Running a CRISPR screen analysis.

This is a tutorial for differential analysis for pooled CRISPR screens starting from next-generation sequencing data in FASTQ format and a CRISPR library.

This tutorial requires a computer equipped with a 3.2 GHz quad-core CPU, 16 GB RAM, and 500 GB hard drive or better. Some command-line experience is assumed. Allow 4-6 hours to complete this tutorial.

Contents.

* Introduction.
* Set up environment.
* Obtain and exorcise library.
* Count reads.
* Define comparisons.
* Run differential analysis.
* Explore results.
* Conclusion.

Introduction.

Data from pooled CRISPR screens are nucleotide reads from next-generation sequencing (NGS) of samples taken at the endpoints, early timepoints, or initial populations. NGS captures the populations of CRISPR guides at the timepoints and these populations are analysed by differential analysis to identify the guides that are enriched or depleted compared to control.

A CRISPR analysis therefore starts by obtaining sequencing reads and mapping them to the CRISPR guide library. Counts are then determined for each guide in the library for each sample in the experiment. Samples are then chosen as differential and reference levels and differential analysis is performed.

In this tutorial, we will validate CRISPR guide libraries by reannotation to GRCh38/GRCm39 using exorcise and perform batch differential analysis using crispr\_tools.

Set up environment.

This section assumes that you have [Anaconda](https://www.anaconda.com/download) and [git](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git) installed. If not, install those first using the appropriate guide for your system. In this section, we will set up a programming environment for library reannotation with exorcise, CRISPR analysis pipeline with crispr\_tools, and downloading files from Sequencing Read Archive with sratoolkit.

Aims:

* Download and install exorcise and crispr\_tools.
* Install exorcise and crispr\_tools conda environments.
* Download and install sratoolkit.

1. Navigate to your desired installation destination and then clone exorcise and crispr\_tools.

git clone <https://github.com/SimonLammmm/exorcise.git>

git clone <https://github.com/SimonLammmm/crispr_tools.git>

1. Install the conda environments. It is acceptable to combine them into a single environment if desired.

conda env create -f exorcise/env/exorcise.yaml

conda env create -f crispr\_tools/env/crispr\_tools.yaml

1. Install exorcise to your .zshrc and crispr\_tools using pip.

conda activate crispr\_tools

pip install -e crispr\_tools/

chmod 755 exorcise/bin/\*

echo "PATH=$PATH:`pwd`/exorcise/bin" >> ~/.zshrc

source ~/.zshrc

1. Download the [SRA toolkit](https://github.com/ncbi/sra-tools/wiki/01.-Downloading-SRA-Toolkit) release appropriate for your system and install it to your .zshrc.

tar -xzvf sratoolkit.3.0.6-mac64.tar.gz

echo "PATH=$PATH:`pwd`/sratoolkit.3.0.6-mac64/bin" >> ~/.zshrc

source ~/.zshrc

1. Feel free to clean up the PATH lines added to your .zshrc if you wish.

Obtain and exorcise library.

In this section, we'll download the Brunello library and use exorcise to reannotate its guides against GRCh38. This has the following effects:

* verifying that the Brunello guides hit exons in GRCh38, and
* harmonising gene symbols so that screens with different libraries can be analysed together later.

Aims:

* Download and exorcise Brunello.
* Learn how to find genomes, exomes, and feature priority lists to send to exorcise.

1. Download Brunello library broadgpp-brunello-library-contents.txt from [Addgene](https://www.addgene.org/pooled-library/broadgpp-human-knockout-brunello/). Scroll down to Depositor Data to find the download link.
2. Download genome hg38.2bit from [UCSC Genome Browser Downloads](https://hgdownload.soe.ucsc.edu/downloads.html). Under "Human genomes", "Dec. 2013 (GRCh38/hg38)", click "Standard genome sequence files and select annotations (2bit, GTF, GC-content, etc)" and then scroll down and click to download the file in the directory listing that shows.
3. Download exome hsa.grch38.refseqall.gz from [UCSC Table Browser](https://genome.ucsc.edu/cgi-bin/hgTables). In the Table Browser, specify the settings:

|  |  |
| --- | --- |
| clade | Mammal |
| genome | Human |
| assembly | Dec. 2013 (GRCh38/hg38) |
| group | Genes and Gene Predictions |
| track | NCBI RefSeq |
| table | RefSeq All (ncbiRefSeq) |
| region | genome |
| identifiers (names/accessions) | Leave default |
| filter | Leave default |
| subtrack merge | Leave default |
| intersection | Leave default |
| output format | selected fields and primary and related tables |
| output filename | hsa.grch38.refseqall |
| output field separator | tsv (tab-separated) |
| file type returned | gzip compressed |

Click "get output". In the page that appears, select the following fields by clicking the checkboxes:

* chrom
* strand
* exonStarts
* exonEnds
* name2

And then click "get output".

1. Download gene annotations from [NCBI Datasets](https://www.ncbi.nlm.nih.gov/datasets/gene/). Type Human in the Taxon field and leave the Gene symbol(s) field blank. Hit Search. In Select columns, tick Symbol and Gene type, then Apply. Select all the genes, click Download, and then Download Table. Save the file as hsa.priorities.tsv. This file will be used as the feature priorities list.

We'll now run exorcise on the downloaded Brunello library which we'll pass using -i. We also need to pass column numbers of the gRNA sequences (-g) and original gene symbols (-n), PAM (-z), control guide names (-c), and the genome, exome, and priorities we downloaded earlier (-v, -w, -y). Specify the output folder with -o.

1. Activate and run exorcise.

conda activate exorcise

exorcise -i broadgpp-brunello-library-contents.txt -g 7 -n 2 -z NGG -c ﻿Non-Targeting -v hg38.2bit -w hsa.grch38.refseqall.gz -y hsa.priorities.tsv -o brunello-library/

1. Inspect the file that appeared at brunello-library/exorcise.tsv. This contains exo\_symbol (guide-level reannotation) and exo\_harm (gene-level harmonisation) columns. Notice that the annotations in exo\_symbol are independent of the original Target Gene Symbol but the harmonisations in the exo\_harm column for guides with the same Target Gene Symbol are the same. Briefly, reannotations reflect the target genome and exome more closely and have the highest potential to maximise discovery but might introduce biases due to an unbalanced number of guides per gene. Harmonisation is more conservative, retaining the number of guides per gene but simply updating the gene symbols to the symbols found in the exome. Harmonisation more closely reflects the authors' original analysis.

Count reads.

Now that we have the library and have verified its sequences, we now know what genes the sequences obtained by NGS correspond to. In this section, we will count reads from FASTQ and map them to the library using exo\_symbol (reannotations) and exo\_harm (harmonisations), separately. Later, we will see the impact this has on discovery.

We'll take the data for this tutorial from the Brunello experiment in Meljuso cells published by [DeWeirdt et al, 2020](https://doi.org/10.1038/s41467-020-14620-6).

Aims:

* Download NGS data in FASTQ format.
* Determine where the protospacer is in NGS reads.
* Count reads and map to an exorcised library.

1. Go to [SRA Run Selector](https://www.ncbi.nlm.nih.gov/Traces/study/) and search for project SRP217813. Select all the runs that used the MelJuSo cell line (10 files in total). Under "Select" > "Selected", click "Metadata" to download the metadata for those samples.
2. Download the selected runs using sratoolkit. This might take a while depending on your internet connection.

prefetch SRR9926536 SRR9926538 SRR9926541 SRR9926543 SRR9926545 SRR9926526 SRR9926535 SRR9926539 SRR9926540 SRR9926544

fastq-dump SRR\*/\*.sra

We now have the FASTQ files for this NGS sequencing run. Determine the position of the protospacer (guide sequence) within the run using exorcise. We only need to do this on the first few reads of the first file as we can assume that the position is fixed across all reads in all files.

1. Still with exorcise activated, determine the protospacer location.

ntByCycle -f SRR9926536.fastq -n 10000 -o ntByCycle/

1. Inspect the nucleotide trace that appears at ntByCycle/SRR9926536.pdf. Notice a region of uniform nucleotide frequency from cycles 30 to 49. Notice that this interval is the same length as a Brunello guide (20 nucleotides). Convert the interval to zero-based half-open.
2. Activate crispr\_tools and then count reads and map to library reannotations and harmonisations, separately.

conda deactivate

conda activate crispr\_tools

count\_reads.py \*.fastq -s 29,49 --library=brunello-library/exorcise.tsv -j exo\_id -g exo\_seq -n exo\_symbol -p reannotate/cts/cts

mv reannotate/cts/\*.rawcount.txt harmonise/cts/

count\_reads.py \*.fastq -s 29,49 --library=brunello-library/exorcise.tsv -j exo\_id -g exo\_seq -n exo\_harm -p harmonise/cts/cts

We now have the read counts and have mapped them to the reannotated and harmonised libraries.

Define comparisons.

In this section, we define which samples to compare against each other. We will edit an Excel document that tells the differential analysis program the sample to set as the differential level and the sample to set as the reference level. The program will run all the comparisons in batch.

1. Open the Excel file at crispr\_tools/det/DETAILSTEMPLATE.xlsx.
2. In Experimental details, populate Experiment name with "reannotate/res/", analysis\_version with "1", and file\_prefix with "reannotate/res/".
3. In Sample details, populate the table with help from the SRA metadata table from earlier, SraRunTable.txt. The Replicate column must be populated with column names in cts.counts.tsv. Populate the Sample column with grouping variables – rows with the same Sample are treated as replicates of that sample.
4. In Control groups, populate the Control sample and Test samples with pairs of Samples from Sample details that you would like to test. Fill the Group column with "endpoints".
5. In Analyses, fill the Control group with "endpoints", Paired with FALSE, Counts files with "reannotate/cts/cts.counts.tsv", and Add pseudocount with 5.
6. Save a copy of the workbook to reannotate/det/reannotate.xlsx. This is the experiment for the guide-level reannotation Brunello.
7. Staying in Excel, replace all instances of "reannotate" with "harmonise" and then save another copy of the workbook to harmonise/det/harmonise.xlsx. This is the experiment for the gene-level harmonisation of Brunello.

Run differential analysis.

We've verified the library guides, found the guide position within the reads, counted the reads, and defined the experiments. We're finally ready to run differential analysis.

1. Still with crispr\_tools activated, run crispr\_pipeline on the two Excel workbooks, separately.

crispr\_pipeline.py reannotate/det/reannotate.xlsx

crispr\_pipeline.py harmonise/det/harmonise.xlsx

This will take a while, especially the MAGeCK steps. Results will appear in reannotate/res/ and harmonise/res/ subdirectories.

Explore results.

The CRISPR screen data have now been analysed and you can start exploring results. This is freeform as you can explore the results in your favourite way but continue following the tutorial for some standard differential analysis explorations.

Aims:

* Verify that the screen analysis replicates the results in [DeWeirdt et al, 2020](https://doi.org/10.1038/s41467-020-14620-6).
* Inspect the impact of guide-level exorcise (reannotation) versus gene-level exorcise (harmonisation).

In the paper, the authors report (Figure 2e):

* loss of BAX leads to resistance to BCL2L1 knockout phenotype in Meljuso cells,
* loss of MARCH5 hypersensitises cells to BCL2L1 knockout in Meljuso cells,
* loss of WSB2 hypersensitises cells to MCL1 knockout in Meljuso cells, and
* loss of TP53 leads to resistance in both BCL2L1 and MCL1 knockouts in Meljuso cells.

We'll create a biplot of BCL2L1 response versus MCL1 response to replicate the authors' analysis.

1. For the purposes of plotting, activate the exorcise environment and open RStudio.

conda deactivate

conda activate exorcise

rstudio

1. Type and run the following code into the editor to plot a biplot.

suppressPackageStartupMessages({

library(dplyr)

library(data.table)

library(ggplot2)

})

x <- fread("harmonise/res/1/drugz/files/result.Control-BCL2L1.tsv") %>% transmute(GENE, BCL2L1 = normZ)

y <- fread("harmonise/res/1/drugz/files/result.Control-MCL1.tsv") %>% transmute(GENE, MCL1 = normZ)

goi <- c("BCL2L2", "TP53", "UBE2J2", "UBE2K", "MCL1", "MARCHF5", "PMAIP1", "BAX", "WSB2", "BCL2L1")

plotdata <- full\_join(x, y, by = "GENE") %>%

mutate(isgoi = case\_when(GENE %in% goi ~ T, T ~ F),

label = case\_when(isgoi ~ GENE, T ~ NA)) %>%

arrange(isgoi)

p <- ggplot(plotdata, aes(x = BCL2L1, y = MCL1, colour = isgoi, label = label)) +

geom\_point() + geom\_text() + theme\_classic()

p

1. In the plot that appears, the x-axis is the normZ score for BCL2L1 knockout versus control and the y-axis is the normZ score for MCL1 knockout versus control. Highlighted genes are those which were also on the authors' original plot. Observe that we reproduced the authors' results.

Next, we'll inspect the effect that reannotation had over harmonisation.

1. Still in RStudio, clear the editor, plots, and environment variables and type and run the following code.

suppressPackageStartupMessages({

library(dplyr)

library(data.table)

library(ggplot2)

})

x <- fread("harmonise/res/1/drugz/files/result.Control-A1331852.tsv") %>% transmute(GENE, harm = normZ)

y <- fread("reannotate/res/1/drugz/files/result.Control-A1331852.tsv") %>% transmute(GENE, re = normZ)

plotdata <- full\_join(x, y, by = "GENE") %>%

mutate(isgoi = case\_when(abs(harm - re) > 4 ~ T, T ~ F),

label = case\_when(isgoi ~ GENE, T ~ NA)) %>%

arrange(isgoi)

p <- ggplot(plotdata, aes(x = harm, y = re, colour = isgoi, label = label)) +

geom\_point() + geom\_text() + theme\_classic()

p

1. In the plot that appears, the x-axis is the normZ score after harmonisation and the y-axis is the normZ score after reannotation. Both are for the comparison of A1331852 drug (BCL2L1 inhibitor) versus control. Labelled are genes with a delta normZ score greater than 4. Notice a general positive correlation between harmonise (harm) and reannotate (re) axes. Also notice that exaggerated normZ scores from harmonisation were corrected by reannotation and that some intermediate hits became stronger upon reannotation. Notice further that BCL2L2 strikingly became a hypersensitising hit after reannotation but this was missed in harmonisation alone.

Conclusion.

Congratulations on reaching the end of the tutorial! You now know how to:

* run exorcise on a CRISPR library to harmonise gene symbols to the exome and to verify guides based on sequence alignment to the genome,
* download FASTQ files from SRA and work out the protospacer coordinates,
* specify the comparisons required to perform CRISPR differential analysis,
* interpret the results of a CRISPR screen, and
* appraise exorcise reannotations versus harmonisations with regard to enabling discovery.