Supporting Information S1. Procedure for automated circularization of contigs.

For Illumina data
Identification of circular contigs using overlapping sequences at both ends and information of paired-end reads
Design principle
If the contigs assembled by IDBA-UD is circular, it should meet the following 2 principles simultaneously:
1) Overlapping sequences might be present in both ends of the contig. IDBA-UD doesn't automatically cut such sequences.
2) We should find some paired reads located on both ends, respectively.
Test data
1 contig: Contig00032.fna (**/home/zfxu/data/tue/testdata/Illumina**)
Clean reads in FASTA format: read.fa
Script path: */home/zfxu/install/perl/*
====Requirement=======
==========
(1) Unix
    - The program has only been tested on UNIX
(2) BLAST
    - blastall 2.2.22
(3) Package AMOS (http://sourceforge.net/projects/amos/files/)
(4) Perl
    - Standard Perl
    - Bioperl
First step.
Identify circular contigs using overlapping sequences in both ends of the contigs.
Chop one contig in half at the middle of the sequence and then merge overlapping sequences through the program minimus2 in the package AMOS. If you get one contig back, you now have a perfect linear representation of a circular contig.
Chop one contig in half at the middle of the sequence. It will generate 2 sequences per contig
perl /home/zfxu/install/perl/chop.sequence.pl Contig00032.fna
Merge overlapping sequences using minimus2 with default parameters:
perl /home/zfxu/install/perl/minimus2.pl chopContig > nohup.minimus2.out
Put all the resulting *.fasta files into a new folder *fasta*. Overlapping sequences at the previous ends have been removed.
mkdir fasta
cp chopContig/.fasta fasta/*
Put all detected circular contigs into a file
perl /home/zfxu/install/perl/collection.pl fasta > circular.fna
Optional: Extract contigs more than 1kb (the majority of true plasmids should be more than 1kb). Please install Biopieces (http://code.google.com/p/biopieces/) if you want to run the following extraction.
read_fasta -i circular.fna | grab -e 'SEQ_LEN>=1000' | write_fasta -x -o contig1kb.fna
Second step
Use paired-end information to further determine the circular contig

Hypothesis: If the contig is circular, we should find some read pairs to connect both ends.
Please use the original contigs to do analysis not ones from the first step!
Extract both ends (500 bp) of the original contig (more than 1kb)
perl /home/zfxu/install/perl/fasta_manipulate.Endseq.pl Contig00032.fna 500
It will generate 2 files: contig.leftseq.fa and contig.rightseq.fa
The name of raw Illumina PE reads need to be modified: remove blank and add length values to the ends of the head line!
perl /home/zfxu/install/perl/fasta_manipulate.PEheader.pl read.fa > read.name.fa
Preparation of 2 databases for blastn
formatdb -i contig.leftseq.fa -p F -o T
formatdb -i contig.rightseq.fa -p F -o T
blastall -p blastn -d contig.leftseq.fa -i read.name.fa -o read2contigLeft.blastn -e 1e-10 -F F -m 8
blastall -p blastn -d contig.rightseq.fa -i read.name.fa -o read2contigRight.blastn -e 1e-10 -F F -m 8
To define the number of Illumina short reads (100 bp, 150 bp, 250 bp) matched to each end of each contigs, we used the following criteria: 100% alignment identity, alignment length of hit read at least 90 bp, and alignment length accounting for the hit read sequence should be greater than 99%.
perl /home/zfxu/install/perl/parse_PEread.1.pl read2contigRight.blastn > hitread2Right.tab
perl /home/zfxu/install/perl/parse_PEread.1.pl read2contigLeft.blastn > hitread2Left.tab
For the left end of the contig, only reserve the reads that are complementary reverse to the contig sequence.
perl /home/zfxu/install/perl/parse_PEread.2.pl hitread2Left.tab > hitread2Left.reverse.tab
If one of a read pair located on the left of the contig, check the other one of a read pair whether located on the right of the contig or not. If true, it will result in 2 files: the contig name (circular.PE.list) and the corresponding read names (circular.PE.read.list).
perl /home/zfxu/install/perl/parse_PEread.3.pl hitread2Left.reverse.tab hitread2Right.tab
Remove the duplicated contig names and reserve only one per contig
sort circular.PE.list | uniq > circular.PE.sort.list
Use the intersection of 2 tests as final result
perl /home/zfxu/install/perl/intesection.pl step1/circular.fna step2/circular.PE.sort.list > circular.final.list

For 454 data
Identification of circular contigs using information of single end reads
Design principle
If the contigs assembled by newbler is circular, it should meet the following principle:
1) The contigs should not have overlapping sequences in both ends. We should find some reads (mean length > 400 bp) just across both ends and perfect matching.
Test data
1 contig: contig.fna (/home/zfxu/data/tue/testdata/454)
Some reads: read.fa
Extract 100bp from each end of contigs, respectively. Concatenate two 100-bp fragments, put the right fragment ahead, and then followed by the left fragment.
perl /home/zfxu/install/perl/fasta_manipulate.pl contig.fna > 200bpEnd.fas

To detect reads that can entirely match the concatenation of both ends of a contig, a *blastn* search using all reads need to be performed against all 200-bp concatenated sequences per contig.
formatdb -i 200bpEnd.fas -p F
blastall -p blastn -d 200bpEnd.fas -i read.fa *-o read2Ends.m8.blastn -F F -e 1e-10 -m 8*
Criteria for extracting hit: alignment length 200bp without gap, 99% identity
perl /home/zfxu/install/perl/extractHit454.1.pl read2Ends.m8.blastn > hit.m8.blastn
The list of contigs detected to be putative circular sequence
perl /home/zfxu/install/perl/extractHit454.2.pl hit.m8.blastn > circularContig.list