CRISPR-Seq Screen best practices:

**Reference library preparation:**

*Until clean libraries are produced*

1. Create a FASTA version of the library with one of the following scripts:
   1. sgRNA\_csv2fasta.py {fileIn}.csv {fileOut.fasta}
   2. sgRNA\_tab2fasta.pl {fileIn}.txt {fileOut.fasta}
2. Generate unique library with:
   1. unique\_sgRNA\_libMaker.py {fastaIn} {fastaOut}

The resulting library is designed around unique sgRNA sequences.

**Data preparation**

1. Pre-trim FastQC of fastq files

module load fastqc

fastqc -f <(zcat (fastq.gz) -o {outputdirectory)

1. Cutadapt
2. Post-trim FastQC
3. *If multiple wells for single sample:* Combine trimmed reads into a single file
   1. Single end:

cat \*.fastq.gz > {combinedReads}.fastq.gz

* 1. Paired end:

cat \*.R1.fastq.gz > {combinedReads}.R1.fastq.gz

cat \*.R2.fastq.gz > {combinedReads}.R2.fastq.gz

1. *If paired end reads*: Combine trimmed files into single file with algorithm of choice (e.g. CASPER)

module load casper

casper -t 24 -o {outfile\_name} <(zcat infile.R1.fastq.gz) <(zcat infile.R2.fastq.gz)

1. Convert read files to FASTA

module load fastxtoolkit

fastq\_to\_fasta -n -v -i {file}.fastq -o {outfile}.fasta

1. *Recommended*: Compress the combined fastq files for storage

tar -xcvf {file}.fastq.tar.gz {file}.fastq

**Alignment**

1. BLAT alignment

module load blat

blat -minscore=0 -stepSize=1 -tileSize=10 {uniqueReferenceFile}.fasta {readFile}.fasta outfile.psl

1. Parse BLAT output into 5 column text file, perfect match only

cat <(echo -e "sgRNA\ttranscript\tmatchLen\tstart\tend") <(tail -n +6 {blatResult}.psl |awk '{if ($1 ==20) print $14"\t"$10"\t"$1"\t"$12"\t"$13}') > {output}.txt

1. Count CRISPR values

crisprCounter.py {parsedBLAT\_tab}.txt {outputHeader}

**Analysis**

*Currently performed all on local machine*

1. Import output counts to R
2. Normalize to counts per million (CPM) on each sample

cpmReads = t(t(reads)/(apply(reads,2,sum)/1e6)

1. Determine fold change between CPM counts
2. Distribution of Log2FC

plot(dist(log2fc)

lines(rnorm(n=1000,mean=mean(log2fc),sd=sd(log2fc))

ks.test(log2fc, rnorm(n=1000,mean=mean(log2fc),sd=sd(log2fc))

*Warning, at large sample sizes, K-S test may be overly sensitive*

* 1. If normal
     1. No replicates: Can take top/bottom 2.5% as up/downregulated
     2. Replicates: Can perform t-test
  2. Else
     1. No replicates: ?
     2. Replicates: Try permutation table or binomial distribution

1. Export normalized count file for MAGeCK
2. Break up combined targets (even distribution)

blatMultimapLibSeparator.py {inputCounts} {outputCounts}

1. KO Experiments: MAGeCK (requires python)

*Citation:* [*https://genomebiology.biomedcentral.com/articles/10.1186/s13059-014-0554-4*](https://genomebiology.biomedcentral.com/articles/10.1186/s13059-014-0554-4)

* 1. Follow installation instructions: <https://sourceforge.net/p/mageck/wiki/Home/>
  2. Test the data:

mageck test -k {countsFile.txt} -t {treatment\_IDs} -c {control\_IDs} -n {outputPrefix}