysis. Of note, the number of small RNAs sorted into the trash bin were 192,129 (18.7%; Col-0 mock), 119,166 (13.5%, Col-0 TMV-Cg), 159,251(19.1%, *rdr1*-1 TMV-Cg), and 141,735 (17.3%, *rdr6*-15 TMV-Cg), respectively. Computational generation of TMV-Cg-derived of 21-nt siRNAs. The TMV-Cg-derived 21-nt siRNAs were generated *in silico* using the Python script "CUT" (see below). The algorithm generates a total of 12,566 TMV-Cg-derived sense and antisense 21-nt small RNAs from the 6,303-nt genome in all possible phases.

A pseudocode, as well as the actual source code for each of the three Python scripts used in this study is provided below.

**Small RNA blot.** Small RNA blot assays were done as described previously (Xie et al., 2005). [<sup>32</sup>P]-end labeled DNA oligonucleotides were used as probes. The probe sequences for each of the TMV-Cg-derived siRNAs were listed below:

5'- ATGAGTTCGGTGCTGCATTGC -3' for TMV-Cg-siR696(-);

5'-GCAATGCAGCACCGAACTCAT-3' for TMV-Cg-siR696(+);

and 5'-TCCAGCTTCAATCCTTAAATT-3' for TMV-Cg-siR5293(+).

**Experimental validation of host targets for viral siRNAs.** Validation of target cleavage by RLM-5'RACE was done as described previously (Llave et al., 2002). Sequences of the gene-specific primers used in 5'RACE are listed below:

(1) At4g36150\_3746R: 5'-CTGTGGGATGATTTGAATAGCCATT-3' and At4g36150\_3646R: 5'-ACCCGTTGGAAGAATTTTAGCTAACT-3' for At4g36150; (2) At1g22870\_2866R: 5'-GCACAGAGGCTCTATGATAAATCACA-3' and At1g22870\_2859R:5'-GCTCTATGATAAATCACAGACGAGAT-3' for At1g22870; (3) At2g01390\_512R: 5'-CCGGAGCTTGAAACCCAGTGAATCA-3' and At2g01390\_412R: 5'-TCCGACCGGCTTCTCCGAAGATAT-3' for At2g01390; (4) At3g10390\_658R: 5'-CGGGTCGTTTCCTACCCTCCAAA-3' and At3g10390\_603R: 5'-CCTAGCTGCAGCCAACCCAGATA-3' for At3g10390; (5) At3g61960\_1772R: 5'-CCCGCGGTCTATTTTCGCTAAAGA-3' and At3g61960\_1743R: 5'-CCGACGATGCTGTAAGTTAGAGAT-3' for At3g61960; (6) AT5G51070\_2961R: 5'-ACCTCACGGACTAATCACTATGTAT-3' and AT5G51070\_2784R: 5'-CCGAACCGATGGGTTTCCGGTAT-3' for At5g51070; (7) At1g43850\_1390R: 5'- GCTCTTCATGCCGCCGAGCACAA-3' and At1g43850\_1260R:5'- CTCCAGGACAATTTGGCCAGATGAA-3' for At1g43850; (8) At5g64550\_1947R: 5'- AGCCTTGTTGTTACCACATAGAGA-3' and At5g64550\_1861R: 5'-GCCCGCCACCATGAACTCTTTCCT-3' for At5g64550; (9) At3g54090\_654R: 5'- CGCCATCATTTTCCCGTCCTTAAACT-3' and At3g54090\_578R: 5'- GCTCTCGTCTGAACTCTCTCCTGATT-3' for At3g54090; (10) At1g67710\_1664R: 5'- GGAAGGTTCTTGGGAAAGCAAGATTA-3' and At1g67710\_1522R: 5'- CAGGGTCAGGCAGCCGCTCATTT-3' for At1g67710; (11) At3g05340\_580R: 5'- TCCCCACGGAAATTTCCTTGTCATAA-3' and At3g05340\_457R: 5'- ACCCACCTGACCCAAGCATTCGTT-3' for At3g05340;

(12) At4g34910\_1089R: 5'- CTGGCTATTATCATCCGTTGCAATCA-3' and At4g34910\_815R: 5'- GGGACGGCCTCTTCCTTGTCATTAT-3' for At4g34910;
(13) At4g36810\_501R: 5'- ACGGCGGAGATCGTCGTTATCCATA-3' and At4g36810\_408R: 5'- GGTTGATTCTTCACCTCCGACGAGTT-3' for At4g36810, and
(14) At3g57880\_1974R: 5'- TAGCTCGTCGGGGGTGGGCTGAGT-3' and At3g57880\_1911R: 5'- CGGTCTCCAACGGTAGTACCAGATA-3' for At3g57880.

# References

- Llave, C., Xie, Z., Kasschau, K.D. and Carrington, J.C. (2002) Cleavage of Scarecrowlike mRNA targets directed by a class of Arabidopsis miRNA. *Science*, **297**, 2053-2056.
- Xie, Z., Allen, E., Wilken, A. and Carrington, J.C. (2005) DICER-LIKE 4 functions in trans-acting small interfering RNA biogenesis and vegetative phase change in Arabidopsis thaliana. *Proc Natl Acad Sci U S A*, **102**, 12984-12989.

# "CLASSIFY"

The input file is the sequencing result file in text (ASCII) format. The first column is the reads and the second column is the sequence. Below is a segment of an example input file:

98053 ACTTGTGGCCGAGGATGTTTCCGTCCTCGTATG78904 TGTTTGGATTGAAGGGAGCTCTATCGTATGCCG53996 CATTTGGATTGAAGGGAGCTCTATCGTATGCCG

*Pseudocode* for **CLASSIFY**:

| Result Parse all sequences in the sequencing result file and output parsing results into output filesOutput: Output files and statistics on the screensum0 $\leftarrow 0$ (reads of useful sequences);sum1 $\leftarrow 0$ (reads of sequences without 3' adapter match);sum2 $\leftarrow 0$ (reads of sequences without 3' adapter match);sum4 $\leftarrow 0$ (reads of sequences without 3' adapter match);sum4 $\leftarrow 0$ (reads of sequences without 3' adapter match);sum4 $\leftarrow 0$ (reads of GFP);Create and open output files used to store parsing resultwhile not at end of the input file dosequences jies (aC, 'CA', 'GT' or 'TG' thenif The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' thenis seq $\leftarrow$ seq without the first two bases.if The last is bases of seq are the same as the first i bases of 3' adapter thenis ym2 $\leftarrow seq$ without the last i basesIf a last i bases of seq are the same as the first i bases of 3' adapter thenis ym3 $\leftarrow seq$ -readsIf the last i bases of seq are the same as the first i bases of 3' adapter thenis ym3 $\leftarrow seq$ -reads (self-aliasing 3' adapter has been found)Jump to step 8elseendsum4 $\leftarrow sum4 + seq-readselseif The last i bases of seq are the same as the first i bases of gfp18 OR gfp24 thenJump to step 8elseif sum4 \leftarrow sum4 +$  
   
  |      | Inpu   | it: Sequencing result file  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
---
--
---
--
---
Output: Output files and statistics on the screen1 sum0 $\leftarrow$ 0 (reads of sequences without 3' adapter match);2 sum1 $\leftarrow$ 0 (reads of sequences without 3' adapter match);3 sum2 $\leftarrow$ 0 (reads of sequences without 3' adapter match);4 sum3 $\leftarrow$ 0 (reads of sequences with self-ligated sequences);5 sum4 $\leftarrow$ 0 (reads of sequences with self-ligated sequences);6 Create and open output files used to store parsing result7 while not at end of the input file do8 Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads9 if The first 2 bases of seq is 'AC', 'CA', 'CT' or 'TG' then10   seq $\leftarrow$ seq without the first two bases.11 if The last i bases of seq are the same as the first i bases of 3' adapter then12   sum2 $\leftarrow$ sum2 $\leftarrow$ seq-reads13   else14   sum3 $\leftarrow$ sum4 $+$ seq-reads15   Jump to step 816   end17   if The last i bases of seq are the same as the first i bases of 3' adapter then18   sum3 $\leftarrow$ sum3 $+$ seq-reads (self-aliasing 3' adapter thas been found)19   Jump to step 810   sum4 $\leftarrow$ sum4 $+$ seq-reads21   sum4 $\leftarrow$ sum4 $+$ seq-reads22   Output seq to corresponding output file23   end24   end25   end26   se end26   se end27   sum4 $\leftarrow$ sum4 $+$ seq-reads28   end29   end30   end31   sum5 $+$ sum1 $+$ seq-reads32   end33   end34   sum1 $\leftarrow$ sum1 $+$ seq-reads35   end36   end <tr <td<="" td=""><td></td><td>Resu</td><td>It: Parse all sequences in the sequencing result file and output parsing results into output files</td></tr> <tr><td><pre>sum0 0 (reads of useful sequences); sum1 0 (reads of sequences without 3' adapter match); sum2 0 (reads of sequences without 3' adapter match); sum3 0 (reads of sequences without 3' adapter match); sum4 0 (reads of sequences without 3' adapter match); create and open output files used to store parsing result while not at end of the input file do Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then seq ← seq without the first two bases. if The last i bases of seq are the same as the first i bases of 3' adapter then seq ← seq without the last i bases else else if The last i bases of seq are the same as the first i bases of 3' adapter then sum3 ← sum3 + seq-reads figure to sum2 + seq-reads sum3 ← sum3 + seq-reads (self-aliasing 3' adapter then sum4 ← sum4 + seq-reads else else i sum4 ← sum4 + seq-reads i sum4 ← sum4 + seq-reads i sum4 ← sum4 + seq-reads i else i sum1 ← sum1 + seq-reads i sum1 ← sum1 + seq-reads i sum1 ← sum4 + seq-reads i sum4 ← sum4 + seq-reads i su</pre></td><td></td><td>Outp</td><td>out: Output files and statistics on the screen</td></tr> <tr><td>2 sum1 ← 0 ( reads of sequences without 5' adapter match); 3 sum2 ← 0 ( reads of sequences without 3' adapter match); 4 sum3 ← 0 ( reads of sequences with self-ligated sequences); 5 sum4 ← 0 ( reads of GFP); 6 Create and open output files used to store parsing result 7 while not at end of the input file do 8 Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads 9 if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then 10   seq ← seq without the first two bases. 11   if The last i bases of seq are the same as the first i bases of 3' adapter then 12   seq ← seq without the last i bases 13   else 14   sum2 ← sum2 + seq-reads 15   Jump to step 8 16   end 17   if The last i bases of seq are the same as the first i bases of 3' adapter then 18   sum3 ← sum3 + seq-reads (self-aliasing 3' adapter has been found) 19   Jump to step 8 20   else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then 21   sum4 ← sum4 + seq-reads 22   oUtput seq to corresponding output file 23   else 24   oUtput seq to corresponding output file 24   end 25   else 26   sum1 ← sum1 + seq-reads 27   Jump to step 8 28   end 29   output seq to step 8 29   end 20   output seq to step 8 20   output seq to step 8 21   output seq to step 8 22   end 23   output seq to step 8 24   output seq to step 8 25   end 26   sum1 ← sum1 + seq-reads 27   Jump to step 8 28   end 29   output seq to step 8 29   end 20   output seq to step 8 20   end 21   output seq to step 8 22   end 32   output seq to step 8 33   output seq to step 8 34   output seq to step 8 35   output seq to step 8 36   end 39   output seq to step 8 30   output seq to step 8 31   output seq to step 8 32   end 33   output seq to step 8 34   output seq to step 8 35   output seq to step 8 35   output seq to step 8 36   output seq to step 8 37   output seq to step 8 38   output seq to step 8 39   output seq to step 8 30   output seq to step 8 31</td><td>1</td><td>sum(</td><td><math>0 \leftarrow 0</math> (reads of useful sequences);</td></tr> <tr><td>a sum2 ← 0 ( reads of sequences without 3' adapter match); 4 sum3 ← 0 ( reads of sequences with self-ligated sequences); 5 sum4 ← 0 ( reads of GFP); 6 Create and open output files used to store parsing result 7 while not at end of the input file do 8 Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads 9 if The first 2 bases of seq is 'AC', 'CA' or 'TG' then 1 seq ← seq without the first two bases. 1 if The last i bases of seq are the same as the first i bases of 3' adapter then 1 seq ← seq without the last i bases 1 else 1 sum2 ← sum2 + seq-reads 1 jump to step 8 1 else 1 if The last i bases of seq are the same as the first i bases of 3' adapter then 1 sum3 ← sum3 + seq-reads (self-aliasing 3' adapter thas been found) 1 jump to step 8 2 else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then 2 sum4 ← sum4 + seq-reads 2 else 2 l Output seq to corresponding output file 2 else 2 l Output seq to corresponding output file 2 else 3 sum4 ← sum1 + seq-reads 3 ljump to step 8 2 else 4 sum1 ← sum1 + seq-reads 2 else 4 sum1 ← sum1 + seq-reads 3 sum3 ← sum3 + seq-reads 4 sum3 ← sum4 ← sum4 + seq-reads 4 sum3 ← sum4 + seq-reads 5 sum3 ← sum3 + seq-reads 5 sum3 ← sum4 + seq-reads 5 sum3 ← sum4 ← sum4 + seq-reads 5 sum3 ← sum4 ← sum4 + seq-reads 5 sum3 ← sum3 ← sum3 ← sum3 ← sum3 ← sum4 ← sum4</td><td>2</td><td>sum1</td><td><math>1 \leftarrow 0</math> (reads of sequences without 5' adapter match);</td></tr> <tr><td>4 sum3 ← 0 ( reads of sequences with self-ligated sequences); 5 sum4 ← 0 ( reads of GFP); 6 Create and open output files used to store parsing result 7 while not at end of the input file do 8 Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads 9 if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then 10 seq seq without the first two bases. 11 if The last i bases of seq are the same as the first i bases of 3' adapter then 12 seq seq without the last i bases 13 else 14 sum2 ← sum2 + seq-reads 15 Jump to step 8 16 end 17 if The last i bases of seq are the same as the first i bases of 3' adapter then 18 sum3 ← sum3 + seq-reads (self-aliasing 3' adapter has been found) 19 Jump to step 8 20 else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then 21 sum4 ← sum4 + seq-reads 22 else 23 else 24 lese if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then 24 sequence sum4 + seq-reads 25 else 26 sequence sum4 + seq-reads 27 sum4 ← sum1 + seq-reads 28 else 29 end 30 print sum0, sum1, sum2, sum3, sum4 a 30 Close output files</td><td>3</td><td>sum2</td><td><math>2 \leftarrow 0</math> (reads of sequences without 3' adapter match);</td></tr> <tr><td>s sum4 ← 0 ( reads of GFP); 6 Create and open output files used to store parsing result 7 while not at end of the input file do 8 Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads 9 if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then 10 <math> seq - seq</math> without the first two bases. 11 if The last i bases of seq are the same as the first i bases of 3' adapter then 12 <math> seq - seq</math> without the last i bases 13 else 14 <math> sum2 \leftarrow sum2 + seq-reads</math> 15 <math> Jump to step 8</math> 16 end 17 if The last i bases of seq are the same as the first i bases of 3' adapter then 18 <math> sum3 \leftarrow sum3 + seq-reads</math> (self-aliasing 3' adapter thas been found) 19 <math> Jump to step 8</math> 20 else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then 21 <math> sum4 \leftarrow sum4 + seq-reads</math> 22 <math> else</math> 23 <math> Output seq to corresponding output file 24 <math> end</math> 25 <math> else</math> 26 <math> sum1 \leftarrow sum1 + seq-reads</math> 27 <math> Jump to step 8</math> 28 <math> end</math> 29 <math>end</math> 30 <math>print sum0, sum1, sum2, sum3, sum4</math> 31 <math> Close output files</math></math></td><td>4</td><td>sum?</td><td><math>3 \leftarrow 0</math> (reads of sequences with self-ligated sequences);</td></tr> <tr><td><ul> <li><sup>6</sup> Create and open output files used to store parsing result</li> <li>7 while not at end of the input file do</li> <li>8 Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads</li> <li>if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then</li> <li>is eq ← seq without the first two bases.</li> <li>if The last i bases of seq are the same as the first i bases of 3' adapter then</li> <li>is eq ← seq without the last i bases</li> <li>if a sum 2 ← sum 2 + seq-reads</li> <li>if The last i bases of seq are the same as the first i bases of 3' adapter then</li> <li>if The last i bases of seq are the same as the first i bases of 3' adapter then</li> <li>if The last i bases of seq are the same as the first i bases of 3' adapter then</li> <li>if The last i bases of seq are the same as the first i bases of 3' adapter then</li> <li>if The last i bases of seq are the same as the first j bases of gfp18 OR gfp24 then</li> <li>is unu ← sum 4 + seq-reads</li> <li>else</li> <li>i Sum 1 ← sum 1 + seq-reads</li> <li>i else</li> <li>i sum 1 ← sum 1 + seq-reads</li> <li>i gunp to step 8</li> <li>else</li> <li>i sum 1 ← sum 1 + seq-reads</li> <li>i sum 1 ← sum 1 + seq-reads</li> <li>i for sum 0, sum 1, sum 2, sum 3, sum 4</li> <li>i Cose output files</li> </ul></td><td>5</td><td>sum4</td><td><math>4 \leftarrow 0</math> (reads of GFP);</td></tr> <tr><td>7while not at end of the input file do8Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads9if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then10seq <math>\leftarrow</math> seq without the first two bases.11if The last i bases of seq are the same as the first i bases of 3' adapter then12seq <math>\leftarrow</math> seq without the last i bases13else14sum2 <math>\leftarrow</math> sum2 + seq-reads15Jump to step 816end17if The last i bases of seq are the same as the first i bases of 3' adapter then18sum3 <math>\leftarrow</math> sum3 + seq-reads (self-aliasing 3' adapter has been found)19Jump to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21sum4 <math>\leftarrow</math> sum4 + seq-reads22else23I24Output seq to corresponding output file25else26sum1 <math>\leftarrow</math> sum1 + seq-reads27Jump to step 828end29is um1 <math>\leftarrow</math> sum1 + seq-reads20seq bese21Sum1 <math>\leftarrow</math> sum1 + seq-reads22else33gip on the seq bese34gip on the seq bese35gip on the seq bese36gip on the seq bese37gip on the seq bese38gip on the seq bese39gip on the seq bese30gip on the seq bese31gip on the seq bese32<td< td=""><td>6</td><td>Crea</td><td>te and open output files used to store parsing result</td></td<></td></tr> <tr><td>8Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads9if The first 2 bases of seq is 'AC', 'GT' or 'TG' then10seq <math>\leftarrow</math> seq without the first two bases.11if The last i bases of seq are the same as the first i bases of 3' adapter then12seq <math>\leftarrow</math> seq without the last i bases13else14sum2 <math>\leftarrow</math> sum2 + seq-reads15Jump to step 816end17if The last i bases of seq are the same as the first i bases of 3' adapter then18sum3 <math>\leftarrow</math> sum3 + seq-reads (self-aliasing 3' adapter has been found)19Jump to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21output seq to corresponding output file22else23Jump to step 824else25else26sum1 <math>\leftarrow</math> sum1 + seq-reads27Jump to step 828end29end20seq are the same as the first j bases of gfp18 OR gfp24 then21sum4 <math>\leftarrow</math> sum4 + seq-reads22else23Jump to step 824end25else26sum1 <math>\leftarrow</math> sum1 + seq-reads27Jump to step 828end29end29end20seq bese21sum1 <math>\leftarrow</math> sum2, sum3, sum432Close output files</td><td>7</td><td>whil</td><td>e not at end of the input file do</td></tr> <tr><td>9if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then10<math>seq \leftarrow seq</math> without the first two bases.11if The last i bases of seq are the same as the first i bases of 3' adapter then12<math>seq \leftarrow seq</math> without the last i bases13else14<math>sum2 \leftarrow sum2 + seq-reads</math>15Jump to step 816end17if The last i bases of seq are the same as the first i bases of 3' adapter then18<math>sum3 \leftarrow sum3 + seq-reads</math> (self-aliasing 3' adapter has been found)19Jump to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21<math>sum4 \leftarrow sum4 + seq-reads</math>22else23  Output seq to corresponding output file24end25else26<math>sum1 \leftarrow sum1 + seq-reads</math>27Jump to step 828end29end29other sum1, sum2, sum3, sum430print sum0, sum1, sum2, sum3, sum431Close output files</td><td>8</td><td>R</td><td>tead in one new line from the input file. Denoted the sequence as <i>seq</i> and the read of the sequence as <i>seq-reads</i></td></tr> <tr><td>10seq <math>\leftarrow</math> seq without the first two bases.11if The last i bases of seq are the same as the first i bases of 3' adapter then12  seq <math>\leftarrow</math> seq without the last i bases13else14  sum2 <math>\leftarrow</math> sum2 + seq-reads15  Jump to step 816end17if The last i bases of seq are the same as the first i bases of 3' adapter then18  sum3 <math>\leftarrow</math> sum3 + seq-reads (self-aliasing 3' adapter has been found)19  Jump to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21  sum4 <math>\leftarrow</math> sum4 + seq-reads22else23  Output seq to corresponding output file24end25else26  sum1 <math>\leftarrow</math> sum1 + seq-reads27  Jump to step 828end29end29end30print sum0, sum1, sum2, sum3, sum431Close output files</td><td>9</td><td>if</td><td>The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' <b>then</b></td></tr> <tr><td>11if The last i bases of seq are the same as the first i bases of 3' adapter then12<math> seq \leftarrow seq</math> without the last i bases13<math>else</math>14<math> sum2 \leftarrow sum2 + seq-reads</math>15<math> Jump to step 8</math>16end17if The last i bases of seq are the same as the first i bases of 3' adapter then18<math> sum3 \leftarrow sum3 + seq-reads</math> (self-aliasing 3' adapter has been found)19<math> Jump to step 8</math>20else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21<math> sum4 \leftarrow sum4 + seq-reads</math>22else23<math> Output seq to corresponding output file24end25else26<math> sum1 \leftarrow sum1 + seq-reads</math>27<math> Jump to step 8</math>28end29end29end20end21<math>sum1 \leftarrow sum1 + seq-reads</math>22<math>else</math>23<math> Output seq to step 8</math>24<math>end</math>25else26<math>sum1 \leftarrow sum1 + seq-reads</math>27<math>Jump to step 8</math>28end29end29end30print sum0, sum1, sum2, sum3, sum431Close output files</math></td><td>10</td><td></td><td>seq <math>\leftarrow</math> seq without the first two bases.</td></tr> <tr><td>12<math> </math> seq <math>\leftarrow</math> seq without the last <i>i</i> bases13<math> </math> sum2 <math>\leftarrow</math> sum2 + seq-reads14<math> </math> sum2 <math>\leftarrow</math> sum2 + seq-reads15<math> </math> Jump to step 816end17if The last <i>i</i> bases of seq are the same as the first <i>i</i> bases of 3' adapter then18<math> </math> sum3 <math>\leftarrow</math> sum3 + seq-reads (self-aliasing 3' adapter has been found)19<math> </math> Jump to step 820else if The last <i>j</i> bases of seq are the same as the first <i>j</i> bases of gfp18 OR gfp24 then21<math> </math> sum4 <math>\leftarrow</math> sum4 + seq-reads22else23<math> </math> Output seq to corresponding output file24end25else26sum1 <math>\leftarrow</math> sum1 + seq-reads27Jump to step 828end29end29end30print sum0, sum1, sum2, sum3, sum431Close output files</td><td>11</td><td></td><td><b>if</b> <i>The last i bases of seq are the same as the first i bases of 3' adapter</i> <b>then</b></td></tr> <tr><td>13else14<math>sum2 \leftarrow sum2 + seq-reads</math>15<math>Jump</math> to step 816end17if The last i bases of seq are the same as the first i bases of 3' adapter then18<math>sum3 \leftarrow sum3 + seq-reads</math> (self-aliasing 3' adapter has been found)19<math>Jump</math> to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21<math>sum4 \leftarrow sum4 + seq-reads</math>22else23<math>Output seq to corresponding output file24end25else26<math>sum1 \leftarrow sum1 + seq-reads</math>27<math>Jump to step 8</math>28end29end20ent sum0, sum1, sum2, sum3, sum431Close output files</math></td><td>12</td><td></td><td>seq <math>\leftarrow</math> seq without the last <i>i</i> bases</td></tr> <tr><td>14<math>  </math><math>sum2 \leftarrow sum2 + seq-reads</math>15<math>  </math>Jump to step 816end17if The last i bases of seq are the same as the first i bases of 3' adapter then18<math>  </math><math>sum3 \leftarrow sum3 + seq-reads (self-aliasing 3' adapter has been found)19<math>  </math>Jump to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21<math>  </math><math>sum4 \leftarrow sum4 + seq-reads22else23<math>  </math>Output seq to corresponding output file24<math>  </math> end25else26<math>  </math><math>sum1 \leftarrow sum1 + seq-reads27<math>  </math>Jump to step 828end29end29end29end29end29end29end29end29end29end29end29end29end20gittsum0, sum1, sum2, sum3, sum431Close output files</math></math></math></td><td>13</td><td></td><td>else</td></tr> <tr><td>15       <math> </math> Jump to step 8         16       end         17       if The last i bases of seq are the same as the first i bases of 3' adapter then         18       <math> </math> sum3 <math>\leftarrow</math> sum3 + seq-reads (self-aliasing 3' adapter has been found)         19       <math> </math> Jump to step 8         20       else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then         21       <math> </math> sum4 <math>\leftarrow</math> sum4 + seq-reads         22       else         23       <math> </math> Output seq to corresponding output file         24       end         25       else         26       <math> </math> sum1 <math>\leftarrow</math> sum1 + seq-reads         27       Jump to step 8         28       end         29       end</td><td>14</td><td></td><td><math>sum2 \leftarrow sum2 + seq</math>-reads</td></tr> <tr><td><ul> <li>end</li> <li>if The last i bases of seq are the same as the first i bases of 3' adapter then</li> <li>sum3 ← sum3 + seq-reads (self-aliasing 3' adapter has been found)</li> <li>Jump to step 8</li> <li>else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then</li> <li>sum4 ← sum4 + seq-reads</li> <li>else</li> <li>output seq to corresponding output file</li> <li>end</li> <li>sum1 ← sum1 + seq-reads</li> <li>jump to step 8</li> <li>end</li> <li>end</li> <li>print sum0, sum1, sum2, sum3, sum4</li> <li>Close output files</li> </ul></td><td>15</td><td></td><td>Jump to step 8</td></tr> <tr><td>17if The last i bases of seq are the same as the first i bases of 3' adapter then18<math>sum3 \leftarrow sum3 + seq</math>-reads (self-aliasing 3' adapter has been found)19Jump to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21<math>sum4 \leftarrow sum4 + seq</math>-reads22else23 24end25else26<math>sum1 \leftarrow sum1 + seq</math>-reads27Jump to step 828end29end20end20<td< td=""><td>16</td><td></td><td>end</td></td<></td></tr> <tr><td>18       <math>sum3 \leftarrow sum3 + seq</math>-reads (self-aliasing 3' adapter has been found)         19       Jump to step 8         20       else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then         21       sum4 <math>\leftarrow</math> sum4 + seq-reads         22       else         23       Output seq to corresponding output file         24       end         25       else         26       sum1 <math>\leftarrow</math> sum1 + seq-reads         27       Jump to step 8         28       end         29       end         30       print sum0, sum1, sum2, sum3, sum4         31       Close output files</td><td>17</td><td></td><td><b>if</b> <i>The last i bases of seq are the same as the first i bases of 3' adapter</i> <b>then</b></td></tr> <tr><td><ul> <li>Image: Jump to step 8</li> <li>else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then</li> <li>sum4 ← sum4 + seq-reads</li> <li>else</li> <li>Output seq to corresponding output file</li> <li>end</li> <li>else</li> <li>sum1 ← sum1 + seq-reads</li> <li>Jump to step 8</li> <li>end</li> <li>end</li> <li>oprint sum0, sum1, sum2, sum3, sum4</li> <li>Close output files</li> </ul></td><td>18</td><td></td><td>sum3 <math>\leftarrow</math> sum3 + seq-reads (self-aliasing 3' adapter has been found)</td></tr> <tr><td>20else if The last j bases of seq are the same as the first j bases of <math>gfp18</math> OR <math>gfp24</math> then21<math>sum4 \leftarrow sum4 + seq-reads</math>22else23<math> </math> Output seq to corresponding output file24end25else26<math>sum1 \leftarrow sum1 + seq-reads</math>27Jump to step 828end29end30print sum0, sum1, sum2, sum3, sum431Close output files</td><td>19</td><td></td><td>Jump to step 8</td></tr> <tr><td>21<math> </math> sum4 <math>\leftarrow</math> sum4 + seq-reads22<math> </math> else23<math> </math> Output seq to corresponding output file24<math> </math> end25<math> </math> sum1 <math>\leftarrow</math> sum1 + seq-reads26<math> </math> sum1 <math>\leftarrow</math> sum1 + seq-reads27<math> </math> Jump to step 828end29end30print sum0, sum1, sum2, sum3, sum431Close output files</td><td>20</td><td></td><td>else if The last j bases of seq are the same as the first j bases of <math>gfp18</math> OR <math>gfp24</math> then</td></tr> <tr><td><ul> <li>22   else</li> <li>23   Output seq to corresponding output file</li> <li>24   end</li> <li>25   else</li> <li>26   sum1 ~ sum1 + seq-reads</li> <li>27   Jump to step 8</li> <li>28   end</li> <li>29 end</li> <li>30 print sum0, sum1, sum2, sum3, sum4</li> <li>31 Close output files</li> </ul></td><td>21</td><td></td><td><math>sum4 \leftarrow sum4 + seq</math>-reads</td></tr> <tr><td>23<math> </math>Output seq to corresponding output file24end25else26<math>sum1 \leftarrow sum1 + seq-reads</math>27Jump to step 828end29end30print sum0, sum1, sum2, sum3, sum431Close output files</td><td>22</td><td></td><td>else</td></tr> <tr><td>24<math> </math> end25else26<math> </math> sum1 <math>\leftarrow</math> sum1 + seq-reads27<math> </math> Jump to step 828end29end30print sum0, sum1, sum2, sum3, sum431Close output files</td><td>23</td><td></td><td>Output seq to corresponding output file</td></tr> <tr><td>25else26<math>  sum1 \leftarrow sum1 + seq-reads</math>27<math>  Jump to step 8</math>28end29end30print sum0, sum1, sum2, sum3, sum431Close output files</td><td>24</td><td></td><td>end</td></tr> <tr><td><ul> <li>26   sum1 ← sum1 + seq-reads</li> <li>27   Jump to step 8</li> <li>28   end</li> <li>29 end</li> <li>30 print sum0, sum1, sum2, sum3, sum4</li> <li>31 Close output files</li> </ul></td><td>25</td><td>e</td><td>lse</td></tr> <tr><td><ul> <li>27 Jump to step 8</li> <li>28 end</li> <li>29 end</li> <li>30 print sum0, sum1, sum2, sum3, sum4</li> <li>31 Close output files</li> </ul></td><td>26</td><td></td><td><math>sum1 \leftarrow sum1 + seq</math>-reads</td></tr> <tr><td><ul> <li>end</li> <li>end</li> <li>print sum0, sum1, sum2, sum3, sum4</li> <li>Close output files</li> </ul></td><td>27</td><td></td><td>Jump to step 8</td></tr> <tr><td><ul> <li>29 end</li> <li>30 print sum0, sum1, sum2, sum3, sum4</li> <li>31 Close output files</li> </ul></td><td>28</td><td colspan="6">28 end</td></tr> <tr><td>30 print sum0, sum1, sum2, sum3, sum4 31 Close output files</td><td>29</td><td>end</td><td></td></tr> <tr><td>31 Close output files</td><td>30</td><td colspan="5">30 print sum0, sum1, sum2, sum3, sum4</td></tr> <tr><td>1</td><td>31</td></tr>
   
  | Resu | It: Parse all sequences in the sequencing result file and output parsing results into output files |   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| <pre>sum0 0 (reads of useful sequences);<br/>sum1 0 (reads of sequences without 3' adapter match);<br/>sum2 0 (reads of sequences without 3' adapter match);<br/>sum3 0 (reads of sequences without 3' adapter match);<br/>sum4 0 (reads of sequences without 3' adapter match);<br/>create and open output files used to store parsing result<br/>while not at end of the input file do<br/>Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads<br/>if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then<br/>seq ← seq without the first two bases.<br/>if The last i bases of seq are the same as the first i bases of 3' adapter then<br/>seq ← seq without the last i bases<br/>else<br/>else<br/>if The last i bases of seq are the same as the first i bases of 3' adapter then<br/>sum3 ← sum3 + seq-reads<br/>figure to sum2 + seq-reads<br/>sum3 ← sum3 + seq-reads (self-aliasing 3' adapter then<br/>sum4 ← sum4 + seq-reads<br/>else<br/>else<br/>i sum4 ← sum4 + seq-reads<br/>i sum4 ← sum4 + seq-reads<br/>i sum4 ← sum4 + seq-reads<br/>i else<br/>i sum1 ← sum1 + seq-reads<br/>i sum1 ← sum1 + seq-reads<br/>i sum1 ← sum4 + seq-reads<br/>i sum4 ← sum4 + seq-reads<br/>i su</pre>  
   
  |      | Outp   | out: Output files and statistics on the screen  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
                |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   |   |      |   |  
  |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |  |    |  |  |   |    |  |                |   
                           |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 2 sum1 ← 0 ( reads of sequences without 5' adapter match);<br>3 sum2 ← 0 ( reads of sequences without 3' adapter match);<br>4 sum3 ← 0 ( reads of sequences with self-ligated sequences);<br>5 sum4 ← 0 ( reads of GFP);<br>6 Create and open output files used to store parsing result<br>7 while not at end of the input file do<br>8 Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads<br>9 if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then<br>10   seq ← seq without the first two bases.<br>11   if The last i bases of seq are the same as the first i bases of 3' adapter then<br>12   seq ← seq without the last i bases<br>13   else<br>14   sum2 ← sum2 + seq-reads<br>15   Jump to step 8<br>16   end<br>17   if The last i bases of seq are the same as the first i bases of 3' adapter then<br>18   sum3 ← sum3 + seq-reads (self-aliasing 3' adapter has been found)<br>19   Jump to step 8<br>20   else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then<br>21   sum4 ← sum4 + seq-reads<br>22   oUtput seq to corresponding output file<br>23   else<br>24   oUtput seq to corresponding output file<br>24   end<br>25   else<br>26   sum1 ← sum1 + seq-reads<br>27   Jump to step 8<br>28   end<br>29   output seq to step 8<br>29   end<br>20   output seq to step 8<br>20   output seq to step 8<br>21   output seq to step 8<br>22   end<br>23   output seq to step 8<br>24   output seq to step 8<br>25   end<br>26   sum1 ← sum1 + seq-reads<br>27   Jump to step 8<br>28   end<br>29   output seq to step 8<br>29   end<br>20   output seq to step 8<br>20   end<br>21   output seq to step 8<br>22   end<br>32   output seq to step 8<br>33   output seq to step 8<br>34   output seq to step 8<br>35   output seq to step 8<br>36   end<br>39   output seq to step 8<br>30   output seq to step 8<br>31   output seq to step 8<br>32   end<br>33   output seq to step 8<br>34   output seq to step 8<br>35   output seq to step 8<br>35   output seq to step 8<br>36   output seq to step 8<br>37   output seq to step 8<br>38   output seq to step 8<br>39   output seq to step 8<br>30   output seq to step 8<br>31   
   
  | 1    | sum(   | $0 \leftarrow 0$ (reads of useful sequences);   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| a sum2 ← 0 ( reads of sequences without 3' adapter match);<br>4 sum3 ← 0 ( reads of sequences with self-ligated sequences);<br>5 sum4 ← 0 ( reads of GFP);<br>6 Create and open output files used to store parsing result<br>7 while not at end of the input file do<br>8 Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads<br>9 if The first 2 bases of seq is 'AC', 'CA' or 'TG' then<br>1 seq ← seq without the first two bases.<br>1 if The last i bases of seq are the same as the first i bases of 3' adapter then<br>1 seq ← seq without the last i bases<br>1 else<br>1 sum2 ← sum2 + seq-reads<br>1 jump to step 8<br>1 else<br>1 if The last i bases of seq are the same as the first i bases of 3' adapter then<br>1 sum3 ← sum3 + seq-reads (self-aliasing 3' adapter thas been found)<br>1 jump to step 8<br>2 else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then<br>2 sum4 ← sum4 + seq-reads<br>2 else<br>2 l Output seq to corresponding output file<br>2 else<br>2 l Output seq to corresponding output file<br>2 else<br>3 sum4 ← sum1 + seq-reads<br>3 ljump to step 8<br>2 else<br>4 sum1 ← sum1 + seq-reads<br>2 else<br>4 sum1 ← sum1 + seq-reads<br>3 sum3 ← sum3 + seq-reads<br>4 sum3 ← sum4 ← sum4 + seq-reads<br>4 sum3 ← sum4 + seq-reads<br>5 sum3 ← sum3 + seq-reads<br>5 sum3 ← sum4 + seq-reads<br>5 sum3 ← sum4 ← sum4 + seq-reads<br>5 sum3 ← sum4 ← sum4 + seq-reads<br>5 sum3 ← sum3 ← sum3 ← sum3 ← sum3 ← sum4   
   
  | 2    | sum1   | $1 \leftarrow 0$ (reads of sequences without 5' adapter match);   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |  |    |  |  |  
  |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 4 sum3 ← 0 ( reads of sequences with self-ligated sequences);<br>5 sum4 ← 0 ( reads of GFP);<br>6 Create and open output files used to store parsing result<br>7 while not at end of the input file do<br>8 Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads<br>9 if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then<br>10 seq seq without the first two bases.<br>11 if The last i bases of seq are the same as the first i bases of 3' adapter then<br>12 seq seq without the last i bases<br>13 else<br>14 sum2 ← sum2 + seq-reads<br>15 Jump to step 8<br>16 end<br>17 if The last i bases of seq are the same as the first i bases of 3' adapter then<br>18 sum3 ← sum3 + seq-reads (self-aliasing 3' adapter has been found)<br>19 Jump to step 8<br>20 else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then<br>21 sum4 ← sum4 + seq-reads<br>22 else<br>23 else<br>24 lese if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then<br>24 sequence sum4 + seq-reads<br>25 else<br>26 sequence sum4 + seq-reads<br>27 sum4 ← sum1 + seq-reads<br>28 else<br>29 end<br>30 print sum0, sum1, sum2, sum3, sum4 a<br>30 Close output files   
   
  | 3    | sum2   | $2 \leftarrow 0$ (reads of sequences without 3' adapter match);   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| s sum4 ← 0 ( reads of GFP);<br>6 Create and open output files used to store parsing result<br>7 while not at end of the input file do<br>8 Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads<br>9 if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then<br>10 $ seq - seq$ without the first two bases.<br>11 if The last i bases of seq are the same as the first i bases of 3' adapter then<br>12 $ seq - seq$ without the last i bases<br>13 else<br>14 $ sum2 \leftarrow sum2 + seq-reads$<br>15 $ Jump to step 8$<br>16 end<br>17 if The last i bases of seq are the same as the first i bases of 3' adapter then<br>18 $ sum3 \leftarrow sum3 + seq-reads$ (self-aliasing 3' adapter thas been found)<br>19 $ Jump to step 8$<br>20 else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then<br>21 $ sum4 \leftarrow sum4 + seq-reads$<br>22 $ else$<br>23 $ Output seq to corresponding output file 24  end25  else26  sum1 \leftarrow sum1 + seq-reads27  Jump to step 828  end29 end30 print sum0, sum1, sum2, sum3, sum431  Close output files$   
   
  | 4    | sum?   | $3 \leftarrow 0$ (reads of sequences with self-ligated sequences);  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| <ul> <li><sup>6</sup> Create and open output files used to store parsing result</li> <li>7 while not at end of the input file do</li> <li>8 Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads</li> <li>if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then</li> <li>is eq ← seq without the first two bases.</li> <li>if The last i bases of seq are the same as the first i bases of 3' adapter then</li> <li>is eq ← seq without the last i bases</li> <li>if a sum 2 ← sum 2 + seq-reads</li> <li>if The last i bases of seq are the same as the first i bases of 3' adapter then</li> <li>if The last i bases of seq are the same as the first i bases of 3' adapter then</li> <li>if The last i bases of seq are the same as the first i bases of 3' adapter then</li> <li>if The last i bases of seq are the same as the first i bases of 3' adapter then</li> <li>if The last i bases of seq are the same as the first j bases of gfp18 OR gfp24 then</li> <li>is unu ← sum 4 + seq-reads</li> <li>else</li> <li>i Sum 1 ← sum 1 + seq-reads</li> <li>i else</li> <li>i sum 1 ← sum 1 + seq-reads</li> <li>i gunp to step 8</li> <li>else</li> <li>i sum 1 ← sum 1 + seq-reads</li> <li>i sum 1 ← sum 1 + seq-reads</li> <li>i for sum 0, sum 1, sum 2, sum 3, sum 4</li> <li>i Cose output files</li> </ul>   
   
  | 5    | sum4   | $4 \leftarrow 0$ (reads of GFP);  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 7while not at end of the input file do8Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads9if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then10seq $\leftarrow$ seq without the first two bases.11if The last i bases of seq are the same as the first i bases of 3' adapter then12seq $\leftarrow$ seq without the last i bases13else14sum2 $\leftarrow$ sum2 + seq-reads15Jump to step 816end17if The last i bases of seq are the same as the first i bases of 3' adapter then18sum3 $\leftarrow$ sum3 + seq-reads (self-aliasing 3' adapter has been found)19Jump to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21sum4 $\leftarrow$ sum4 + seq-reads22else23I24Output seq to corresponding output file25else26sum1 $\leftarrow$ sum1 + seq-reads27Jump to step 828end29is um1 $\leftarrow$ sum1 + seq-reads20seq bese21Sum1 $\leftarrow$ sum1 + seq-reads22else33gip on the seq bese34gip on the seq bese35gip on the seq bese36gip on the seq bese37gip on the seq bese38gip on the seq bese39gip on the seq bese30gip on the seq bese31gip on the seq bese32 <td< td=""><td>6</td><td>Crea</td><td>te and open output files used to store parsing result</td></td<>  
   
  | 6    | Crea   | te and open output files used to store parsing result   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 8Read in one new line from the input file. Denoted the sequence as seq and the read of the sequence as seq-reads9if The first 2 bases of seq is 'AC', 'GT' or 'TG' then10seq $\leftarrow$ seq without the first two bases.11if The last i bases of seq are the same as the first i bases of 3' adapter then12seq $\leftarrow$ seq without the last i bases13else14sum2 $\leftarrow$ sum2 + seq-reads15Jump to step 816end17if The last i bases of seq are the same as the first i bases of 3' adapter then18sum3 $\leftarrow$ sum3 + seq-reads (self-aliasing 3' adapter has been found)19Jump to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21output seq to corresponding output file22else23Jump to step 824else25else26sum1 $\leftarrow$ sum1 + seq-reads27Jump to step 828end29end20seq are the same as the first j bases of gfp18 OR gfp24 then21sum4 $\leftarrow$ sum4 + seq-reads22else23Jump to step 824end25else26sum1 $\leftarrow$ sum1 + seq-reads27Jump to step 828end29end29end20seq bese21sum1 $\leftarrow$ sum2, sum3, sum432Close output files  
   
  | 7    | whil   | e not at end of the input file do   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 9if The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' then10 $seq \leftarrow seq$ without the first two bases.11if The last i bases of seq are the same as the first i bases of 3' adapter then12 $seq \leftarrow seq$ without the last i bases13else14 $sum2 \leftarrow sum2 + seq-reads$ 15Jump to step 816end17if The last i bases of seq are the same as the first i bases of 3' adapter then18 $sum3 \leftarrow sum3 + seq-reads$ (self-aliasing 3' adapter has been found)19Jump to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21 $sum4 \leftarrow sum4 + seq-reads$ 22else23  Output seq to corresponding output file24end25else26 $sum1 \leftarrow sum1 + seq-reads$ 27Jump to step 828end29end29other sum1, sum2, sum3, sum430print sum0, sum1, sum2, sum3, sum431Close output files  
   
  | 8    | R  | tead in one new line from the input file. Denoted the sequence as <i>seq</i> and the read of the sequence as <i>seq-reads</i> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 10seq $\leftarrow$ seq without the first two bases.11if The last i bases of seq are the same as the first i bases of 3' adapter then12  seq $\leftarrow$ seq without the last i bases13else14  sum2 $\leftarrow$ sum2 + seq-reads15  Jump to step 816end17if The last i bases of seq are the same as the first i bases of 3' adapter then18  sum3 $\leftarrow$ sum3 + seq-reads (self-aliasing 3' adapter has been found)19  Jump to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21  sum4 $\leftarrow$ sum4 + seq-reads22else23  Output seq to corresponding output file24end25else26  sum1 $\leftarrow$ sum1 + seq-reads27  Jump to step 828end29end29end30print sum0, sum1, sum2, sum3, sum431Close output files   
   
  | 9    | if   | The first 2 bases of seq is 'AC', 'CA', 'GT' or 'TG' <b>then</b>  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 11if The last i bases of seq are the same as the first i bases of 3' adapter then12 $ seq \leftarrow seq$ without the last i bases13 $else$ 14 $ sum2 \leftarrow sum2 + seq-reads$ 15 $ Jump to step 8$ 16end17if The last i bases of seq are the same as the first i bases of 3' adapter then18 $ sum3 \leftarrow sum3 + seq-reads$ (self-aliasing 3' adapter has been found)19 $ Jump to step 8$ 20else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21 $ sum4 \leftarrow sum4 + seq-reads$ 22else23 $ Output seq to corresponding output file24end25else26 sum1 \leftarrow sum1 + seq-reads27 Jump to step 828end29end29end20end21sum1 \leftarrow sum1 + seq-reads22else23 Output seq to step 824end25else26sum1 \leftarrow sum1 + seq-reads27Jump to step 828end29end29end30print sum0, sum1, sum2, sum3, sum431Close output files$  
   
  | 10   |  | seq $\leftarrow$ seq without the first two bases.   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 12 $ $ seq $\leftarrow$ seq without the last <i>i</i> bases13 $ $ sum2 $\leftarrow$ sum2 + seq-reads14 $ $ sum2 $\leftarrow$ sum2 + seq-reads15 $ $ Jump to step 816end17if The last <i>i</i> bases of seq are the same as the first <i>i</i> bases of 3' adapter then18 $ $ sum3 $\leftarrow$ sum3 + seq-reads (self-aliasing 3' adapter has been found)19 $ $ Jump to step 820else if The last <i>j</i> bases of seq are the same as the first <i>j</i> bases of gfp18 OR gfp24 then21 $ $ sum4 $\leftarrow$ sum4 + seq-reads22else23 $ $ Output seq to corresponding output file24end25else26sum1 $\leftarrow$ sum1 + seq-reads27Jump to step 828end29end29end30print sum0, sum1, sum2, sum3, sum431Close output files  
   
  | 11   |  | <b>if</b> <i>The last i bases of seq are the same as the first i bases of 3' adapter</i> <b>then</b>                          | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 13else14 $sum2 \leftarrow sum2 + seq-reads$ 15 $Jump$ to step 816end17if The last i bases of seq are the same as the first i bases of 3' adapter then18 $sum3 \leftarrow sum3 + seq-reads$ (self-aliasing 3' adapter has been found)19 $Jump$ to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21 $sum4 \leftarrow sum4 + seq-reads$ 22else23 $Output seq to corresponding output file24end25else26sum1 \leftarrow sum1 + seq-reads27Jump to step 828end29end20ent sum0, sum1, sum2, sum3, sum431Close output files$  
   
  | 12   |  | seq $\leftarrow$ seq without the last <i>i</i> bases  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 14 $  $ $sum2 \leftarrow sum2 + seq-reads$ 15 $  $ Jump to step 816end17if The last i bases of seq are the same as the first i bases of 3' adapter then18 $  $ $sum3 \leftarrow sum3 + seq-reads (self-aliasing 3' adapter has been found)19  Jump to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21  sum4 \leftarrow sum4 + seq-reads22else23  Output seq to corresponding output file24   end25else26  sum1 \leftarrow sum1 + seq-reads27  Jump to step 828end29end29end29end29end29end29end29end29end29end29end29end29end20gittsum0, sum1, sum2, sum3, sum431Close output files$  
   
  | 13   |  | else  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 15 $ $ Jump to step 8         16       end         17       if The last i bases of seq are the same as the first i bases of 3' adapter then         18 $ $ sum3 $\leftarrow$ sum3 + seq-reads (self-aliasing 3' adapter has been found)         19 $ $ Jump to step 8         20       else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then         21 $ $ sum4 $\leftarrow$ sum4 + seq-reads         22       else         23 $ $ Output seq to corresponding output file         24       end         25       else         26 $ $ sum1 $\leftarrow$ sum1 + seq-reads         27       Jump to step 8         28       end         29       end  
   
   | 14   |  | $sum2 \leftarrow sum2 + seq$ -reads   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
   |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |  
  |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |   
  |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| <ul> <li>end</li> <li>if The last i bases of seq are the same as the first i bases of 3' adapter then</li> <li>sum3 ← sum3 + seq-reads (self-aliasing 3' adapter has been found)</li> <li>Jump to step 8</li> <li>else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then</li> <li>sum4 ← sum4 + seq-reads</li> <li>else</li> <li>output seq to corresponding output file</li> <li>end</li> <li>sum1 ← sum1 + seq-reads</li> <li>jump to step 8</li> <li>end</li> <li>end</li> <li>print sum0, sum1, sum2, sum3, sum4</li> <li>Close output files</li> </ul>  
   
  | 15   |  | Jump to step 8  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 17if The last i bases of seq are the same as the first i bases of 3' adapter then18 $sum3 \leftarrow sum3 + seq$ -reads (self-aliasing 3' adapter has been found)19Jump to step 820else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then21 $sum4 \leftarrow sum4 + seq$ -reads22else23 24end25else26 $sum1 \leftarrow sum1 + seq$ -reads27Jump to step 828end29end20end20 <td< td=""><td>16</td><td></td><td>end</td></td<>   
   
  | 16   |  | end   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 18 $sum3 \leftarrow sum3 + seq$ -reads (self-aliasing 3' adapter has been found)         19       Jump to step 8         20       else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then         21       sum4 $\leftarrow$ sum4 + seq-reads         22       else         23       Output seq to corresponding output file         24       end         25       else         26       sum1 $\leftarrow$ sum1 + seq-reads         27       Jump to step 8         28       end         29       end         30       print sum0, sum1, sum2, sum3, sum4         31       Close output files   
   
  | 17   |  | <b>if</b> <i>The last i bases of seq are the same as the first i bases of 3' adapter</i> <b>then</b>                          | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |  
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| <ul> <li>Image: Jump to step 8</li> <li>else if The last j bases of seq are the same as the first j bases of gfp18 OR gfp24 then</li> <li>sum4 ← sum4 + seq-reads</li> <li>else</li> <li>Output seq to corresponding output file</li> <li>end</li> <li>else</li> <li>sum1 ← sum1 + seq-reads</li> <li>Jump to step 8</li> <li>end</li> <li>end</li> <li>oprint sum0, sum1, sum2, sum3, sum4</li> <li>Close output files</li> </ul>   
   
  | 18   |  | sum3 $\leftarrow$ sum3 + seq-reads (self-aliasing 3' adapter has been found)  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 20else if The last j bases of seq are the same as the first j bases of $gfp18$ OR $gfp24$ then21 $sum4 \leftarrow sum4 + seq-reads$ 22else23 $ $ Output seq to corresponding output file24end25else26 $sum1 \leftarrow sum1 + seq-reads$ 27Jump to step 828end29end30print sum0, sum1, sum2, sum3, sum431Close output files  
   
  | 19   |  | Jump to step 8  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 21 $ $ sum4 $\leftarrow$ sum4 + seq-reads22 $ $ else23 $ $ Output seq to corresponding output file24 $ $ end25 $ $ sum1 $\leftarrow$ sum1 + seq-reads26 $ $ sum1 $\leftarrow$ sum1 + seq-reads27 $ $ Jump to step 828end29end30print sum0, sum1, sum2, sum3, sum431Close output files  
   
  | 20   |  | else if The last j bases of seq are the same as the first j bases of $gfp18$ OR $gfp24$ then                                  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| <ul> <li>22   else</li> <li>23   Output seq to corresponding output file</li> <li>24   end</li> <li>25   else</li> <li>26   sum1 ~ sum1 + seq-reads</li> <li>27   Jump to step 8</li> <li>28   end</li> <li>29 end</li> <li>30 print sum0, sum1, sum2, sum3, sum4</li> <li>31 Close output files</li> </ul>  
   
  | 21   |  | $sum4 \leftarrow sum4 + seq$ -reads   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 23 $ $ Output seq to corresponding output file24end25else26 $sum1 \leftarrow sum1 + seq-reads$ 27Jump to step 828end29end30print sum0, sum1, sum2, sum3, sum431Close output files  
   
  | 22   |  | else  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 24 $ $ end25else26 $ $ sum1 $\leftarrow$ sum1 + seq-reads27 $ $ Jump to step 828end29end30print sum0, sum1, sum2, sum3, sum431Close output files   
   
  | 23   |  | Output seq to corresponding output file   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 25else26 $  sum1 \leftarrow sum1 + seq-reads$ 27 $  Jump to step 8$ 28end29end30print sum0, sum1, sum2, sum3, sum431Close output files   
   
  | 24   |  | end   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| <ul> <li>26   sum1 ← sum1 + seq-reads</li> <li>27   Jump to step 8</li> <li>28   end</li> <li>29 end</li> <li>30 print sum0, sum1, sum2, sum3, sum4</li> <li>31 Close output files</li> </ul>  
   
  | 25   | e  | lse   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| <ul> <li>27 Jump to step 8</li> <li>28 end</li> <li>29 end</li> <li>30 print sum0, sum1, sum2, sum3, sum4</li> <li>31 Close output files</li> </ul>  
   
  | 26   |  | $sum1 \leftarrow sum1 + seq$ -reads   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| <ul> <li>end</li> <li>end</li> <li>print sum0, sum1, sum2, sum3, sum4</li> <li>Close output files</li> </ul>   
   
  | 27   |  | Jump to step 8  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| <ul> <li>29 end</li> <li>30 print sum0, sum1, sum2, sum3, sum4</li> <li>31 Close output files</li> </ul>   
   
  | 28   | 28 end   |   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 30 print sum0, sum1, sum2, sum3, sum4<br>31 Close output files   
   
  | 29   | end  |   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 31 Close output files  
   
  | 30   | 30 print sum0, sum1, sum2, sum3, sum4  |   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |
| 1  
   
  | 31   |  |   | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
  |  |      |  |  |   |      |   |  |   |      |   |  |   |      |   |  |   |      |  |  |   |      |                                  |   
   |   |      |   |   |   |      |                                   |   |   |   |   |  |   |    |  |   |    |  |   |   |    |  |  |   |    |  |  |   |    |  |      |  |    |  |                                     |   |    |  |                |  |    |  |     |  |    |  |  |                
   |    |  |  |   |    |  |                |   |    |  |  |   |    |  |                                     |   |    |  |      |  |    |  |   |  |    |  |     |   |    |   |     |   |    |  |                                     |  |    |  |                |  |    |        |  |  |  |  |  |  |    |     |  |                       |    |                                       |  |  |  |  |   |    |

#### Source code for CLASSIFY:

```
#This software is a free software licensed under GNU General Public License version 3 or later.
# Cite this program as: Xiaopeng Qi, Forrest Sheng Bao and Zhixin Xie,
# Small RNA Deep Sequencing Reveals Role for Arabidopsis thaliana RNA-dependent
# RNA Polymerases in Viral siRNA Biogenesis, PLoS One, xxx: xxx, 2009
#usage: python classify.py <pool>
#no shell program embedding it
import string,os
from os.path import *
from numpy import *
from pylab import *
def classification(samples):
        #samples
        #[0]: segments
        #[1]: reads
        #[2]: class
        trim5, trim3, result = [],[],[]
        if samples[1][:2]=='AC':
                trim5=[samples[1][2:], int(samples[0]), 1]
        elif samples[1][:2] == 'CA':
                trim5=[samples[1][2:], int(samples[0]), 2]
        elif samples[1][:2]=='GT':
                trim5=[samples[1][2:], int(samples[0]), 3]
        elif samples[1][:2]=='TG':
                trim5=[samples[1][2:], int(samples[0]), 4]
        else:
                return -1; # no 5' adaptor
        tail = "TCGTATGCCGTCTTCTGCTTG";
        windowL = 3; # initial slide window length
        windowU = len(tail); # slide window length to determine dumping a seq to trash3
        subject = trim5[0];
        should = 0;
        for j in xrange (windowL, windowU+1): #begin sliding from $windowL, don't exceed $windowU
                if subject[-j:] == tail[:j]:
                        trim3=[subject[:-j],trim5[1],trim5[2]];
                        should = 1;
                        if len(subject) > 34:
                                print subject, len(subject)
                        break;
                if should ==1:
                        print "if not break, let me know";
        if should == 0:
                return -2; #no 3' adapter
        match = 'TCGTATGCCGTCTTCTGCTTGTT';
        gfp18 = 'CATCCTATACGGCCACAA';
        gfp24 = 'TTGTGGCCGAGGATGTTTCCGTCC';
        if match.find(trim3[0]) == -1 and gfp18.find(trim3[0]) == -1 and gfp24.find(trim3[0]) == -1:
                return trim3;
        elif match.find(trim3[0])>=0:
```

```
return -3; # self-ligation
        elif qfp18.find(trim3[0])>=0 or qfp24.find(trim3[0])>=0:
                return -4; # GFP
        else:
                print match.find(trim3[i][0]);
        return 0; # no meaningful result
sum0, sum1, sum2, sum3, sum4 = 0, 0, 0, 0, 0
out = [];
for i in xrange(0,4):
        out.append([])
        for j in xrange(15,29):
                storefile = "./classified/s_6-A"+str(i)+"-"+str(j)+".txt"
                if (exists(storefile)):
                        os.remove(storefile)
                out[i].append(open(storefile, 'a'))
f1=open(sys.argv[1],'r')
samples=[]
totalseq = 0;
while True:
        line = f1.readline()
        if len(line) == 0:
                break #EOF
        totalseq+=1;
        result = classification(line.split())
        if result==-1:
                sum1+=int(line.split()[0])
        elif result==-2:
                sum2+=int(line.split()[0])
        elif result==-3:
                sum3+=int(line.split()[0])
        elif result==-4:
                sum4+=int(line.split()[0])
        elif result==0:
                break;
        else:
                sum0+=int(line.split()[0])
                Ax=result[2]-1
                if len(result[0])<16:</pre>
                        box = 15 - 15
                else:
                        box = len(result[0])-15
                out[Ax][box].write(result[0]+'\t'+str(result[1])+'\t'+str(result[2])+'\n')
print "reads of useful sequences", sum0, double(sum0)*100/(sum0+sum1+sum2+sum3+sum4), "%";
print "reads of sequences without 5' adapter match:", sum1, \
double(sum1)*100/(sum0+sum1+sum2+sum3+sum4),"%";
print "reads of sequences without 3' adapter match:",sum2,\
double(sum2)*100/(sum0+sum1+sum2+sum3+sum4),"%";
print "reads of sequences self-aliasing 3' adapter:", sum3, \
double(sum3)*100/(sum0+sum1+sum2+sum3+sum4),"%";
print "reads of GFP", sum4, double(sum4)*100/(sum0+sum1+sum2+sum3+sum4), "%";
print "reads of sequences:", sum0+sum1+sum2+sum3+sum4;
for i in xrange(0,3):
        for j in xrange(15,29):
                out[i][j-15].close()
```

# "MAP"

The input file is the parsing result file in text (ASCII) format. The first column is the sequence. The second column is the reads. The last column is small RNA library Id. Below is a segment of an example input file:

TTTGGATTGAAGGGAGCTCTA	53996	2
TTTGGATTGAAGGGAGCTCTT	10708	2
TCGGACCAGGCTTCATTCCCC	7441 2	2

### *Pseudocode* for **MAP**:

Input: A parsing result file and a genome file in FASTA format				
<b>Result</b> : Find siRNAs in the genome with perfect match and return their locations				
Output: One output file recording searching results and two additional files for plotting				
1 Create and open output files.				
2 Load the genome sequence into the memory.				
3 while not at end of the parsing result file do				
4 Read in one new line from the parsing result file. Denoted the sequence as <i>seq</i> and the read of the sequence as				
seq-reads				
5 if 18 < length(seq) < 25 then				
6 find all occurrence of <i>seq</i> on the genome				
7 output all occurence to the file recording searching results				
8 <b>Repeat</b> steps 6 and 7 to the complementary strand of the genome				
9 Output data for visualizing those occurences				
io end				
11 Close all files				

#### *source code* for **MAP**:

```
#This software is a free software licensed under GNU General Public License version 3 or later.
# Cite this program as: Xiaopeng Qi, Forrest Sheng Bao and Zhixin Xie,
# Small RNA Deep Sequencing Reveals Role for Arabidopsis thaliana RNA-dependent
# RNA Polymerases in Viral siRNA Biogenesis, PLoS One, xxx: xxx, 2009
# run it in "classified" folder ; python ../map.py s_6_tag-A3.txt ../tmvcg
import string,os
from os.path import *
from numpy import *
from pylab import *
import cPickle
def fastaread(filename):
        print "Begin Read"
#
        seq0 = [];
        for line in filename.readlines():
                seq0.append(line.split());
        seq0 = seq0[1:];
        seq='';
        for i in xrange(0,len(seq0)):
                seq = seq + seq0[i][0];
        del seq0
        print "Read done"
#
        return seq;
#f1=open("./results/"+sys.argv[1], 'r')
f1=open(sys.argv[1], 'r')
pool=[]
totalseq = 0;
for line in f1.readlines():
        pool.append(line.split());
f1.close()
#print pool
for i in xrange(0,len(pool)):
        pool[i][1] = int(pool[i][1])
        pool[i].append([])
genomename = sys.argv[2]
f2=open (genomename, 'r')
genome = fastaread(f2)
f2.close()
genome = genome.upper();
#print pool
#print genome
#samples
#[0]: segments
#[1]: reads
#[2]: class
#[3]: position on mapped genome [location, S/AS]
#[4]: section # in 9 sections, from 0 to 8
# this function is changed in the new version
def addone(target, position, reads, direction):
```

```
if target.get(position-1, 0) == 0:#padzero
                target[position-1] = 0
        if target.get(position+1, 0) == 0:#padzero
                target[position+1] = 0
#
        target[position]=reads*direction
        target[position] = target.get(position, 0) + reads*direction
# end
def comp(genome):
        compi = { 'A': 'T', 'T': 'A', 'G': 'C', 'C': 'G' }
        genomecom = list(genome)
        for i in xrange(0,len(genome)):
                genomecom[i] = compi.get(genome[i], '?')
        return ''.join(genomecom)
        """ find a gen in pool in genome via direction"""
def plotcount (genome, pool, direction):
        genomeplot = \{\}
        for i in xrange(0,len(pool)):
#
                print "seq", i, "/", len(pool), "->", direction, sys.argv[1]
                segment = pool[i][0];
                if len(segment)>=18 and len(segment)<=25: # control the segment length
                        delimiter = 0;
                        while delimiter != -1:
                                 if direction ==1:
                                         location = genome.find(segment, delimiter, len(genome));
                                         ####just once, we need to consider all conditions.
                                 elif direction == -1:
                                         location = genome.find(segment[::-1], delimiter, len(genome))
                                         ####just once, we need to consider
                                 if location >= 0:
                                         addone(genomeplot, location, pool[i][1], direction)
# no need to zero pad if using stem plot
                                         if genomeplot.get(location-1, 0) == 0:#padzero
#
#
                                                 genomeplot[location-1] = 0
                                         if genomeplot.get(location+1, 0) == 0:#padzero
#
                                                 genomeplot[location+1] = 0
# end of zero pad
                                         genomeplot[location] = genomeplot.get(location, 0)\
                                          + pool[i][1]*direction
                                         pool[i][3].append([location, direction])
                                         delimiter = location + len(pool[i][0]);
                                 elif location == -1: # can't find, then give up
                                         delimiter = -1
                                 else:
                                         print "error"
        return genomeplot;
genomeplot = plotcount(genome,pool,1)
genomecomp = comp(genome)
genomecompplot = plotcount(genomecomp, pool, -1)
Aplotfilename = "../plotdata/"+sys.argv[2][3:]+sys.argv[1][:-4] + '.A.txt'
if (exists(Aplotfilename)):
        os.remove (Aplotfilename)
Aplotfile = open(Aplotfilename, 'a')
cPickle.dump(genomeplot, Aplotfile)
ASplotfilename = "../plotdata/"+sys.argv[2][3:]+sys.argv[1][:-4] + '.AS.txt'
```

```
if (exists(ASplotfilename)):
        os.remove (ASplotfilename)
ASplotfile = open(ASplotfilename, 'a')
cPickle.dump(genomecompplot, ASplotfile)
storefile = "../mapped/"+sys.argv[2][3:]+'.'+sys.argv[1]
if (exists(storefile)):
        os.remove(storefile)
out = open(storefile, 'a')
#dump file written format
#[0]: segments
#[1]: reads
#[4,5]: position on mapped genome [location, S/AS] !!!written at last
OutputThreshold = 0 # output sequences of reads higher than OutputThreshold
for i in xrange(0,len(pool)):
        written = '';
        if pool[i][3]!=[] and pool[i][1]>OutputThreshold:
                written += pool[i][0] + "\t" + str(pool[i][1]) + "\t"
                for j in xrange(0,len(pool[i][3])):
        # write 1) segments, 2) reads 5) location on genome 6) sense or antisense $repeat 5 6$
                        written += str(pool[i][3][j][0]) + "\t" + str(pool[i][3][j][1]) + "\t"
                written += ' n'
                out.write(written)
```

# **"CUT"** The input file is a FASTA format file of TMV-Cg genome. *pseudocode* for **CUT**:

Input: A FASTA fileResult: Generate 21-nt small RNAs from TMV genomeOutput: Another FASTA file1 Create and open output file.2 Load the TMV genome sequence into the memory.3 for i = 1 to (length of TMV genome - 21) do4 $eq \leftarrow$  bases on the genome from the *i*-th base to the (i + 20)-th base5Append seq and its FASTA header into output file

- 6 end
- 7 Close all files

*source code* for **CUT**:

```
#This software is a free software licensed under GNU General Public License version 3 or later.
# Cite this program as: Xiaopeng Qi, Forrest Sheng Bao and Zhixin Xie,
# Small RNA Deep Sequencing Reveals Role for Arabidopsis thaliana RNA-dependent
# RNA Polymerases in Viral siRNA Biogenesis, PLoS One, xxx: xxx, 2009
import sys,string
from numpy import *
def fastaread(filename):# this fastaread() is different from the one in miRNAclassify.py
        seq0 = '';
        for line in filename.readlines():
                if line[0]!= '>':
                        st = line.split()
                        st = st[0].upper()
                        seq0+=st;
        return seq0;
#read the first sequence
f1=open(sys.argv[1], 'r')
tmvpool = fastaread(f1)
#seq1=f1.readline()
#seq1=string.strip(seq1)
for i in xrange(0,len(tmvpool)-21):
        print '>',i,'| Random ';
        print tmvpool[i:i+21];
```