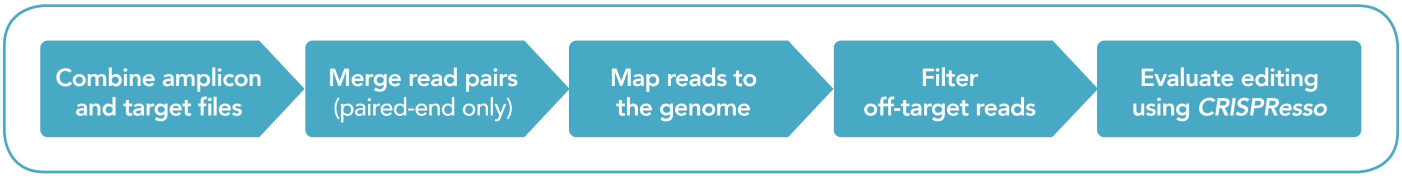
Evaluate CRISPR DNA editing with rhAmpSeq sequencing data

Introduction

This guideline demonstrates an example workflow for analyzing CRISPR DNA editing using PCR for target enrichment during next generation sequencing (NGS). Analyses will result in quantification of percent target editing and editing characterization. This workflow—useful for both biologists and bioinformaticians—maximizes accuracy and reduces complexity. It assumes a basic grasp of the Linux command line and that the Referenced software packages have been installed.

The workflow starts from a demultiplexed FASTQ file generated after sequencing. Since sequencing FASTQ files can originate from single- or paired-end sequencing. This workflow analyzes both singleplex and multiplexed PCR target enrichment.

Workflow


Combine amplicon and target files
Merge read pairs (paired-end only)
Map reads to the genome
Filter off-target reads
Evaluate editing using *CRISPResso*

Note: This workflow assumes that the sequencing run is demultiplexed using the standard methods recommended for your specific sequencing platform.

Reference software packages

The following software packages are referenced within our examples:

Package	Version	License	URL
Picard	2.18.11	MIT	https://github.com/broadinstitute/picard
minimap2	2.12	MIT	https://github.com/lh3/minimap2
samtools	1.9	CC	http://www.htslib.org/doc/samtools.html
FLASH	1.2.11	OSD	https://ccb.jhu.edu/software/FLASH/
bedtools	2.27.1	GPL2	https://bedtools.readthedocs.io/en/latest/
CRISPResso	1.0.13	GPL	https://github.com/lucapinello/CRISPResso
BamTools	2.4.1	MIT	https://github.com/pezmaster31/bamtools

Note: Since input parameters for each tool affect analysis results, we recommend that you first run individual tools with the *--help* option to view the entire list of available parameters.

Combine amplicon and target files

Before starting, make sure you have two separate input files containing the genomic coordinates for the guide RNA sequences and expected amplicons in BED file format (.bed). BED files should be 6-column, tab-delimited files (without headers) that contain the following fields:

```
Chromosome,Start,Stop,Name,Score,Strand
```

Tip: The value of the *Score* field is a placeholder in this instance and can be any numeric value.

Combine guide and amplicon BED files by coordinate intersection to validate that each guide references the correct amplicon target.

Example output:

```
bedtools intersect -loj -a amplicons.bed \
-b guides.bed > merged.bed
```

The resulting file will contain amplicons (-a) with their corresponding guides (-b) on the same line.

Note: If multiple guides overlap the same amplicon, they will be added to the same line. Guides that do not overlap an amplicon will not be a part of the output.

Merge read pairs (R1/R2)

If you are starting with single-end sequencing data, skip merging and go directly to Map reads to the genome.

Illumina paired-end sequencing results in two FASTQ files that together represent the observed amplicon. Merge the read pairs (R1 and R2) into a single fragment to increase FASTQ read mapping accuracy.

To merge R1 and R2 read pairs into a single fragment, set the *-M* parameter to a value larger than the sequenced read length to prevent read pairs with a large sequence overlap from being discarded.

Example output:

```
flash -O -M 152 reads1.fq reads2.fq 2>&1 | tee flash.log
```

Example output:

```
- out.extendedFrags.fastq = The merged reads.
- out.notCombined_1.fastq = Read 1 of mate pairs not merged
- out.notCombined_2.fastq = Read 2 of mate pairs not merged
- out.hist = Numeric histogram of merged read lengths
- out.histogram = Visual histogram of merged read lengths
```

Note: Amplicons larger than twice the sequencing length will not be merged.

Map reads to the genome

Reads are aligned twice. First, reads are alignment to the reference genome enabling read-to-amplicon assignment. Second, reads assigned to each amplicon are realigned to that amplicon using a more strigent approach, improving mutation characterization.

1. Create a minimizer index of the reference genome.

   Code example:

   ```
   minimap2 -d reference.mmi reference.fasta
   ```

   Tip: Make sure you are using an appropriate reference genome for your experiment. See details of what an appropriate reference is here.

   Important! Ensure your computer has enough working memory since a large amount of RAM (~14 GB) is required to index the human reference genome.

2. Map merged reads to the genome using minimap2.

   Example output:

   ```
   minimap2 -a -c --MD -no-end-flt \
   reference.mmi out.extendedFrags.fastq | \
   samtools view -hb -F 4 -F 0x900 - > mapped.bam
   ```

   Note: Reads with R1 and R2 pairs that did not merge will not be included in the analysis.

Filter off-target reads

CRISPResso analyzes single amplicon targets using reads from a FASTQ file. Therefore, to prevent cross-contamination of reads across amplicon targets, it is best to isolate sequencing reads generated from each amplicon into their own corresponding FASTQ file.

1. Separate reads by the intended amplicon target, then tag genome-aligned reads with the associated amplicon using bedtools.

   Example output:

   ```
   bedtools tag -names -i mapped.bam -files merged.bed > tag.bam
   ```

2. Split the reads into separate BAM files by the target amplicon.

   Example output:

   ```
   bamtools split -in tag.bam -tag YB -stub Sample \
   -tagPrefix Target_
   ```

   Note: BamTools creates a BAM output file, but CRISPResso requires a FASTQ input.

3. Convert the separate BAM files into FASTQ using the SamToFastq (Picard) tool.

   Tip: Use a loop to simplify processing, as indicated in the example.

   Example output:

   ```
   for bam in *.bam; do
   java -jar picard.jar SamToFastq \
   Input=${bam} \
   FASTQ=./$(basename ${bam} .bam).fastq.gz
   done
   ```

Evaluate editing with CRISPResso

We recommend using CRISPResso to characterize your target editing. CRISPResso requires the amplicon and guide sequence of each target individually. To provide this, you will have to generate files that contain the amplicon and guide sequences.

1. Generate all target amplicon sequences into a FASTA file.

   Example output:

   ```
   cut -f 1-6 merged.bed | bedtools getfasta -fi reference.fa \
   -bed - -name -fo amplicon_seq.fasta
   ```

2. Generate the guide sequences.

   Use a similar command as the previous example, but change the columns and add a flag that forces genomic strandedness (-s).

   Tip: CRISPResso requires the guide sequence to be in the correct strand orientation to correctly locate the PAM (protospacer adjacent motif).

   Example output:

   ```
   cut -f 7-12 merged.bed | bedtools getfasta -fi reference.fa \
   -bed - -name -s -fo guide_seq.fasta
   ```

3. Run a function that extracts the guide and amplicon sequences for a target from files. Use the following command to extract the amplicon (-a) and guide (-g) sequence with the relevant FASTQ file generated earlier (-r1).

   Note: We recommend looking for mutations within 10 bp of the expected cut site (-w).

   Example output:

   ```
   for name in $(cat merged.bed | cut -f 10); do
   guide_seq=$(grep -A 1 -w ${name} guide_seq.fasta | tail -n 1);
   amplicon_seq=$(grep -A 1 -w ${name} amplicon_seq.fasta | tail -n 1);
   fastq_path=$(ls ./*${name}.*fastq.gz);
   CRISPResso -r1 ${fastq_path} -a ${amplicon_seq} -g ${guide_seq} -w 10 -n \
   ${name};
   done
   ```

Note: If your reads were not adapter-trimmed during demultiplexing, you can specify that CRISPResso trims your reads now. Refer to the CRISPResso documentation for any necessary commands for your specific adapters.

Important! Your analysis may require some experiment-specific modifications:

- If your experiment involved homology-directed repair (HDR), add -e, followed by the sequence of the theoretically perfect HDR amplicon.
- If using Cas12a (Cpf1), provide the following flag to correctly identify the nuclease cut site: *--cleavage_offset 1*.
- If an amplicon was targeted by multiple guides, each guide can be comma-separated to define the multiple cut sites (e.g., -g ACTGGGTC,GGTCGGTCG).

Your results are ready to be interpreted. For more information on the output graphs and summaries, see the original CRISPResso publication.

Note: There are other run options for this tool such as *CRISPRessoPooled* and *CRISPRessoWGS*; however, we do not recommend using these tools for analysis due to their inherent risk for introducing potential errors that could produce inaccurate results.

(Optional) Validate your pipeline

To validate that your pipeline is working as expected, we provide a dataset with expected results to compare against.

Note: This example contains data generated for an on-target guide targeting the HPRT38087 locus (HPRT38087_iGS_1) and all off-target locations identified using the GUIDE-seq off-target identification method.

Example output:

```
diff -u file1 file2 | sed -n `1,2d;/^[-+]/p`
```

Note: As a quick check, look for any differences between the two equivalent files. Any differences between your results, and the generated results, will be highlighted line-by-line.

When there are no apparent differences between the two equivalent files, you have successfully validated your pipeline run.

References

1. UCSC Genome Browser—https://genome.ucsc.edu/FAQ/FAQformat.html#format1
2. Heng Li's blog 13 November 2017—https://lh3.github.io/2017/11/13/which-human-reference-genome-to-use
3. GitHub—https://github.com/lucapinello/CRISPResso
4. NCBI—https://www.ncbi.nlm.nih.gov/pubmed/27404874
5. IDT—https://www.idtdna.com/pages/docs/CRISPR-sample-data-rhAmpSeq-tar.gz

Technical support:

applicationsupport@idtdna.com

For Research Use Only. Not for use in diagnostic procedures.