SOAPfilter--Illumina data filter program

——st_asm ShiYujian

Updata:

1、Add the ability of filtering sequencing data with native quality.
2、Improve PCR duplication filtering algorithm, reduce the memory consumption.
3、Filter PCR duplication within the whole library instead of the lane.
4、Merge the procedures of low quality data filtering and PCR duplication filtering. Save the IO (input/output) operation of temporary files to enhance the efficiency.

Program function:

1、Filter adapter contamination.
2、Filter reads with undersize insert size
3、Filter reads with low sequencing quality
   a) For read-end quality masking by EAMSS algorithm:
      i. Trim the low quality fragment at both end of sequencing reads.
      ii. Filter reads with too many (>=cutoff) N bases or polyA.
      iii. Filter reads with too many (>=cutoff) low quality bases.
   b) For native quality sequencing data:
      Estimate the error rate for each base of read based on phred quality, then do the filtering according to error rate and read length cutoff.
4、Filter PCR duplication, keep only one copy for duplicated reads per library.

implementation details:

1、Filter adapter contamination
   Filter read pair with adapter contamination when the insert size is undersize (less than the read length) (Figure 1):

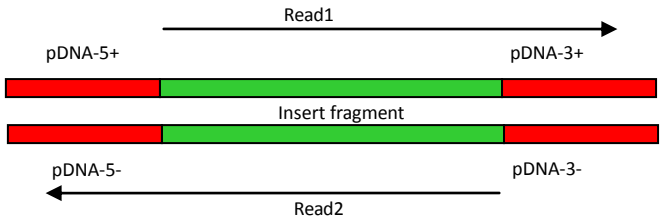

Figure 1: Read pair with adapter contamination. The insert size is undersize thus adapters were sequenced at both ends of reads.

2、Filter reads with undersize insert size
   Check whether read1 overlaps read2. Filter the reads pair when read1 overlaps read2 at least 10bp. Skip this step when the total length of read1 and read2 is longer than the designed insert size(for example: read length is 100, while insert size is 170).

3、Filter reads with low sequencing quality

a) For read-end quality masking by EAMSS algorithm:
   Filter read pair according to the cutoffs of low quality base or N.

b) For sequencing data with native quality:
   The native quality illumina used is phred quality(Q=log10[err_rate]), so we can estimate the sequencing error rate of each base of read. Then we do the filtering as below. First, set the minimum required read length(min_len) and maximum allowed error rate(max_err) for each read. Read pair will be kept if the estimated error rates of both reads are no larger than max_err. Otherwise if the longest fragment of read has error rate smaller than max_err, and length of at least min_len, the fragment will be output.

4、Filter PCR duplication, keep only one copy for duplicated reads.

a) As the sequencing length is becoming longer and longer, the FN (false negative) of filtering PCR duplication will increase if duplication is identified by detecting whether two or more whole reads-pairs are identical. Program now allows user to choose the fragments of read1 and read2 as tags to distinguish PCR duplication.

b) Introduce bloom filter to vastly decrease the memory usage.

c) Filter PCR duplication within the whole library instead of the lane.

Usage:

SOAPfilter_v2.2 [options] <lane.lst> <stat_file>

-q <int> the quality shift value 64 or 33, if this value is not set, program will detect it automatically
-m <int> the reads pair number in buffer,default: 1000000
-t <int> thread number, default: 8
-i <int> library insert size, default: 500
-y filter reads with adapter
  -F <str> adapter sequence for read1,default: AGATCGGAAGAGCGGTTCAGCAGGAATGCCGAG
  -R <str> adapter sequence for read2,default: AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT
-z filter reads with undersize insert size
-p filter PCR duplication
  -s<int> trimmed length at 5' end of read1 when distinguishing duplication, default: 0
  -l<int> the sub-length of read1 used to distinguish duplication,-1 for whole read, default: -1
  -S<int> trimmed length at 5' end of read2 when distinguishing duplication, default: 0
  -L<int> the sub-length of read2 used to distinguish duplication,-1 for whole read, default: -1

-g <int> output type, 0 for text, 1 for gz compressed format, default: 1
-o <str> suffix of output file name, default: clean

-M <int> how to filter low quality reads? 0 for no filtering, 1 for native quality reads mode, 2 for read-end quality masking by EAMSS algorithm(mask B); default: 0

For quality masked by EAMSS algorithm at the end of the read(-M 2):
-f <int> trim flag:-1 for no trimming, 0 for unify trimming, 1 for trimming min(maskB length,-b/-d);default:

0

-Q <int> the maximum low quality value, default: 7

-h output help information

Note:

"lane.lst" include the input read files from same library and their low-quality filtering parameters. The format of it is as below:

For quality masked by EAMSS algorithm at the end of the read(-M 2):

lane1_seq_file1 a b B

lane1_seq_file2 c d W

...

a,b mean the trimmed length at 5' and 3' end of reads in lane1_seq_file1, default: 0

c,d mean the trimmed length at 5' and 3' end of reads in lane1_seq_file2, default: 0

B means the low quality cutoff, reads with >=B% low quality bases(set by -Q) will be filtered, default: 40

W means the N base cutoff, reads with >=W% N bases will be filtered, default: 10

For native quality(-M 1):

lane1_seq_file1 e

lane1_seq_file2 n

...

e means the maximum allowed error ratio(estimated by phred quality) of result reads, default: 0.02

n means the minimum required length of result reads to output, -1 for the raw read length, default: -1

Output:

There are three output files:

1、Two fastq file for clean data of each lane. And they are gz compressed if -g was set to be 1.

2、The third file contains the statistical information of filtering(9 columns). First row is the header:

raw_read_id:the file name of raw data to distinguish different lane

raw_read_pair_num: raw read pair number

raw_read_length: raw read length

raw_base_num: raw base number

low_qual_filter(%): filtered low quality data in percentage

adapter_filter(%): filtered adapter contaminated data in percentage

undersize_ins_filter(%): filtered data with undersize insert size

duplicated_filter(%): filtered duplication data

clean_read_pair_num: result read pair number

clean_read_length: result read length

clean_base_num: result base number

Example:

./SOAPfilter_v2.2 -y -z -p -i 500 -M 2 -f 0 -g 0 -o clean lane.lst stat.txt >fil.out 2>fil.err

where the content of lane.lst is：

110114_I481_FC81C7HABXX_L5_HUMiqvDBTDIAAPE_1.fq.gz 0 0 40
110114_I481_FC81C7HABXX_L5_HUMiqvDBTDIAAPE_2.fq.gz 0 0 10

Notes:

1、Users can set the parameters to run each function independently.
2、The parameter of insert size is "-i" now, which is different from Filter_data_5.
3、"-M" is a very important parameter. It must be set according to the quality mode of raw data. The quality mode can be easily distinguished by the quality distribution of raw data.
4、The input file lane.lst is used to set the input raw data files and low quality filtering cutoff parameter. Note that the format should be chosen according to the quality mode of data.
5、Parameters "-s", "-l", "-S", "-L" take effect only when "-p" is set. And they are independent of trimming parameter set in "lane.lst". They work on the raw data and are used to distinguish PCR duplication only.
6、For native quality data, the maximum allowed error rate parameter "e" in "lane.lst" means the maximum estimated error rate of each single read of the clean data. So the total error rate of the clean data will be smaller than it.
7、For read-end quality masking by EAMSS algorithm data, parameters "B" and "w" in "lane.lst" are percentage, not numbers. Reads will be filtered if their low quality or N bases percentage is no less than the cutoff.
8、About the memory usage of this program: It will store 1M reads pair in the RAM, and filtering PCR duplication needs extra RAM. The larger the duplication rate of the raw data is, the more memory is needed. Test results show that, for 100M reads pair PE100 (20G bases) no jump data(duplication rate 0.13%), program needs about 2G RAM; and for 100M reads pair PE90 (18G bases) jump data(duplication rate 22.5%), program needs about 3G RAM.

Test result:

Resource consumption:
Test data：103M reads pair, PE100, total bases 20.58G
Used parameters:
1、For Filter_data_5:
First filter low quality data
parameters:-t 8 -m 1000000 -y -z -l 500 -a 0 -b 0 -c 0 -d 25 -B 40 -w 10; then filter PCR duplication.
2、For SOAPfilter_v2.2:
-t 8 -m 1000000 -g 0 -y -z -p -i 500 -M 2 -f 0； lane.lst：
XXX_1.fq.gz 0 0 40
XXX_2.fq.gz 0 25 10
The resource consumptions are as below:

Program	Time(sec)	Average memory usage(MB)	Maximum memory usage(MB)

Filter_data_5	filter_data_parallel	3029.69	1215.46	1221.16
	duplication	6998.41	5129.09	10110.72
	TOTAL	10028.10	3946.70	10110.72
SOAPfilter_v2.2		3994.10	1711.44	1985.43

Test results show that: SOAPfilter_v2.2 is much efficient in both running time and memory usage than the Filter_data_5. It can finish the filtering work of 20G raw data with 2G RAM in 67 minutes.

Filtering of PCR duplication :

Test data: raw sequencing data of two lanes from a same library. 44.3M reads pair, PE90 for each lane; total bases 15.95G

1. Filter PCR duplication for each lane independently. Used parameters:
   a) -t 8 -m 1000000 -g 0 -y -z -p –i 2580 -M 2 -f 0； lane1.lst：
      XXX_L1_XXX_1.fq.gz 2 3 40
      XXX_L1_XXX_2.fq.gz 2 3 10
   b) -t 8 -m 1000000 -g 0 -y -z -p –i 2580 -M 2 -f 0； lane2.lst：
      XXX_L2_XXX_1.fq.gz 2 3 40
      XXX_L2_XXX_2.fq.gz 2 3 10
2. Filter PCR duplication within the whole library. Used parameters:
   -t 8 -m 1000000 -g 0 -y -z -p –i 2580 -M 2 -f 0； lane.lst：
   XXX_L1_XXX_1.fq.gz 2 3 40
   XXX_L1_XXX_2.fq.gz 2 3 10
   XXX_L2_XXX_1.fq.gz 2 3 40
   XXX_L2_XXX_2.fq.gz 2 3 10

The results are asd below:

	filter within whole lib		filter within the lane	
	duplicated filter(%)	clean_base_num	duplicated filter(%)	clean_base_num
Lane1	3.41387	5,106,582,400	3.41387	5,106,582,400
Lane2	7.31925	4,818,266,650	3.53619	5,103,178,320
Total	5.36656	9,924,849,050	3.47503	10,209,760,720

Test results show that: To filter PCR duplication within the whole library can recognize and filter more duplicated reads. It will get more accurate results, which will decrease the influence of PCR duplication in downstream analyses.

Any questions or suggestions, please feel free to contact:

shiyujian@genomics.org.cn or st_asm@genomics.org.cn