User Guider

for

ASIIQT v 1.0

2020.02

Table of Content

1. Installation instructions

1.1 Installation environment requirements

Operating system: Linux, Mac or Windows
Software requirements: Golang compilation environment (required when compiling with source code; not required when using compiled program directly).

1.2 Installation

The compiled executable program does not need to be installed, and can be run directly in Command Prompt. After downloading the source package, you need to pre-install and configure the Golang compilation environment, and then compile the source code one by one to get the executable file and save it to the bin subfolder. In Win system, you can also compile all module source programs by double-clicking the *build.bat* file in the ASIIQT.v.1.0 folder to automatically generate all executable programs in the bin subfolder. Under each system, compile module by module using similar commands:
Linux / Mac system:

```
$ cd bin
$ go build ../codes/[module].go
```

Windows system:

```
$ cd bin
$ go build .. \ codes \[module].go
```

2. Protocols for ASIIQT analysis

i) Identfication of ASIs

0. Given the CCS reads of the single-molecule RNA-Seq: sample_raw.longReads.fasta;
   the high-quality isoform assembled using IsoSeq: sample_hq_isoform.fastq;
   NGS RNA-Seq reads: sample_NGS.reads.fasta;
   Reference genome: ref.fa

1. Mapping the CCS reads to reference genome with GMAP and minIntronLength = 25 nt, generating SAM file: sample_raw.longReads.sam

2.
```
./transcriptIdentify      sample_raw.longReads.sam      >sample_transcripts.txt
```

3. ./pseudoReadRefCoord sample_transcripts.txt >sample_pseudo.longReads.coord.txt

4. ./pseudoReadsSeqExt ref.fa sample_pseudo.longReads.coord.txt >sample_pseudo.longReads.fasta

5. ./rawReadSeqModify sample_transcripts.txt sample_raw.longReads.fasta
   >sample_modified.longReads.fasta

6. Mapping 'sample_NGS.reads.fasta' against 'sample_pseudo.longReads.fasta';
   Recalling SNPs with SAMtools, generating BCF-derived VCF file: sample_bcf.txt;
   Low-quality filtering of 'sample_bcf.txt', generating: sample_filtered.vcf.txt

7. ./bcfCorrect sample_bcf.txt sample_pseudo.longReads.fasta
   >sample_NGS.corrected.pseudo.longReads.fasta

8. ./lqMark sample_vcf.txt sample_NGS.corrected.pseudo.longReads.fasta
   >sample_lq_marked.NGS_corrected.pseudo.longReads.fasta

9. ./snpMark sample_vcf.txt sample_lq_marked.NGS_corrected.pseudo.longReads.fasta
   >sample_SNP_marked.NGS_corrected.pseudo.longReads.fasta

10. ./modifiedLongReadsCorrect sample_pseudo.longReads.coord.txt
    sample_modified.longReads.fasta sample_filtered.vcf.txt
    sample_SNP_marked.NGS_corrected.pseudo.longReads.fasta
    #Automatically generating two files: SNP_marked.NGS_corrected.modified.longReads.fasta;
    modified.longReads.SNP.annotation.txt;
    Renaming the files with 'sample_SNP_marked.NGS_corrected.modified.longReads.fasta' and
    'sample_modified.longReads.SNP.annotation.txt'

11. ./snpSet sample_pseudo.longReads.coord.txt sample_filtered.vcf.txt
    #Automatically generating a file: chr_index_SNP.txt;
    Renaming the file with 'sample_chr_index_SNP.txt'

12. ./phasing sample_modified.longReads.SNP.annotation.txt sample_chr_index_SNP
    sample_SNP_marked.NGS_corrected.modified.longReads.fasta
    #Automatically generating 9 files:
    (1) to_be_phased.txt ;
    (2) to_be_phased_2.txt;
    (3) to_be_phased_2_ann.txt;
    (4) to_be_phased_2_stat.txt;
    (5) phased_raw.txt;
    (6) phased_correction.txt
    (7) read_phased.txt;
    (8) corrected.modified.longReads.fasta;

(9) phased.modified.longReads.fasta

Renaming the files with a prefix 'sample_' added.

ii) Quantification of ASIs

13. Mapping 'sample_hq_isoform.fastq' to reference genome with GMAP, generating SAM file: sample_hq_isoform.sam

14.
```
./transcriptIdentify     sample_hq_isoform.sam     >sample_hq_isoform_transcripts.txt
```

15.
```
./phasedIso     sample_chr_index_SNP.txt     sample_hq_isoform.fastq
sample_phased_raw.txt     sample_hq_isoform_transcripts.txt
```
#Automatically generating 2 files:， isoform_phased.txt; phased.hq_isoform.fasta;
Renaming the files with 'sample_isoform_phased.txt' and 'sample_phased.hq_isoform.fasta' resepctively

16. Mapping 'sample_NGS.reads.fasta' against 'sample_hq_isoform.fastq' with bowtie or bwa, reporting at most 2 best matches, generating SAM file: sample_ngs.hq_isoform.sam

17.
```
./asiqFilt  sample_ngs.hq_isoform.sam     >sample_ngs.hq_isoform.filtered.sam
```
18.
```
./asiqsample_isoform_phased.txt   sample_ngs.hq_isoform.filtered.sam   >sample_asiq.out.txt
```

3. Instructions for use

3.1 Brief description

ASIIQT is developed by the Golang. Users need to run each module step by step in Command Prompt to get NGS_corrected_CCS_reads, allele-specific transcripts and quantification of transcripts. The source code of each program is stored in "codes", and the compiled program is stored in "bin".

3.2 Module Introduction

1. Module name: transcriptIdentify

Description: Filter out unqualified comparison read information.

Design: . Filter out:The length of sequences that can be compared is less than 0.67 of the full length of this read. Filtered out:Among the sequences on the alignment, if the length of the correct alignment is less than 75%. If there are reads that can be repeatedly compared, only the reads comparison information with the highest comparison quality is retained (if the quality of repeated comparisons is close, neither is retained)

Input: the SAM file (sam_file) generated by aligning the CCS reading to the human genome

Output: transcripts.txt generated from filtering the sam file. (This output needs to be manually output at the command prompt:> transcripts.txt)

2. Module name: pseudoReadRefCoord

Description: Extract the position information of each read alignment on the reference genome for the next module to extract the Pseudo-long reads

Design: Extract according to the matching starting position column and 'CIGAR' column information in the SAM file. For example, starting position is 1000, CIGAR is 8S42M1I30M2D5M10H. The reference genome matching position is calculated as: 1000_ (1000 + 42 + 30 + 2 + 5-1), which is 1000_1078 its starting position is 1000 and its ending position is 1078)

Input: transcripts.txt generated from filtering the sam file

Output: pseudo.longReads.coord.txt. (This output needs to be manually output at the command prompt:> pseudo.longReads.coord.txt)

3. Module name: pseudoReadSeqExt

Description: Get pseudo-long read (extract bases of reference genome in the framework of third-generation sequencing reads)

Design: According to the information in pseudo.longReads.coord.txt (which chromosome is reads aligned, and the position on the chromosome), the sequence is extracted and spliced, the introns are removed to obtain the pseudo-long reads.

Input: Reference genome sequence file (fasta file); pseudo.longReads.coord.txt

Output: pseudo long sequence *pseudo.longReads.fasta*. (This output needs to be manually output at the command prompt:> pseudo.longReads.coord.txt)

4. Module name: rawReadSeqModify

Description: When the SMS reads is mapped to the reference genome, it is necessary to convert the sequence aligned to the reverse strand of the reference genome into its reverse complement.

Design: The second column of the transcripts.txt file is the information on the alignment of the forward or reverse strand. If it is '16', the reads is reverse aligned with the reference genome,which is need to be converted into its reverse complement.'0', not required.

Input: transcripts.txt; third-generation raw sequence (RAW_LONG_READS.FASTA)

Output: modified.longReads.fasta (this output needs to be manually output at the command prompt:> modified.longReads.fasta)

5. Module name: bcfCorrect

Description: Use second generation sequencing data (NGS) with lower error rate to correct the pseudo-long reads.

Design: In the pseudo-long reads, the sequence covered by NGS is changed to upper case (corrected), and the uncovered sequence is lower case (no correction).

Input: ①.BCF file of second-generation sequence aligned with pseudo-long sequence (obtained through software *Bowtie*);

②.pseudo-long sequence (pseudo.longReads.fasta)

Output: NGS.corrected.pseudo.longReads.fasta (1st-corrected pseudo-long reads) (This output needs to be manually output at the command prompt:> NGS.corrected.pseudo.longReads.fasta)

6. Module name: lqMark

Description: In the VCF file (the information file of the second-generation sequence align to the pseudo-long sequence), find out the low-quality SNP location and mark it on the pseudo-long sequence

Design: Find the low-quality SNP loci in the VCF file (the information file of the second-generation sequence align to the pseudo-long sequence):if the information in the file is: AC = 0; AN = 0, change the base of the site in the pseudo-long sequence to lowercase.

Input: filtered SNPs annotated VCF file (obtained through software bowtie); NGS.corrected.pseudo.longReads.fasta

Output: lq_marked.NGS_corrected.pseudo.longReads.fasta (this output needs to be manually output at the command prompt:> lq_marked.NGS_corrected.pseudo.longReads.fasta)

7. Module name: snpMark

Description: In the VCF file, find homozygous and heterozygous mutations and mark them on the pseudo-long sequence in different ways.

Design: The information of the homozygous mutation in the file is: AC = 2; AN = 2, the sequence of the homozygous mutation SNP site is converted into 'X', and the information of the heterozygous mutation in the file is: AC = 1; AN = 2 Or AC = 1,1; AN = 2, mark it as '1'.

Input: filtered SNPs annotated VCF file;

lq_marked.NGS_corrected.pseudo.longReads.fasta

Output: SNP_marked.NGS_corrected.pseudo.longReads.fasta (this output needs to be manually output at the command prompt:> SNP_marked.NGS_corrected.pseudo.longReads.fasta)

8. Module name: modifiedLongReadsCorrect

Description: Correct the SNPs points marked in the pseudo-long sequence.

Design: For lowercase sequences, replace with the SMS reads original sequence; For sequences marked with 'X', replace with NGS sequences after mutation; For sequences marked with '1', judge in NGS Whether the two SNPs are the same as the third-generation sequencing original sequence. If they are the same, modify them to the same SNP; if they are not the same, use the SMS reads original sequence to replace them.

Input: . Pseudo.longReads.coord.txt (the output file of pseudoReadRefCoord module).

②.modified.longReads.fasta (original SMS reads file)

③.filtered SNPs annotated VCF file (obtained through software *bowtie*)

④.SNP_marked.NGS_corrected.pseudo.longReads.fasta.

Output: SNP_marked.NGS_corrected.modified.longReads.fasta:(fasta file marked with SNP).

②modified.longReads.SNP.annotation.txt(A three column table: reads name;category;chromosome_location_ SNPs composition_category) (category='1': no SMS reads sequencing error, and same as the NGS reads ; category='0': SMS reads sequencing is incorrect and inconsistent with the NGS reads) (if more than one '1', separated by '|' for example:chromosome_location_SNPs composition_category | chromosome_location_SNPs composition_category)

(This output is automatic output, you can use 'cd' to change the path of the output file)

9. Module name: snpSet

Description: Extract all SNPs and map them to chromosomes

Design: By comparing the third generation sequencing data with the reference genome (VCF file) and the second generation sequencing data (NGS) with the pseudo-long sequence alignment file, the SNP position can be traced back to the chromosome.

Input: . Pseudo.longReads.coord.txt (the output file of pseudoReadRefCoord module)

②.filtered SNPs annotated VCF file (obtained through software *bowtie*)

Output: chr_index_SNP.txt (two columns: chromosome name_position on chromosome;SNP composition)

(This output is automatic output, you can use 'cd' to change the path of the output file)

10. Module name: phasing

Description: Linking identified SNP loci

Design: No less than 2 SNPs whose 'Category' is '1' are extracted and made into a file "to_be_phased.txt" //e.g., Chr6_1110057_A_1 | Chr6_1110251_T_1 | Chr6_1110503_C_1

②. Combine the file "to_be_phased.txt" in pairs and separate them into multiple lines to generate "to_be_phased_2.txt"; // As in the example above: split into two lines, that is, Chr6_1110057_A_1 | Chr6_1110251_T_1 and Chr6_1110251_T_1 | Chr6_1110503_C_1

③ Annotate the above file "to_be_phased_2.txt" to generate "to_be_phased_2_ann.txt". As in the above example: each row is commented into two columns, the first column is the position, and the second column is the base composition. e.g., the first line is: Chr6_1110057 | Chr6_1110251 A | T. The second line comment is: Chr6_1110251 | Chr6_1110503 T | C

④ Calculate the above file "to_be_phased_2_ann.txt", and finally present the results in the file "to_be_phased_2_stat.txt". // For example, Chr6_1110057 | Chr6_1110251 A | T, C | A 3,6 2 (indicating that there are two possible linkages at those two positions, namely A | T and C | A, each of which appears 3 times and 6 times respectively , The last "2" indicates two possible linkages)

⑤ Remove 3 or more chaining possibilities in "to_be_phased_2_stat.txt". For only one type of linkage, a new file called "phased_raw.txt" is obtained by adding another SNP composition situation through the filter.vcf file and the base composition set of each SNP site.

⑥ Correct the partially uncertain SNPs through the identified linked SNPs, and record the changed one in the new file "phased_correction.txt" (three columns: sequence name; SNP sequence in this sequence; chromosome_position _ Corrected SNP composition.)

⑦In the modified.longReads.SNP.annotation.txt file, the third column contains '|' (regardless of one or more '|'), based on the pairwise position and base composition, combined with "phased_raw.txt" In the record of "2" possible linkage relationships, verify and correct the linkage relationship of each point and the corresponding base composition. For "2" in "phased_raw.txt", the unregistered locus or linkage relationship will be deleted, and the final reservation is confirmed. According to the SNP composition of the phasing result, record in two columns to "read_phased.txt": read name;chromosome_position_corrected SNP composition_1.It is possible that no information is recorded in the entire line in "read_phased.txt".

⑧ Correct the SNP_marked.NGS_corrected.modified.longReads.fasta according to the "phased_correction.txt" file. The resulting sequence file is called "Corrected_SMS_CCS.fasta".

⑨ Extract the "Corrected_SMS_CCS.fasta" sequence corresponding to the reads recorded in "read_phased.txt"; extract the reads with only 1 SNP and type "1" recorded in the original input table from the "Corrected_SMS_CCS.fasta" sequence; The above together generates a file: "phased_SMS_CCS.fasta", which contains the SNP and gets the fully modified sequence read.

Input: modified.longReads.SNP.annotation.txt

Chr_index_SNP

③SNP_marked.NGS_corrected.modified.longReads.fasta

Output: to_be_phased.txt

②to_be_phased_2.txt

③to_be_phased_2_ann.txt

④to_be_phased_2_stat.txt

⑤phased_raw.txt

⑥phased_correction.txt

⑦read_phased.txt

⑧corrected.modified.longReads.fasta

⑨phased.modified.longReads.fasta

(This output is automatic output, you can cd to a different path to modify the path of the output file)

Rename the above 9 files and add the sample tag prefix "sample_". The following are allele-specific high-quality isoform quantitative analysis

11. Module name: phasedIso

Description: Isolate RNA containing SNPs that have been linked into two.

Design: According to the SNP composition determined by the chr_index_SNP.txt file, the transcript containing SNP is separated into two (one from the mother and one from the father)

Input: .sample_chr_index_SNP.txt

②.sample_hq_isoform.fastq; (If the user inputs a fasta format file, the phaseIsoFA module can be used, except that the fastq file is replaced with a fasta file, and other inputs are the same as the phasedIso module input)

③.sample_phased_raw.txt;

④.sample_hq_isoform_transcripts.txt (generated by isotran.vs.ref.sam file processed by transcriptIdentify.go module)

Output: . Isoform_phased.txt;

②. Phased.hq_isoform.fasta

(This output is automatic. You can use 'cd' to change the path of the output file.)

12. Module name: asiqFilt

Description: Identify the NGS reads that are specifically aligned to the SMS reads and identify the positional relationship between them.

Design: Design: Filter out NGS reads that match one NGS read to multiple SMS reads and remove redundant sequences, and filter NGS sequences that result in low quality comparisons

Input: sample_ngs.hq_isoform.sam (Sam file obtained after NGS sequence comparison to SMS read, and then redundant; generated by software BWA or bowtie)

Output: sample_ngs.hq_isoform.filtered.sam (this output needs to be manually output at the command prompt:> sample_ngs.hq_isoform.filtered.sam)

13. Module name: asiq

Description: According to the sam file (a large number of NGS sequencing data align to CCS sequencing data), the transcripts that have been phased are counted and quantified.

Design: If a specific SNP is present in each NGS read, increase the specific transcript count by 1

Input: sample_isoform_phased.txt

sample_ngs.hq_isoform.filtered.sam (generated by the module asiqFilt)

Output: .asiq.out.txt (reads name_SNP position_SNP count)

(This output needs to be output manually at the command console:> asiq.out.txt)

Module name	Output file	Description

Module name	Output file	Description
transcriptIdentify	transcripts.txt	10 columns: read name (column 1), mapping flag (column 2), mapped chromosome number (column 3), starting position of matching (column 4), ending position of matching (starting position " CIGAR M / D / N / = / X character before the number "Sum-1), CIGAR (column 6), mapLength (the length of the sequence that can be compared), mapQuality (in the sequence that has been compared, the alignment Correct sequence ratio), intron set (if there is no intron, use "-" to represent; if there is, use a string to complete the expression. Multiple introns are separated by "|". Each intron It is expressed by "start position in reference sequence_end position in reference sequence".), Read only compares the previous chromosome to '1', and multiples to '2'. (For more specific details, please refer to the SAM format file description)
pseudoReadRefCoord	pseudo.longReads.coord.txt	5 columns: Read name, chromosome, match start site, CIGAR, chromosome match position (for example, start site 1000, CIGAR is 8S42M1I30M2D5M10H, chromosome match position is calculated as: 1000_ (1000 + 42 + 30 + 2 + 5 -1), which is 1000_1078)
pseudoReadSeqExt	pseudo.longReads.fasta	pseudo-long read (extract bases of reference genome in the framework of third-generation sequencing reads)

Module name	Output file	Description
rawReadSeqModify	modified.longReads.fasta	SMS original sequence (reversely complementary reads in the reverse alignment)
bcfCorrect	NGS.corrected.pseudo.longReads.fasta	The uppercase sequence is the NGS coverage area, and the lowercase sequence is the NGS uncovered area.
lqMark	lq_marked.NGS_corrected.pseudo.longReads.fasta	Lower case is NGS uncovered area or low quality SNP
snpMark	SNP_marked.NGS_corrected.pseudo.longReads.fasta	Added 'X' and '1' to the above file, where X is a homozygous SNP and 1 is a heterozygous SNP
modifiedLongReadsCorrect	SNP_marked.NGS_corrected.modified.longReads.fasta	Sequences with [] are SNP loci
	modified.longReads.SNP.annotation.txt	3 columns: sequence name, category, chromosome_location_specific SNP composition_category (category is '1': no third-generation sequencing error, consistent with second-generation; category is '0': there is third-generation sequencing error, inconsistent with second-generation) (More than one '1', separated by |: chromosome_location_specific SNP composition_category | chromosome_location_specific SNP composition_category)
snpSet	chr_index_SNP.txt	Column 2: [chromosome]_[position on chromosome], [SNP composition]

Module name	Output file	Description
phasing	to_be_phased.txt	Chromosome_Position on Chromosome_SNP Composition | Chromosome_Position on Chromosome_SNP Composition (For details, see the module introduction)
	to_be_phased_2.txt	Chromosome_Position on Chromosome_SNP Composition | Chromosome_Position on Chromosome_SNP Composition (For details, see the module introduction)）
	to_be_phased_2_ann.txt	Column 2: Position on chromosome_chromosome | Position on chromosome_chromosome; SNP composition | SNP composition
	to_be_phased_2_stat.txt	4 columns: chromosome_chromosome position|chromosome_chromosome position, SNP composition|SNP composition, SNP composition|SNP composition, number of occurrences, number of linkage relationships
	phased_raw.txt	4 columns: chromosome_chromosome position|chromosome_chromosome position, SNP composition|SNP composition, SNP composition|SNP composition, number of occurrences, number of linkage relationships
	phased_correction.txt	3 columns: sequence name, SNP order in the sequence, chromosome_position_corrected SNP composition

Module name	Output file	Description
	read_phased.txt	2 columns: read name, chromosome_location_corrected SNP composition_1
	corrected.modified.longReads.fasta	Sequences with [] are SNP loci
	phased.modified.longReads.fasta	Sequences with [] are SNP loci
phasedIso	isoform_phased.txt	4 columns: read name_SNP position at read_SNP composition, read name, chromosome_SNP position on chromosome_SNP composition, SNP position at read_SNP composition
	phased.hq_isoform.fasta	Fasta file: two allele-specific transcripts for every two lines
asiqFilt	sample_ngs.hq_isoform.filtered.sam	Sam file
asiq	sample_asiq.out.txt	2 columns: read name_SNP position_SNP composition | SNP position_SNP composition, quantity