SEECER: SEquencing Error CorrEction for Rna-Seq data

Hai-Son Le, Marcel Schulz and Ziv Bar-Joseph
{hple,maschulz,zivbj}@cs.cmu.edu

May 12, 2012

1 Overview

SEECER can work with single or paired-end read library. The input files should be in FASTA or FASTQ format. The pipeline of SEECER composes of 4 steps:

1.1 Step 1: Replacing ambiguous bases Ns

We remove ambiguous bases (Ns) from the read sequences before running SEECER by randomly substituting an N with one of the nucleotides (A,T,G,C). However, if there are regions with many Ns in a read, we discard the whole read unless these regions occur at the end, in which case, we truncate and keep the read if the new truncated length is at least half of the original. Reads that have more than 70% of their bases all As or all Ts are also discarded, as they likely originate from sequenced poly-A tails.To save memory, we also strip out read IDs and replaced by integers.

1.2 Step 2: Counting k-mers with JELLYFISH[1]

We support two options to construct the k-mer dictionary. The first option is directly counting the k-mers by iteratively going through the reads. The second option is using JELLYFISH as a proxy to obtain the k-mer counts. This option is recommended because JELLYFISH uses a clever method that supports a very large dataset with an extremely small memory footprint.

1.3 Step 3: Running the main program

This step runs the main program (in `src/`) to correct errors. The input to the main program is the read files, the optional k-mer count file (produced by JELLYFISH) and some parameters which are discussed more later.

1.4 Step 4: Cleaning up and putting back the original read IDs

In this final step, we put back the original read IDs. We also include in the final output the reads which were removed in Step 1. Therefore, the output files should contain exactly the same number of reads as in the input files.

2 Pipeline script

The pipeline is implemented in a BASH script: `run_seecer.sh` (in the folder `bin/`). The script accepts the following arguments:

```
run_seecer.sh [options] read1 read2

   read1, read2: are Fasta/Fastaq files.
      If only read1 is provided, the reads are considered singles.
```

```
            Otherwise, read1 and read2 are paired-end reads.

   Options:
      -t <v> : *required* specify a temporary working directory.
      -k <v> : specify a different K value (default = 17).
      -j <v> : specify the location of JELLYFISH binary
               (default = ../jellyfish-1.1.4/bin/jellyfish).
      -p <v> : specify extra SEECER parameters (default = '').
      -s <v> : specify the starting step ( default = 1). Values = 1,2,3,4.
      -h : help message
```

Specifically,

- `-t` **is *required*:** This argument specifies a temporary directory to store the output of JELLYFISH and some intermediate files of SEECER. It is important that the user provides different directory if many instances of the scripts are run on the same set of files.
- `-p`**:** This specifies extra SEECER parameters. We describe them in the next section.
- `-k`**:** This specifies the K value, which is the size of k-mers used by SEECER. The default value is 17.
- `-j`**:** This specifies the location of JELLYFISH binary.
- `-s`**:** This specifies the starting step. This is convenient when users want to experiment with different SEECER parameters. They need not to rerun step 1 and 2 in this case.

2.1 An example

To run the pipeline on a provided small data set, try:

```
bash ./bin/run_seecer.sh   testdata/SRR027877-small-p1.fastq \
    testdata/SRR027877-small-p2.fastq
```

3 Main program

The main part of SEECER is implemented in C++ files in `src/`. Users can change the default parameters of SEECER by the following command line options.

```
seecer [options] read1 [read2]
--------------------------------------------
 read1, read2: are Fasta/Fastaq files.
    If only read1 is provided, the reads are considered singles.
    Otherwise, read1 and read2 are paired-end reads.
*** Important ***:
Reads should not contain Ns. Please use the provided run_seecer.sh
script to handle Ns.
--------------------------------------------
Options:
--kmer <k> : specify a different K value (default = 17).
--kmerCount <f> : specify the file containing kmer counts. This file
    is produced by JELLYFISH, we provided a Bash script to generate this file
    (run_seecer.sh).
    If the parameter is not set, SEECER will count kmers by itself
    (slower and memory-inefficient).
--clusterLLH <e> : specify a different log likelihood threshold (default = -1).
--entropy <e> : specify a different entropy threshold (default = 0.6).
--help, -h : this help message.
```

- `--kmer`: This option specifies the K value, which is the size of k-mers used by SEECER. The default value is 17.
- `--kmerCount`: This option specifies file containing the k-mer counts. This file is produced by JELLYFISH. We also provided a Bash script to generate this file (`run_seecer.sh`). If the parameter is not set, SEECER will count the number of k-mers by itself (slower and memory-inefficient).
- `--clusterLLH`: This option specifies a different log likelihood threshold. The threshold defines a threshold value such that only reads whose the log-likelihood exceeds this value is assigned to the contig. The default value is -1.
- `--entropy`: This option specifies a different entropy cut-off. When SEECER extends the contigs, only bases whose entropy (of the emission probabilities) is below this value are added to the contig. The default value is 0.6.

References

[1] Marçais, G. and Kingsford, C. *Bioinformatics (Oxford, England)* **27**(6), 764–770 March (2011).