REPdenovo: Inferring de novo repeat motifs from short sequence reads

User Manual

Version 1.0.2

March 2, 2015

Chong Chu and Yufeng Wu
CSE Department, University of Connecticut Storrs, CT 06269, U.S.A.
Email: chong.chu,ywu@engr.uconn.edu

 A manuscript for this software is submitted to Bioinformatics. If you use this program in a publication, please cite the following reference: Chong Chu, Rasmus Nielsen and Yufeng Wu, REPdenovo: Inferring de novo repeat motifs from short sequence reads, Bioinformatics, 2015, submitted.
Please check back this site for more up-to-date information on citing this software.

1 Functionalities and Usage of REPdenovo

Based on the simple idea of frequent k-mer assembly, REPdenovo provides much more functionalities than REPARK. REPdenovo supports the following main functionalities.

1. Assembly. This step performs k-mer counting. Then we find frequent k-mers whose frequencies are over certain threshold. We then assemble these frequent k-mers into consensus repeats (in the form of contigs). Then we merge the constructed contigs to more completeness ones.

2. Scaffolding. We use paired-end reads to connect repeat contigs into scaffolds.

1.1 Dependencies

REPdenovo needs the following tools to be installed in the machine you are working on.

1. A k-mer counting tool. REPdenovo uses Jellyfish program for performing k-mer counting. Jellyfish can be downloaded from `https://github.com/gmarcais/Jellyfish`.

2. An reads assembler. REPdenovo uses Velvet at this point. In the future, we may support different assembler. Velvet can be downloaded from: `https://www.ebi.ac.uk/~zerbino/velvet/`. Caution: if you want to assemble k-mers that are longer than 30 bp, you need to recompile Velvet to let it work with longer sequence length. For example: make MAXKMERLENGTH=60. This makes Velvet work for k-mer length up to 60.

3. Reads mapping. REPdenovo uses BWA. BWA can be downloaded from `http://bio-bwa.sourceforge.net/`.

4. Sequence processing utilities. These include the commonly used SAMtools. Our code also uses BAMtools (`https://github.com/pezmaster31/bamtools`).

1.2 Preparing inputs

REPdenovo takes sequence reads in the FASTQ format (uncompressed or compressed). A raw reads file which list the path, mean and standard derivation of the insert size should be provided in the format:

Read-file-path group mean-insert-size insert-size-standard-derivation

For single end reads, group, mean-insert-size and insert-size-standard-derivation should be set to -1. For paired-end reads, left raw reads and right raw reads should be in separate files, and in two lines (one line for left raw reads, and the other for right raw reads). The "group" should be same for these two lines. Users can find one sample (for paired-end reads) from the same folder in this github cite.

REPdenovo needs a configuration file, which tells REPdenovo the basic settings. The following shows a typical settings file.

Here, we give an explanation on the parameters. In general, you should have all the entries shown in the figure. For some parameters, the values shown in the example are perhaps those that you should use (especially those are said to not change below).

1. MIN_REPEAT_FREQ. This is the cutoff of k-mers that are considered to be frequent for assembly. Note that this is the relative to the average coverage of the sequence reads. The average coverage of the sequence reads is calculated by the number of reads, reads length and the genome size.

2. RANGE_ASM_FREQ_DEC and RANGE_ASM_FREQ_GAP: these are used for assembly. Usually you don't need to change these.

3. K_MIN, K_MAX and K_INC: the smallest value, maximum value and increment of K. REPdenovo can use different K. In the example shown in the figure, three K values will be used: 30, 40 and 50.

4. K_DFT: default value of K value. This is equivalent of setting K_MIN = K_MAX = K_DFT.

```
MIN_REPEAT_FREQ 10
RANGE_ASM_FREQ_DEC 2
RANGE_ASM_FREQ_GAP 0.8
K_MIN 30
K_MAX 50
K_INC 10
K_DFT 30
READ_LENGTH 101
READ_DEPTH 7.206702255
JELLYFISH_TREADS 5
GENOME_LENGTH 3209300000
ASM_NODE_LENGTH_OFFSET -1
MIN_CONTIG_LENGTH 100
IS_DUPLICATE_REPEATS 0.85
COV_DIFF_CUTOFF 0.5
MIN_SUPPORT_PAIRS 20
MIN_FULLY_MAP_RATIO 0.2
TR_SIMILARITY 0.85
JELLYFISH_PATH /scratch2/jellyfish-2.1.4/bin/
VELVET_PATH /scratch2/velvet-master/
BWA_THREADS 5
OUTPUT_FOLDER ./human_NA12889/
VERBOSE 1
```

Figure 1: *Settings of REPdenovo.*

5. READ_LENGTH: length of reads.
6. READ_DEPTH: reads depth.
7. JELLYFISH_THREADS: how many threads to use to run Jellyfish.
8. GENOME_LENGTH: the length of the genome. Can only provide an approximate one.
9. ASM_NODE_LENGTH_OFFSET: if set to -1, then require each k-mer in the repeat be frequent. That is, all k-mers in a repeat is considered to be frequent.
10. MIN_CONTIG_LENGTH: the minimum length of the contigs output.
11. IS_DUPLICATE_REPEATS: ratio used to check whether two repeats are duplicate. If the similarity is over this threshold, then see the two repeats are duplicate.
12. COV_DIFF_CUTOFF, MIN_SUPPORT_PAIRS, MIN_FULLY_MAP_RATIO, : used by REPdenovo in improving quality of assembled repeats. You don't usually need to change these.
13. TR_SIMILARITY: REPdenovo merges two assembled repeats if their similarity is over this threshold.
14. JELLYFISH_PATH: set to the path of Jellyfish executable.
15. VELVET_PATH: set to the path of Velvet executable.
16. BWA_THREADS: the number of threads BWA is set to use.

17. OUTPUT_FOLDER : where to output the results. It is relative to the installation folder of REPdenovo.

18. VERBOSE. If set to be 1, output more information about the current running states of REPdenovo.

1.3 Basic usage

Run the assembly part: ▷ python ./main.py Assembly <configuration-file-name> <raw-reads-file-name>

Then run the scaffolding part: ▷ python ./main.py Scaffolding <configuration-file-name> <raw-reads-file-name>

1.4 Ouptput

Three main output:
contigs.fa which contains the constructed and merged contigs
X_merged.fa contains the scaffolds
X_contig_pairs_info.txt_cov_info_with_cutoff.txt contains the repeat coverage information

2 Credits

REPdenovo is developed by Chong Chu and Yufeng Wu. Rasmus Nielsen (UC Berkeley) provided many insights to the project.