Host 16s rDNA specific gRNA design for Cas-16S-seq

Kabin Xie

2019.09.09

A. Required software and programs for gRNA design
    EMBOSS package
    VSEARCH
    fun2bed.pl
    sam_summary.pl
    and couple lines of AWK and SED

B. The data processing procedure

1. Download unaligned fasta files of 16S rDNA from RDP, release 11, http://rdp.cme.msu.edu/misc/resources.jsp.

Gene	Aligned FASTA	Unaligned	Coverage Chart
Bacteria 16S	Aligned	Fasta, Genbank	Excel
Archaea 16S	Aligned	Fasta, Genbank	Excel
Fungal 28S	Aligned	Fasta, Genbank	Excel

2. Merged the sequences.

```
$grep -c ">" db/*fa
db/current_Archaea_unaligned.fa:160767
db/current_Bacteria_unaligned.fa:3196041

#combined three files in one fasta file
$ cat *fa > current.microbe.16S.fa
```

3. Identify PAM positions in RDP-rRNA collections.

```
#EMBOSS fuzznuc program is used to search the fasta file
# Get NGG sites from both strands.
$ fuzznuc -sequence db/current.microbe.16S.fa -pattern GG -outfile 16S_NGG.fuzznuc
$ fuzznuc -sequence db/current.microbe.16S.fa -pattern CC -outfile 16S_NCC.fuzznuc

#Get NAG site from both strands.
$ fuzznuc -sequence db/current.microbe.16S.fa -pattern AG -outfile 16S_NAG.fuzznuc
$ fuzznuc -sequence db/current.microbe.16S.fa -pattern CT -outfile 16S_CT.fuzznuc
```

4. Extract the guide sequence from fuzznuc results and output to to bed formate files

```
#Using fun2bed as: fun2bed.pl fuzznuc.result output-bed-file 0/1
# 0 indicates reverse complementary strand, 1 indicates forward strand NGG-PAM
# always examine the output files to make sure get correct results
```

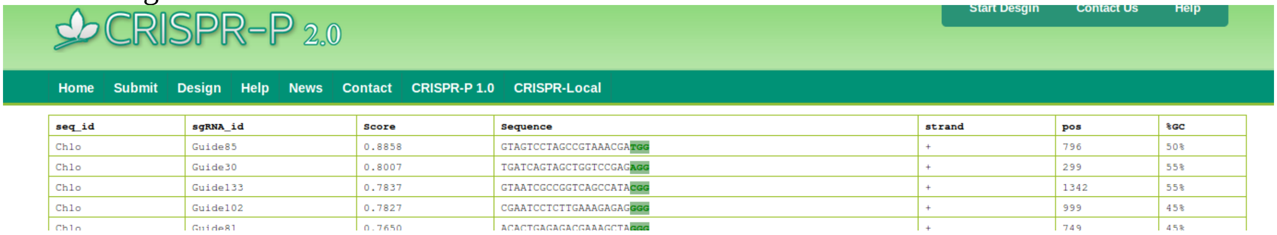
```
$./fun2bed.pl 16S_NCC.fuzznuc 16S_CC.bed 0
$./fun2bed.pl 16S_NGG.fuzznuc 16S_GG.bed 1

$ ../fun2bed.pl 16S_NAG.fuzznuc 16S_AG.bed 1
$ ../fun2bed.pl 16S_CT.fuzznuc 16S_CT.bed 0
```

5. Extract guide sequences from RDP-rRNA fasta files

```
#Extract the sequences from fasta file using bedtools.
$bedtools getfasta -fi ../db/current.microbe.16S.fa  -bed 16S_AG.bed -fo 16S_AG.tab -tab
$bedtools getfasta -fi ../db/current.microbe.16S.fa -bed 16S_CT.bed -fo 16S_CT.tab -tab  -s

# merged the bed files (NAG and NGG sites)
# remove redundant sequences and keep the frequency in sequence name.
# example:  >S001337072:1354-1374(-)#27, #27 indicate 27 records are merged.

$cat 16S_CC.tab 16S_GG.tab | sort -k 2 | uniq -c -f 1 | awk '{print ">" $2"#"$1 "\n" $3 }' > 16S_GG.all.uniq.fasta
$cat 16S_CT.tab 16S_AG.tab | sort -k 2 | uniq -c -f 1 | awk '{print ">" $1 "\n" $2}' > 16S_AG.all.uniq.fasta
```

6. Prepare host 16S rDNA guide sequence.

```
# Extract the 16S rDNA from rice reference genome (IRGSP4).
# Put the mt and cp rDNA sequence in Mito.fasta and chlo.fasta files, respectively
# Upload the fasta file to CRISPR-P V2.0/design

# Following is a screen shot of CRISPR-P 2.0 results.
```



seq_id	sgRNA_id	Score	Sequence	strand	pos	%GC
Chlo	Guide85	0.8858	GTAGTCCTAGCCGTAAACGATGG	+	796	50%
Chlo	Guide30	0.8007	TGATCAGTAGCTGGTCCGAGAGG	+	299	55%
Chlo	Guide133	0.7837	GTAATCGCCGGTCAGCCATACGG	+	1342	55%
Chlo	Guide102	0.7827	CGAATCCTCTTGAAAGAGAGGGG	+	999	45%
Chlo	Guide81	0.7650	ACACTGAGAGACGAAAGCTAGGG	+	749	45%

7. Recognize the PAM and extract the 20bp guide sequences using CRISPR P2.0

```
#Store the CRISPR P2.0 results in txt file (e.g. Os_16S_spacer.txt for mitochondrial gRNA, ).
# Extract the guide sequence and ID rom CRISPR-P output
#Using awk to covert txt file to fasta sequence file.
# The gRNA were renamed according to the PAM position with prefix as Chlo and Os(For mitochondrion gRNAs)

$awk '{print $8 "\n" $5}' Os_16S_spacer.txt > Os_16S_spacer.fasta

#extract the gRNA_ID and sequence to a tab separated file
$awk '{ print $8 "\t" $5 }' Os_16S_spacer.txt  | sed 's/>//' > Mito/Mito_spacer.tab
```

8. Perform global alignment using VSEARCH.

```
# Align to RDP_rRNA NGG-sites and NAG sites separately.

#mt-gRNA vs RDP-rRNA
$vsearch --usearch_global ./Os_16S_guide.fasta -db 16S_GG.all.uniq.fasta --id 0.6 --strand plus
--samout Os_microbe.vs.alnGG.sam --iddef 4 --minseqlength 1 --minwordmatches 1 --maxrejects
0 --maxaccepts 0
$vsearch --usearch_global ./Os_16S_guide.fasta -db 16S_AG.all.uniq.fasta --id 0.6 --strand plus
--samout Os_microbe.vs.alnAG.sam --iddef 4 --minseqlength 1 --minwordmatches 1 --maxrejects
0 --maxaccepts 0

# cp-gRNA vs microbial rDNA
$vsearch --usearch_global ./chl_rRNA_guide.fa -db ../db/16S_GG.all.uniq.fasta --id 0.6 --strand
plus --samout OsChl_microbe.vs.alnGG.sam --iddef 4 --minseqlength 1 --minwordmatches 1
--maxrejects 0 --maxaccepts 0

$vsearch --usearch_global ./chl_rRNA_guide.fa -db ../db/16S_AG.all.uniq.fasta --id 0.6 --strand
plus --samout OsChl_microbe.vs.alnAG.sam --iddef 4 --minseqlength 1 --minwordmatches 1
--maxrejects 0 --maxaccepts 0
```

9. Summarize the off-target number for each gRNA from alignment file using perl script.

```
#Usage: sam_summary.pl seed_region unaln_threshold unaln_seed_threshold <in.sam>
<out1_offtarget.tab> <out2_OT_no.tab>
#out1 extract align information from the input sam files, just for checking
#out2 file contains the gRNA_ID and total OT number. This is the data we need.

#for mt-gRNA
$../sam_summary.pl 12 4 1 Os_microbe.vs.alnGG.sam Os_microbe.vs.alnGG.sam.tab
Os_microbe.vs.alnGG.sam_ot_no.tab
$../sam_summary.pl 12 2 1 Os_microbe.vs.alnAG.sam Os_microbe.vs.alnAG.sam.tab
Os_microbe.vs.alnAG.sam_ot_no.tab

#For cp-gRNA
$../sam_summary.pl 12 4 1 OsChl_microbe.vs.alnGG.sam OsChl.vs.alnGG.sam.tab
OsChl.vs.alnGG.sam_ot_no.tab
$../sam_summary.pl 12 2 1 OsChl_microbe.vs.alnAG.sam OsChl.vs.alnAG.sam.tab
OsChl.vs.alnAG.sam_ot_no.tab

#We use following command to identify 100% matched GG sites.
$../sam_summary.pl 12 0 0 OsChl_microbe.vs.alnGG.sam OsChl.vs.alnGG.sam.00.tab
OsChl.vs.alnGG.sam_ot_no.00.tab
```

10. Combined the alnGG.sam_ot_no.tab, alnAG.sam_ot_no.tab and gRNA guide sequences using R or other table processing software.

11. Analyze the selected gRNA with 16S-seq results

```
$makeblastdb -in all.otus100.fasta -input_type fasta -dbtype nucl -parse_seqids -out OTUdb -title
OTUdb
```

```
$makeblastdb -in otu100.bact.fasta -input_type fasta -dbtype nucl -parse_seqids -out OTU_bact_db
-title  OTU_bact_db
```

```
blastn -query mt-gRNA.fasta -task blastn-short -db OTUdb -out mtvsOTU.out
-num_threads 4
```

```
blastn -query mt-gRNA.fasta -task blastn-short -db OTU_bact_db
-out mtvsOTU_bact.out -num_threads 4 -word_size 7 -evalue 10
```

```
vsearch --usearch_global ./mt-gRNA.fasta -db ./all.otus100.fasta --id 0.6 --strand plus --samout
./off_target/mt_gRNA.vs.OTU100.sam --iddef 4 --minseqlength 1 --minwordmatches 1
--maxrejects 0 --maxaccepts 0
```

```
makeblastdb -in all.otus100.v4.fasta -input_type fasta -dbtype nucl -parse_seqids -out
all.otu100.v4.db -title  all.otu100.v4.db
```