CRISPR-UMI

Step-by-step protocol

Abstract

CRISPR-UMI extends the existing repertoire of CRSIPR-screening methods. It circumvents cell heterogeneity, a consequence of Cas9 genome editing, by scoring single cell derived clones individually. The strength of this new CRISPR screening method is its robustness towards clonal heterogeneity and clonal outliers and is therefore expected to be most useful in challenging biological screens with strong bottlenecks and clonal effects such as organoid or in-vivo screens.

This step-by-step protocol is an addition to the submitted manuscript *CRISPR-UMI: Single cell lineage tracing of pooled CRISPR/Cas9 screens*. It contains a detailed description for pooled CRISPR screening using CRISPR-UMI. It especially highlights the steps which are critical and unique to the use of CRISPR-UMI. Those critical steps are library preparation at very high complexity of up to 100million unique plasmids, and data analysis were unique guide-UMI pairs are evaluated separately.

Georg Michlits, Thomas R Burkard, Maria Novatchkova, Ulrich Elling

Ulrich.Elling@imba.oeaw.ac.at

Overview

Procedure

Library cloning

Step1: Vector design, sequencing strategy.
Step2: Barcode-library cloning (complexity 1 million).
Step3: Guide selection
Step4: sgRNA cloning (complexity 100 million).

Screening

Step1: Generation of virus in PlatE cells.
Step2: Execution of the screen.
Step3: Genomic DNA isolation, PCR amplification and next generation sequencing.

Data analysis

Step1: Assignment and counting of Sequencing reads.
Step2: Negative selection; CRISPR-UMI pipeline
Step3: Positive selection; Incidence vs abundance analysis.
Step4: Clonal size estimation in reprogramming screen
Step5: Comparison of CRISPR-UMI vs conventional CRISPR-Screening

Costs and benefits of CRISPR-UMI

References

Resources:

All Resources (Scripts, Sequences, Files, Tables) are available in the online version of this protocol under “Supplementary Information”. Deep sequencing data are available at NCBI Sequence Read Archive PRJNA383356 or at the link http://mendel.imp.ac.at/SEQUENCES/Michlits/.

Procedure:

Library cloning

Step 1: Vector design and sequencing strategy

General notes:

The Vector backbone contains ampicillin resistance for amplification in bacteria, viral packaging sequence Psi and long terminal repeats (LTRs) for generation of retrovirus. However, a lentiviral backbone can be equally used. The sgRNA cassette contains a U6 promotor and cloning site for CRISPR-guides (Step 4), improved CRISPR-Scaffold as described [1] and PGK Neo^R for selection. Cloning of P5 and P7 Illumina adaptor sequences into the vector backbone allows direct Sequencing of the viral cassette.

The essential modification for CRISPR-UMI is the integration of random sequences termed barcodes (barcodes in combination with sgRNA make the UMIs Unique molecular identifiers) and the illumina i7 ('index') primer binding site for barcode-sequencing. A PCR product reaching from illumina P5 Adaptor to illumina P7 adaptor can be used directly for next generation sequencing on an Illumina HiSeq2500 sequencer using dual indexing. Illumina's "read 1" is read with a custom primer (see Table 1 Primer_oligos.xls) and gives the CRISPR-guide sequence, "index1-read" gives the barcode sequence, and "index2-read" the experimental index to differentiate between samples (e.g. treated vs control, or replicas).

A further modification is flanking the PCR-Amplicon with Pac-I restriction sites which enable enrichment of the integrated cassette from genomic DNA by performing size selective precipitation on magnetic beads.

Library cloning is a two-step cloning process. First, random nucleotides (10nt) at a complexity of about 1 million later referred to as barcodes (BCs) are cloned in to the vector backbone (here referred to as step 2), then CRISPR-guides are cloned into that barcode-library (here referred to as step 3) reaching library complexities of 100 million due to combining barcodes and guides.

Comment: The illumina i7 binding site is usually used for reading out the experimental index to differentiate between samples, we make use of illumina's dual-indexing approach were a second (in the case of CRISPR-UMI the only) experimental index can be read adjacent to the P5 adaptor.

Resources:

- Sequence 1 CRISPR-UMI-library plasmid: CRISPR-UMI-library.gbk

  CRISPR-UMI Cassette:
  TTAATTAACCAATGATACGGCGACCACCGAGATCTACACGACCAGcgagggcctatttcccatgattccttcata
  tttgcatatacgatacaaggctgttagagagataattggaattaatttgactgtaaacacaaagatattagtacaaaatacgtgacgt
  agaaagtaataatttcttgggtagtttgcagttttaaaattatgttttaaaatggactatcatatgcttaccgtaacttgaaagtatttc
  gatttcttggctttatatatcttGTGGAAAGGACGAAACACCgNNNNNNNNNNNNNNNNNNNNgtttaagagc
  tatgctgGAAAcagcatagcaagtttaaataaggctagtccgttatcaacttgaaaaagtggcaccgagtcggtgcTTTTTTgttt
  tagagctagaaatagcaagttaaaataaggctagtccgtTTTTagcgcgtgcgccaattctgcagacaaatggctctagaGATC
  GGAAGAGCACACGTCTGAACTCCAGTCACNNNNNNNNNNGtcctcatctgagagctactcatcaacggtATC
  TCGTATGCCGTCTTaTGCTTGTTAATTAA

Step2: Barcode-library cloning (complexity 1 million).

This library cloning step introduces random nucleotides of a length of 10bp downstream of Illuminas i7 primer binding site together with the P7 Adaptor into the vector backbone. While the theoretical complexity of a 10bp random sequence is limited to about 10^6 variations. Cloning complexity should be at least 1 million but higher complexities are desirable.

Resources:

- Sequence 2 Vector_backbone_CRISPR-UMI.gbk
- Sequence 3 Barcode-oligo: Barcode-oligo.gbk (4nmol Ultramere® IDT)
- Table 1 Primer_oligos.xks
- Sequence 4 Product: Barcode-library.gbk

Materials:

- XbaI (NEB)
- MfeI (NEB)
- EcoRI (NEB)
- rSAP (NEB)
- Gel Extraction Kit (QIAGEN)
- PCR Purification Kit (QIAGEN)
- Phusion® Polymerase (NEB)
- T4 Ligase (NEB M0202)
- XL-1 Blue Electrocompetent cells (Agilent)
- SOB 20g/L Bacto Tryptone BD, 5g/L Yeast Extract BD, 10mM NaCl Merck, 2.5mM KCl Sigma Aldrich
- NucleoBond® Xtra Maxi (Macherey-Nagel)

Step-by-step:

- Digest CRISPR-UMI_plasmid_0 with XbaI and MfeI (NEB).
- Dephosphorylate the fragments with rSAP (according to NEB recommendations).
- Separate plasmid fragments on a 1% agarose gel, the plasmid backbone loses the 1.5kb stuffer. Excise the 6.9 kb fragment and isolated DNA using QIAGEN gel extraction kit. This is the vector backbone for UMI-ligation. Comment: We changed the QIAGEN standard protocol and did 2 washes with 700ul Wash Buffer per column. We invert the column to make sure to also wash lid and all walls of the spin column sufficiently.
- Carry out a second strand synthesis with 0.5nmol of barcode-oligo using UMI_RV_primer and Phusion® polymerase. Pool the reactions and purify product using QIAquick PCR purification kit.
- Digest the double stranded UMI-oligo with XbaI and EcoRI.
- Separate UMI-oligo fragments on a 2% Agarose Gel. Excise the 111 bp Fragment from the gel and isolated DNA using QIAGEN Gel Extraction Kit.
- Ethanol precipitate the 111bp UMI-oligo fragment, wash with 70% ethanol and dissolve in Tris-EDTA. This is the insert for UMI-ligation
- Ligate vector backbone and insert at a V:I ratio of 1:4, ligate a total amount of 1 ug Vector. Using T4 DNA ligase (NEB M02020). Setup and pool 10 reactions a 10ul.
- Digest the ligation in 10x volumes Cutsmart buffer (NEB) with EcoRI (NEB) and MfeI (NEB) overnight. This step is supposed to linearize and remove unwanted side products
  Comment: The 10x dilution is necessary to dilute the DTT present in the NEB T4 Ligase buffer.
- Ethanol precipitate the digested ligation product, wash and dissolve in sterile water. Comment: Alternatively we also used Mini-Elute columns (QIAGEN) to purify ligation product successfully, if doing so ensure that you wash well with 70% Ethanol (2x 10min incubation) to remove salts.

- Electroporate into XL-1 Blue electrocompetent cells (Agilent) according to manufacturers recommendation. Comments: recommendations say to incubate cells for 1h in recovery medium. We incubated only 40min to avoid an overestimation of electroporation efficiency due to cell proliferation. Alternatively, you can prepare your own electrocompetent cells in large quantities, grow XL-1Blue cells in SOB (20g/L Bacto Tryptone BD, 5g/L Yeast Extract BD, 10mM NaCl Merck, 2.5mM KCl Sigma Aldrich) to an OD of 0.9-0.99. Continue work on ice. centrifuge cells at 4C 2810xg 10min and very carefully resuspend cells in 10% Glycerol in water (hypotonic buffer). Rolling the pellet on a roller drum in a 50mL Falcon tube with about 15ml 10% Glycerol at 4°C for 5min-10min is sufficient, do not pipette up and down – this will kill the cells. Repeat centrifugation and 10% glycerol wash twice to remove all residual medium. Very carefully resuspend cells in as little 10% glycerol solution as possible. We used 80ul XL1 blue cells in 0.2cm electroporation cuvettes with 2.5kV 25μF and 200Ω and electroporated about 100ng Ligation reaction per electroporation. We pooled 3 – 10 electroporations per library subpool to achieve necessary electroporation efficiency.
- After 40min recovery at 37°C cells were spread on a 245x245mm LB-Amp-Agar dishes. Dilutions were also spread on small LB-Amp-Agar dishes to estimate cloning efficiency. We aim for 1-5 million cfu in order to have even representation of UMIs
- Dishes were incubated overnight at 37°C (no more than 10h). A dense very even bacterial lawn should grow on the plates with no single colonies visible
- Scrape the cells of the dish and rinse the dish 2 times with LB-Amp.
- Collect all cells and grow them for 2-4 h in 2L LB-Amp. OD should go from about 0.4 to 2.0)
- Freeze 10% of the UMI-library carrying XL-1 as viable glycerol stocks
- Isolate plasmid DNA from the rest of the suspension using Macherey Nagel NucleoBond® Xtra Maxi Kit. Comment: Note that the vector backbone is of low copy-number and high volumes of Resuspension-/Lysis-/Neutralization - Buffer are required to isolate plasmid DNA
- This UMI-plasmid-library is the starting plasmid for golden gate library cloning of CRISPR-UMI libraries (Step 4) and termed UMI_library (see Sequence 4 UMI_library.gbk)

Step 3: Guide selection

General notes:

sgRNAs targeting mouse nuclear genes as well as drugged orthologues and a set of hand selected genes with 4 sgRNAs per gene (5 sgRNAs per gene for the subset drugged genes) were selected by a bioinformatics pipeline. We aimed to design a guide selection algorithm taking both guide efficiency as well as biological effect due to gene structure into account. The basis of the guide selection is the activity score as described by Doench et al.[1]. Additionally, we identified properties of each guide and exon under consideration and penalized the Doench score accordingly. We identified all exonic PAM sites in the mouse genome mm10 [2]. We excluded sgRNAs that are incompatible with our cloning strategy (contain: GAAGAC, GTCTCC, CTCGAG, CGTCTC or GAGACG, start with: AAGAC or end with: CTCGA). We then calculated Doench-scores for all potential sgRNAs. We penalized the Doench-scores based on heuristic rules that aim to select sgRNAs which most likely lead to LOF phenotypes. Those rules include exon properties such as presence or absence of protein domains annotated in Pfam database [3], exon size, and whether or not exon length is a multiple of 3bp. Then we created penalties for exon distribution, to spread sgRNAs over many exons where only the sgRNA with the best Doench score per exon does not get penalized. We also avoided sgRNAs that are less than 4nt away from another better scoring sgRNA. Furthermore, we penalized sgRNAs that cut DNA upstream of a possible alternative ATG start codon and sgRNAs that cut in exons that are not common to all annotated transcripts from that locus. We avoided sgRNAs that contain a stretch of 4 or more T in a row which would act as a Pol-III Terminator. We calculated a distance-penalty based on the distance from the sgRNA to the transcriptional start ranging from 1 to 0.5. Then we calculated a simple off-target prediction (see associated publication) against all exonic sequences containing a PAM site. The off-target prediction scores weight mismatches by position in the sgRNA sequence [4,5]. We re-ranked the penalized Doench score including the off-target analysis and picked the top 4 sgRNAs per gene (the top 5 sgRNAs for Druggable genes) for chip oligo synthesis (CustomArray Inc.). For negative control guides we used a published list of human control guides [6] and removed all guides which had a perfect match against the mouse genome. We included a total of 112 control guides into our mouse library targeting 6560 genes.

Resources:

Scripts:

- Script 1 searchCRISPRcut.pl: search genome for valid gDNA sites (NGG)
- Script 2 correctExonNum.pl: Reverse exon number of (-)-strand and sort
- Script 3 getFasta.R: get FASTA sequences and position of the 2nd ATG, a potential alternative start site.
- Script 4 properties.new.pl: retrieve and calculate biological effect score.
- Script 5 calcCDSofftarget.pl: calculate offtarget score
- Script 6 mergeData.2MM.new.pl: merge and penalizes the Doench score to give metascore. Guides are selected based on this metascore.
- Script 7 TTTT_problem.py: identifies and sorts back guides with 4 or more T in a row
- Script 8 reduce_to_top_x_guides.py: generates a shorter file with only top x guides per gene
- Script 9 select_genes.py: this file requires a genelist input and selects top x guides per gene from guide lists

- Script 10 custom_array_order.py: this script adds custom upstream and downstream sequences to the chosen guides and generates a file as required for ordering from CustomArray.

Downloaded files from UCSC:

- refGene.txtmm10.fa

Files, Intermediate files and results:

- File 1 mouse_neg_contr_list.xlsx: A list of published human Ctrl guides [7]. Ctrl guides with perfect matches to the mouse genome were sorted out.
- File 2 Druggable_genes_toronto.ms.txt: An example list for gene selection, contains mouse analogs of human genes that can be targeted with drugs.
- File 3 Ms_druggable.txt: An example list of the top 4 guides per gene against “Ms_druggable” genes.
- File 4 Ms_druggable_with_negctrl.txt: An example top 5 list after adding ctrl guides manually
- File 5 Ms_druggable_order.txt: Example file as it is used for ordering oligo-pools from CustomArray. Multiple such files (library subpools) were combined and ordered on a 12K oligo-chip.

Step-by-Step:

- Download FASTA of the mm10 genome (here referred to as "mm10.fa")
- Index the FASTA with samtools

```
samtools faidx mm10.fa
```

- Extract all possible cut regions (14-20nt) as BED file since the cut of gDNAs is between 16./17. or 17./18. nucleotide of the gDNA (20nt before NGG)

```
mkdir tmp
perl searchCRISPRcut.pl | sort -k1,1 -k2,2n > tmp/cut.bed
```

- Download refGene.txt.gz from http://hgdownload.cse.ucsc.edu/goldenPath/mm10/database/
- Gunzipped mouse (mm10) "refGene.txt" from 6.5.2015 (Filedate: 3.5.2015) provided
- Extract columns 2 to end, convert to GTF with genePredToGTF (kentUtils), grep coding transcripts ("NM_")

```
cut -f 2- refGene.txt | genePredToGtf file stdin stdout | grep NM_ | sed "s/\tstdin\t/\tprotein_coding\t/" >
tmp/refGene.gtf
```

- Correct exon number with perl script

```
perl correctExonNum.pl tmp/refGene.gtf > tmp/mm10_refGene.gtf
```

- grep CDS annotation

```
grep $'\t'CDS$'\t' tmp/mm10_refGene.gtf > tmp/mm10_refSeq.cds.gtf
```

- Add mod3 (modulo 3) & length of the CDS exons

```
awk -vFS="\t" -vOFS="\t" '{L=($5-$4)+1; OUT=$9" mod3 \""L%3"\"; length \""L"\";"; print $1, $2, $3, $4, $5, $6, $7, $8, OUT}' tmp/mm10_refSeq.cds.gtf > tmp/mm10_refSeq.cdsMod3Len.gtf
```

- Generate an sqlite database "TxDB" in R version 3.1.2, since in newer R version makeTxDbFromGFF resulted in wrong CDSs due to wrong assignment of exon_rank. In this step duplicated genes are excluded.
- Extract CDS and protein sequence. Calculate 2nd ATG position

```
Rscript getFasta.R
```

- Run pfam_scan.pl from http://pfam.xfam.org/ with HMMER3.1b on the protein FASTA "tmp/mm10_refGene.protein.fa".
- Extract properties of CDS exons which overlap (BEDtools) with the gDNA cut region. Columns content:
    - Gene ID
    - gDNA region (4N-gDNA(20)-NGG-3N)
    - exon length
    - Dividable by 3 (TRUE/FALSE)
    - All protein coding transcripts overlapping (TRUE/FALSE)
    - exon IDs
    - PFAM hits
    - distance penalty
    - distance
    - transcript length
    - alternative ATG

```
intersectBed -a tmp/cut.bed -b tmp/mm10_refSeq.cdsMod3Len.gtf -f 1 -wo | properties.new.pl > tmp/cut.propertie.new.txt
```

- Extract cuts overlapping with CDS regions

```
intersectBed -a tmp/cut.bed -b tmp/mm10_refSeq.cds.gtf -f 1 -wo | perl -pe 'chomp; @c = split "\t"; $g = $_; $g =~ s/.*gene_id "(.*?)";.*/\1/; $_ = $c[0] . "\t" . $c[1] . "\t" . $c[2] . "\t" . $g . "\t" . $c[4] . "\t" . $c[5] . "\n";' | sort -k1,1 -k2,2n -k3,3n -k6,6 -k4,4 -u > tmp/cut.cds.bed
```

- Slop BED to gDNA region (4N-gDNA(20)-NGG-3N) (BEDtools)

```
slopBed -i tmp/cut.cds.bed -g mm10.fa.fai -l 18 -r 6 -s > tmp/cut.guide30.bed
```

- Extract sequence of gDNA region (4N-gDNA(20)-NGG-3N) (BEDtools)

```
fastaFromBed -fi mm10.fa -bed tmp/cut.guide30.bed -s -fo tmp/cut.guide30.fa
```

- Extract gDNA (20nt) FASTA sequence

```
perl -pe 'if (! />/) { $_=substr($_, 4, 20) . "\n" }' tmp/cut.guide30.fa > tmp/guide.fa
```

- Align gDNA (20nt) with bowtie to genome. 2 mismatches allowed. Submit to cluster if possible.

  bowtie-build mm10.fa tmp/mm10

  bowtie tmp/mm10 -f guide.fa -S -a -v 2 --tryhard --un guide.2MM.nm --max guide.2MM.rep | samtools view -bS -F 0x0004 - > tmp/guide.2MM.bam

- Extract alignments which are followed by an "NGG" (BEDtools). Create one file for an edit distance of 0 and one unfiltered.

  bamToBed -i tmp/guide.2MM.bam -ed | awk '$5 == "0" { print $0 }' | slopBed -i stdin -g mm10.fa.fai -l 0 -r 3 -s | fastaFromBed -fi mm10.fa -bed stdin -s -fo stdout -name | paste - - | grep -i GG$ | cut -f 1 | sort | uniq -c > tmp/guide.2MM.NM0count.txt

  bamToBed -i tmp/guide.2MM.bam -ed | slopBed -i stdin -g mm10.fa.fai -l 0 -r 3 -s | fastaFromBed -fi mm10.fa -bed stdin -s -fo stdout -name | paste - - | grep -i GG$ | cut -f 1 | sort | uniq -c > tmp/guide.2MM.NM2count.txt

- Calculate Doench score for all gDNA regions in "tmp/cut.guide30.fa". doenchScore.py is available at https://github.com/maximilianh/crisporWebsite/blob/master/doenchScore.py. python doenchScore.py tmp/cut.guide30.fa > cut.guide30.score.txt

- Merge all above generated data and calculate metascore. Exclude sequence restriction enzyme site, with less than 30nt, with none ATGC letters.

  mergeData.2MM.new.pl > mergeData.2MM.new.txt 2> mergeData.2MM.new.err

- Calculate offtarget score. Slow, hence it's implemented here with gridengine on a cluster.

  mkdir tmp/calcCDSofftarget/

  qsubcmd -sync y -N offtarget -o log/ -e log/ -t 1:10103 'calcCDSofftarget.pl -t $SGE_TASK_ID'

  cat tmp/calcCDSofftarget/offTarget.* > offtarget.txt

- Re-run Script 6 mergeData.2MM.new.pl to generate final output table including offtarget scores. Final ouputfile "mergeData.2MM.new.txt. Column content:
  - genename
  - pos: position of the gDNA region
  - geneOverlap: overlapping genes
  - seq [4N-gDNA(20)-NGG-3N]: sequence of gDNA region
  - score: Doench score
  - metascore: Metascore of pipeline
  - distanceScore: distance score
  - min CDS distance: minimal distance to CDS start
  - CDS length: CDS length
  - exon length: exon length
  - mod3: dividable by 3 (TRUE/FALSE)

- exon: exon IDs
- exonUsage: usage of the same exon
- pfam: PFAM hits
- ATG: alternative ATG
- MM2: offtarget with <=2 mismatches (TRUE/FALSE)
- commonCDS: common to all CDSs (TRUE/FALSE)
- OT95: hits with OT95 score
- OT90: hits with OT90 score
- OT80: hits with OT95 score

mergeData.2MM.new.pl > mergeData.2MM.new.txt 2> mergeData.2MM.new.err

- Run python script 7 TTTT_problem.py: script identifies guides with 4 or more T in a row and lists them last – it kicks our guides that contain a reverse BbSI site formed together with the upstream G.

python 1_TTTT_problem.py

enter inputfilename (e.g. mergeData.2MM.new.txt): mergeData.2MM.new.txt

enter outputfilename (e.g. Ms_top50_resorted.txt): Ms_top50_resorted.txt

finished

- Run python script 8 reduce_to_top_x_guides.py: script generates a new file with top x guides for every gene.

python 2_reduce_to_top_x_guides.py

enter inputfile (e.g. Ms_top50_resorted.txt): Ms_top50_resorted.txt

enter output_filename (e.g. Ms_top4.txt): Ms_top4.txt

number of guides selected per gene (4): 4

finished

- Generate a list of genes to be targeted (e.g. File 2 Druggable_genes_toronto.ms.txt)
- Run python script 9 select_genes.py: Script generates a new file containing only the guides of target genes

python 3_select_genes.py

enter filename of target gene_list (e.g. druggable_genes_toronto.ms.txt):

druggable_genes_toronto.ms.txt

input filename (e.g. Ms_top4.txt): Ms_top4.txt

enter output filename (e.g. Ms_druggable.txt): Ms_druggable.txt

number of guides that go into screen. (4): 4

guides available: 3548

guides not available: 52

finished

- Manually copy ctrl guides to the set. See mouse_neg_contr_list.xlsx and an example of the intermediate file with control guides is Ms_druggable_with_negctrl.txt
- Run python script 10 custom_array_order.py: assembles 20nt guides with upstream and downstream sequences and generates a file with 77nt pooled oligos as required for CustomArray ordering. Upstream and downstream sequences can be obtained from Table 1 Primers_Oligos.xlsx

  python 4_custom_array_order.py

  enter input filename (e.g. Ms_druggable_with_negctrl.txt): Ms_druggable_with_negctrl.txt

  enter output filename (e.g. Ms_druggable_order.txt): Ms_druggable_order.txt

  enter upstream sequence (e.g. GATTACATGGTCAGACGAAGACgaCACCG):

  GATTACATGGTCAGACGAAGACgaCACCG

  enter downstream Sequence (e.g. GTTTctGTCTTCTTACCACACCAGTCGA):

  GTTTctGTCTTCTTACCACACCAGTCGA

  enter subpool name (e.g. Ms_Druggable): Ms_druggable_order

  total genes: 924 including ctrl guides

  total guides: 3585 including ctrl guides

- Guides were ordered as 12K oligo-pools from CustomArray Inc.

Step 4: sgRNA cloning (complexity 100 million).

General Notes:

For CRISPR-UMI library cloning a complex insert (e.g. 26500 sgRNAs PCR amplified from chip-oligo synthesis) is cloned into a complex vector backbone (containing up to 1 million different UMIs). In every possible combination, this would allow a theoretical complexity of 26.5 billion (when using 26500 guides) unique CRISPR-UMI pairs. Cloning efficiency should be at least 1000X per guide (i.e. 30million for 30.000 guides) to generate a library complex enough for CRISPR-UMI. We aimed to generate Libraries of a complexity of about 85 million for 26500 guides.

Resources:

- sgRNA cloning starting vector: Sequence 4 Barcode-library.gbk
- All primer sequences: Table 1 Primers_oligos.xlsx
- Final plasmid constructs: Sequence 1 CRISPR-UMI-library.gbk

Materials:

- BbsI (NEB)
- XhoI (NEB)
- rSAP (NEB)
- Gel Extraction Kit (QIAGEN)
- PCR Purification Kit (QIAGEN)
- Phusion® Polymerase (NEB)
- T4 Ligase (NEB M0202)
- XL-1 Blue Electrocompetent cells (Agilent)
- Phenol:Chloroform:Isoamylalcohol = 25:24:1 (Carl Roth)
- Chloroform (Sigma Aldrich)
- NucleoBond® Xtra Maxi (Macherey-Nagel)

Step-by-step:

- Amplify subpools from guide-oligo-pool with subpool specific primers (see Table 1 Primers_oligos.xlsx). For every subpool run 8x 50ul PCR-reactions with 2ng PCR template per reaction (2ng of only the subpool specific oligos), using Phusion® Polymerase, run 10cycles PCR: 95°C 2min, (95°C 10sec, 58°C 20sec, 72°C 10sec, 10x), 72°C 2min.
- Pool the reactions and purify product using QIAquick PCR purification kit. This is the Insert for sgRNA golden gate cloning. Comment: We changed the QIAGEN standard protocol and did 2 washes with 700ul Wash Buffer per column. We invert the column to make sure to also wash lid and all walls of the spin column sufficiently.
- Digest the starting vector (Sequence 4 Barcode-library.gbk) with BbsI (NEB).
- Dephosphorylate the fragments with rSAP (NEB).
- Separate plasmid fragments on a 1% agarose gel, excise the linearized 7.0 kb fragment and isolated DNA using QIAGEN gel extraction kit. This is the vector backbone for barcode-ligation. Comment: We changed the QIAGEN standard protocol and did 2 washes with 700ul Wash Buffer per column. We invert the column to make sure to also wash lid and all walls of the spin column sufficiently.
- Digest 300ng insert in a volume of 10ul with 1ul BbsI in cutsmart (NEB) buffer, then dilute to 3ng/ul.

- Carry out golden gate ligation. In a volume of 50ul mix 500ng vector backbone with 16.5ng Insert (V:I = 1:3) add 10U BbsI (NEB) and 500U T4 Ligase (NEB) in T4 Ligase Buffer (NEB). Run goldengate program on (37C5min, 16C5min, 35x), 50C10min, 4Chold.
- To 50ul ligation reaction add 404ul water, 45ul 10X Cutsmart (NEB) and 1ul XhoI (20U, NEB). Incubate at 37°C for 4h.
- Extract 2 times with Phenol:Chloroform:Isoamylalcohol = 25:24:1
- Extract once with Chloroform (no Isoamylalcohol)
- Ethanol precipitate DNA and wash 2 times with 70% EtOH
  Comment: Alternatively to phenol-extraction and ethanol precipitation we also used Mini-Elute columns (QIAGEN) to purify ligation product successfully, if doing so ensure that you wash well with 70% Ethanol (2x 10min incubation) to remove salts.
- Dissolve DNA in 5-10ul and use 100ng per electroporation. Electroporate DNA into electrocompetent XL-1 Blue cells. We did multiple (5-10) electroporations using 80ul homemade XL-1 Blue electrocompent cells in a 0.2cm kuvette on a BioRad Gene Pulser® II settings: 10uF, 600, 2.5kV, 200Ω, 25μF. . Comments: recommendations say to incubate cells for 1h in recovery medium. We incubated only 40min to avoid an overestimation of electroporation efficiency due to cell proliferation. Alternatively, you can prepare your own electro competent cells in large quantities, grow XL-1Blue cells in SOB (20g/L Bacto Tryptone BD, 5g/L Yeast Extract BD, 10mM NaCl Merck, 2.5mM KCl Sigma Aldrich) to an OD of 0.9-0.99. Continue work on ice. centrifuge cells at 4C 2810xg 10min and very carefully resuspend cells in 10% Glycerol in water (hypotonic buffer). Rolling the pellet on a roller drum in a 50mL Falcon tube with about 15ml 10% Glycerol at 4°C for 5min-10min is sufficient, do not pipette up and down – this will kill the cells. Repeat centrifugation and 10% glycerol wash twice to remove medium. Very carefully resuspend cells in as little 10% glycerol solution as possible. We used 80ul XL1 blue cells in 0.2cm electroporation cuvettes with 2.5kV 25μF and 200Ω and electroporated about 100ng Ligation reaction per electroporation. We pooled 3 – 10 electroporations per library subpool to achieve necessary electroporation efficiency.
- After 40min recovery at 37°C cells were spread on 245x245mm LB-Amp-Agar dishes (2mL per plate). Dilutions were also spread on small LB-Amp-Agar dishes to estimate cloning efficiency. We aim for more than 1000 cfu per sgRNA in the pool in order to have a high number of independent barcodes cloned for every guide.
- Dishes were incubated overnight at 37°C (no more than 10h). a dense bacterial lawn should grow on the plates with no single colonies visible
- Scrape the cells off the dish and rinse the dish 2 times with LB-Amp.
- Collect all cells and grow them for 2-4 h in 2L LB-Amp. OD should go from about 0.4 to 2.0)
- Freeze 10% of the CRISPR-UMI-library carrying XL-1 as viable glycerol stocks
- Isolate plasmid DNA from the rest of the suspension using Macherey Nagel NucleoBond® Xtra Maxi Kit. Comment: Note that the vector backbone is of low copy-number and high volumes of Resuspension-/Lysis-/Neutralization - Buffer are required to isolate plasmid DNA.
- This CRISPR-UMI-library is the plasmid used for generation of Retrovirus. See Sequence 1 CRISPR-UMI-library.gbk.

Screening

Step 1: Generation of virus in Plat-E cells

General notes:

We use Plat-E cells for packaging virus. Since the CRISPR-UMI plasmid library is of very high complexity (e.g. 85 million) and we want to keep complexity and even representation of individual guide-barcode pairs. We recommend to infect at least six 150mm dishes (about 250million cells) for a 26500 sgRNA library to retain the necessary complexity.

Materials:

- Platinum-E cells (Cell Biolabs RV-101)
- RV-helper plasmid: pCMV-Eco Envelope Vector (Cell Biolabs RV-112)
- HBS Buffer (280mM NaCl, 50mM HEPES, 1.5mM Na_2HPO_4, adjust pH to 7.00 with 0.5M NaOH)
- Polybrene (Sigma-Aldrich)
- G418 (Gibco)

Step-by-Step:

- Grow six 150mm cell culture dishes of PlatE cells to a confluency of about 70%.
- Cotransfect the CRISPR-UMI library with retrovirus helper plasmid using $Ca_3(PO_4)_2$ – DNA precipitate.
- Mix 380ug CRISPR-UMI library with 640ug RV-helper plasmid and 2.25ml 1M CaCl2 Solution and bring to 9ml with mono-Q sterile water
- Dropwise, add the 9ml DNA-Ca solution to 9ml of HBS buffer while vortexing.
- Dropwise add the 18ml DNA- Ca3(PO4)2 suspension to the PlatE cells.
- Incubate for 24h at 37°C
- Change medium to target medium
- Incubate for 24h at 37°C
- Collect supernatant from all plates, pool all supernatant and filter through a 0.45um Filter (Optional: Feed cells again with target medium for a second harvest after 12h).
- Freeze virus in aliquots.
- To estimate multiplicity of infection MOI of your frozen virus aliquots infect cells with different dilutions of virus in the presence of 2ug/ml polybrene. Comment: We reach an MOI of 0.1 for 1:30 dilutions of Virus (i.e. 1ml to 29ml in a 150mm dish) when infection mouse embryonic stem cells. Due to the robustness of CRSIPR-UMI it should be possible to run screens at higher multiplicity of infection, this could reduce workload and consumables in cell culture or on the other hand increase coverage and throughput of guides, but we have not tried that yet.
- After 24h of infection seed cells embryonic stem cells at 1.000 cells per plate without G418 selection (for CRISPR-UMI cassettes), and 10.000 cells per plate with G418 selection. From the number of colonies MOI can be estimated.

Step2: Execution of the screen

General notes:

A basic principle to keep in mind are cell numbers that need to be (or can be) carried through the experiment. For example, if running the screen with 30.000 guides per gene we aim to always keep at least 30million cells in the experiment (1000x representation). We grow cells from 30million to 300million and split every 2nd day at a ratio of 1:10 (keep at least 30 million cells and discard the rest). CRISPR-UMI offers 2 variations: with or without limiting dilution - clonal expansion.

With limiting dilution - clonal expansion:

In this protocol CRISPR-UMI introduces an artificial bottleneck after CRISPR-gene editing has occurred. Depending on the screen setting introducing a strong bottleneck means discarding 95-99 % of all cells and then expanding the remaining 1-5% of cells. By doing so we reach cell numbers much lower than the complexity of the CRISPR-UMI library and most cells in the experiment will carry a unique guide-barcode pair (UMI). Therefore, after expansion every UMI will carry a clonally selected uniquely repaired CRISPR cutting site. This contrasts with conventional CRISPR screens were cells carrying the same guide are heterogeneous in the way the CRISPR-cut was repaired. We recommend a limiting dilution and expansion for negative selection screens when comparing two conditions, because you can make use of multiple isogenic clones that you can compare in two settings. The cost of a limiting dilution is that the extra time required for expansion can cause shifts in representation and that under-represented guides can be lost completely from the experiment. Note that some experiments (like in-vivo screens with bottlenecks such as engraftment of cells or differentiation screens with moderate efficiency) introduce this "limiting dilution" step inherently.

Comment: Why a limiting dilution generates isogenic clones: Assume a single cell is infected with a single virus carrying a unique UMI. Before and during gene-editing this will give rise to a handful of daughter cells which all carry the same guide but generate different mutations due to random mistakes in error-prone repair mechanisms. As a consequence daughter cells are heterogeneous like in a conventional CRISPR screen. By introducing a strong enough dilution step after CRISPR mutations are set, only one daughter cell will remain in the experiment, after expanding the population again the UMI will now be unique to all "grand-daughters" and in contrast to a conventional CRISPR screen, all grand-daughters will carry the same CRISPR-mutation. Note that during the dilution-expansion step most UMIs will be completely lost with not a single daughter cell remaining in the experiment.

Without liming dilution – clonal expansion:

Not introducing a strong limiting dilution but still using CRISPR-UMI is also an option. While the benefit of isogenic clones is lost, you can still use UMIs as conceptual replicates and detect and exclude artefacts or outliers from data analysis. Also in positive selection where selection events are considered rare occasions, you can use UMIs to differentiate between incidence of an event (were the number of independent UMIs indicates the frequency of an event) and abundance (indicated by the counts per UMI which give information about the extent of the positive selection event).

Materials:

- Cell line: dox inducible Cas9 mouse embryonic stem cells AN3-12. [8]
- Embryonic stem cell medium (450 ml DMEM (Sigma D1152); 75 ml FCS (Invitrogen); 5.5 ml P/S (Sigma P0781); 5.5 ml NEAA (Sigma M7145); 5.5 ml LGlu (Sigma G7513); 5.5 ml NaPyr (Sigma S8636); 0.55 ml ME (Merck 805740; dilute 10ul bME in 2.85 ml PBS for a 1000x stock), 7.5ul LIF (Sigma; 2mg/ml)).

- Polybrene (Sigma-Aldrich)
- G418 (Gibco)
- Doxycycline (Sigma-Aldrich)

Step-by-Step:

- Day -2: Culture doxycycline inducible Cas9 cell line in 1ug/ml doxycycline for 2 days prior to infection (+dox)
- Day 0: Seed 30 times 10million cells on a 150mm dish, to give a total of 300million cells (+dox)
- Day 0: After about 5-6 h when cells are attached infect with CRIPR-UMI virus in the presence of 2ug/ml polybrene (depending on the virus titer 1-3ml per plate should give an MOI of about 0.1). Keep cells on doxycylin (+dox)
- Day 1: 20h after infection change to selection medium (Neomycin/G418 500ug/ml) and change medium every 12h, this speeds up selection and postpones the first and critical split by one day (+dox, +Neo, 2x)
- Day2: change to selection medium every 12h (this postpones the first and critical split by one day) (+dox, +Neo, 2x)
- Day3: split cells 1:4 on 45 plates continue selection changing medium every 12h (+dox, +Neo, 2x)
- Day4: keep cells on dox and selection (+dox, +Neo)
- Day5: remove dox but keep cells on Neo (-dox, +Neo)
- Day6: keep cells, wash out dox and Neo (-dox, -Neo)
- Day7: limiting dilution: split cells and seed cells (Aiming for about 200 clones per guide with 30.000 guides in the experiment seed: 6 million cells, that means going from 30 plates to 1 plate).
- Day8-10: expand cells to about 500 million cells.
- Day 11: split cells into 2 conditions: +Etoposide 3nM or + mock treatment, seed 120million cells per experiment (15 x 8million cells per 150mm dish)
- Day 12: Change medium + Etoposide 3nM or + mock treatment
- Day 13 – Day19: split cells 1:8 every 2 days, change medium every day +Etoposide 3nM or + mock treatment
- Day 19: Harvest cells. Comment: If cells are not dense e.g. If Etoposide treated cells are 50% confluent harvest only control cells and expand treated cells one more day. As a rule of thumb I harvest about 10times more cells than the number of reads I expect to get from one flowcell (currently about 250million). I typically use 1/3 of the cells for lysis, 1/3 of the cells I freeze as pellet (backup) and 1/3 of the cells I freeze in DMSO.
- Proceed with genomic DNA isolation, PCR amplification and NGS (step3)

Step 3: Genomic DNA isolation, PCR amplification and next generation sequencing

General notes:

If cell numbers arebot limiting, we recommend to harvest 3 fold more cells that the number of reads to be retrieved from NGS Sequencing. For 1 lane on a HiSeq2500 that gives about 250million reads we recommend to harvest 750million cells. More cells may be harvested as backups or frozen as live stocks. All quantities given in the protocol are for processing 750million cells. Realistically those 750million cells will be subdivided into different experimental conditions, but for the purpose of this protocol total quantities for processing 750million cells are given.

Materials:

- 2X SDS Lysis Buffer: (10mM Tris-HCl pH 8.0, 5mM EDTA, 100mM NaCl, 2% SDS, add fresh directly before use: Proteinase K from a 10X stock 10mg/ml Proteinase K stored in 50% Glycerol at -20°C. To a final concentration of 1mg/ml)
- RNaseA (100mg/ml) (Qiagen)
- Phenol:Chloroform:Isoamylacohol = 25:24:1 (Roth)
- Chloroform (Sigma)
- 5M NaCl solution (Sigma)
- RNase A (QIAGEN)
- TE, Tris-EDTA solution: (10mM Tris-HCl pH 8.0, 5mM EDTA)
- SpeedBeads™ magnetic carboxylate modified particles (GE45152105050250 Sigma Aldrich)
- PHUSION® polymerase (NEB)
- Klentaq polymerase (DNA Polymerase Technology)
- Binding Buffer (20% PEG8000 (FLUKA) 2.5M NaCl)
- Eppi-Magnet (for large quantities 15mL Falcon Magnet)
- PCR Purification Kit (QIAGEN)

Step-by-step:

DNA Isolation and Pac-I digest

- Trypsinize 750million cells.
- Wash 2x with 20mL PBS (5min 300xg).
- Resuspend pellet in 20ml MonoQ water, immediately add 20ml 2X Lysis buffer and mix well.
- Incubate at 55°C overnight.
- Add 40ul RNaseA (1000x stock 100mg/ml, to a final concentration of 0.1mg/ml).
- Incubate at 37°C for 1h shaking at 200rpm.
- Let phenol, chloroform, 5M NaCl-solution and lysates adjust to room temperature.
- To 40ml lysate add 16ml of 5M NaCl. A SDS-salt-protein pellet forms.
  Comment: This step is very critical and temperature-dependent. Do not precipitate DNA with too much NaCl. We recommend to calibrate this step with small aliquots of samples for the buffers used.
- Centrifuge at RT for 30min 3000 x g.
- Transfer supernatant to a new tube and dilute 1:3 with MonoQ water (add 112ml to a total volume of 168ml).
- Extract 2x with 1 volume (168ml) of phenol:chloroform:isoamylalcohol 25:24:1 solution.
- Extract with 1 volume (168ml) chloroform.
- Add 1 volume of isopropanol (168ml) to precipitate DNA. The expected total amount of DNA for 750million cells is approximately 7.5mg.

- Wash the DNA precipitate twice with 70% Ethanol (no centrifugation, just transfer the sticky DNA with a pipette) and shake at 37°C for 1h.
- Centrifuge at 3000 x g for 10min.
- Discard ethanol.
- Centrifuge again to spin down remaining ethanol, withdraw ethanol.
- Let DNA dry at RT for 10min.
- Add 15ml TE to the DNA pellet and place the sample in a 55°C dry incubator for 30min to evaporate remaining ethanol.
- Ensure that the DNA is well dissolved, If the DNA is not dissolved homogenously you can push it through a 27G syringe or do freeze-thaw cycles.
- Bring DNA to a concentration of 250ng/ul in cutsmart buffer (NEB) (approx. 27ml DNA solutions + 3ml cutsmart 10X buffer).
- To ensure optimal Pac-I digest we digest DNA of 2x the number of cells relative to the number of NGS reads required. That is DNA of approximately 500 million cells – 5 mg DNA.
- Add 60ul of Pac-I (NEB) to the DNA and digest for 24h. Large quantities of Pac-I are required to efficiently cut all Pac-I sites.
- Add additional 60ul Pac-I (NEB) to the DNA and digest for another 24h.
- Proceed with size selective precipitations using SpeedBeads™.

Size selective precipitation using SpeedBeads™

- Vortex magnetic beads well to suspend them equally.
- Pipette 9ml of magnetic beads into 15ml tubes.
- Wash magnetic beads 3x with TE.
- Dilute the DNA sample 1:2 with TE (total volume of 60ml) to reach a final concentration of approximately 125ng/ul and half of the working concentration of cutsmart buffer.
- Resuspend the beads (9ml) with the DNA sample (60ml).
- Add 22.5ml binding buffer and immediately pipette up and down to mix sample. This step is crucial and very sensitive, the exact amount of binding buffer to be used may be calibrated for every batch of binding buffer.
- Allow DNA to precipitate 30min at 4°C. Fragments >2kb in size will precipitate and be adsorbed onto the magnetic beads. Critical step, be accurate with time and temperature.
- During precipitation prepare more magnetic beads. Wash another 1.5ml magnetic beads with TE (3x).
- Put the tube (after 30min at 4°C) with precipitated >2kb fragments into a magnet rack to separate beads from supernatant. Withdraw Supernatant.
    - Magnetic Beads: DNA fragments >2kb, precipitated on the beads can be washed with 70% Ethanol and eluted from the beads using TE.
- Supernatant: Resuspend the freshly washed magnetic beads with the supernatant (containing the DNA-fragments <2kb including the CRIPSR-UMI cassette for PCR amplification).
- Add 36ml binding buffer and incubate at 4°C for 30min to precipitate the smaller fragments (<2kb including the CRISPR-UMI cassette) onto the beads.
- Put the tube with precipitated >2kb fragments into a magnet rack to separate beads from supernatant. This may take 5-10min due to higher viscosity of the sample.
- Wash beads 2x with 70%ethanol (60ml).
- Remove ethanol as good as possible.
- Let the tubes stand on RT for 5-10min.

- Add TE (1.5ml) and put the samples into a 55°C dry incubator for 10min with an open lid to evaporate ethanol and dissolve DNA from the beads.
- Put the tube back into the Magnet-rack and withdraw the supernatant containing the dissolved <2kb DNA fragments including the CRISPR-UMI cassette. The Supernatant is the template for PCR amplification.

PCR amplification

- PCR amplify the CRISPR-UMI cassette using forward primer CrSc_FW1-42 (for up to 42 individual samples) using multiple PCR reaction (for a whole screen I use about 600 PCR reaction of 50ul each) about 25cycles PCR are usually sufficient. For primers see Table 1 Primer_oligos.xls.
- We use a 1:1 mix of Klentaq and Phusion Polymerases which are commercially available.
- PCR settings: 95°C 3min ,(95°C 10sec,59°C 20sec,72°C 30sec, 25x) 72°C 3min.
- Pool all PCR reactions per sample and purify the PCR product using a PCR purification kit (Qiagen).
- Determine DNA concentration and pool all samples. Pool different samples proportional to the read depth you want to get per sample, e.g. equal amounts of control and treated cells in negative selection screens.
- Run the pooled DNA gel on a 2% agarose gel to separate the 589bp target amplicon from primer-dimers and by-products
- Excise the 589bp product and isolate the DNA using Gel extract kit (Qiagen).
  This product is the sample for next generation sequencing

Next generation sequencing

- We are sequencing samples on a HiSeq2500 sequencer
- We use a 44bp custom primer CRISPR-UMI_NGS_custom_primer (see Primers_oligos.xlsx) at a concentration of 5uM, that is 10 times the recommended concentration of custom sequencing primer.
- We do not spike in PhiX DNA
- For clustering we use the cBOT program, using an extra 8 tube strip for clustering. By doing so we avoid mixing our custom primer with the illumina primer mix.
- The read mode is SR50, dual indexing. Read at least 10nt for index 1 and at least 6nt for index 2.

Data analysis

Step 1: Assignment and counting of sequencing reads

General notes:

We use samtools, fastx-toolkit and bowtie to assign guides and experiments to sequencing reads and then count sequencing reads of UMIs. This section describes how we convert the bamfile from Illumina sequencing to a tab separated text file with the columns:

- Guidename
- Samplename (e.g. ctrl_1, treated_1)
- Barcode Sequence
- Read count

In the later sections of this protocol this tab separated text file will be the input and starting point of more specialized analysis scripts.

Resources:

Example Sequence files:
Example BAM files from the associated publication are available via NCBI Sequence Read Archive PRJNA383356 or at the link http://mendel.imp.ac.at/SEQUENCES/Michlits/.
BAM file structure:

Read1: A 50bp sequence read, the first 20nt are the guide sequence the following 30nt are constant sequence from the scaffold. A drop in read quality after base 20 is expected.
Index1: A 10bp sequence read, that we term barcode.
Index2: A 6bp sequence read, the experimental index that was introduced with the FW-PCR primer

Scripts

- Script 11 Merge_2files.py:
  This script combines any two text files (e.g. SAM files). It simply reads in 2 text files and writes out one long text file. It allows to analysis multiple BAM sequencing files.
- Script 12 Merge_bowtie_shortsam.py:
  This script reads in the output file from bowtie and the sequencing file (see step-by step protocol) and assigns guidenames to reads.
- Script 13 make_Masterdict.py:
  This script assigns experiment names and counts the number of sequencing reads for every guide-barcode combination(=UMI) in every experiment. It generates an output file that serves as a starting point for CRISPR-UMI analysis pipelines (see Steps 2 and 3). In this protocol we refer to the output file as Masterdictionary because they can easily be read into a dictionary that is useful for processing the data.

Dependencies:

- Python
- Scipy
- Samtools
- fastx-toolkit/0.0.13
- bowtie/0.12.9

Files and results:

- File 6: 30kMs.tsv:
  A tab separated text file of all guides in the library (column 1 name, column2 sequence).
- File 7: 30kMs.fa:
  Fasta file of all guides in the library.
- File 8: indEtop.csv:
  list of indices and corresponding experiments (column 2 sample name, column 5 6nt index).

Step-by-Step:

Extraction of reads from bamfile

- Run bowtie-build with a fasta file containing all guides in library.

```
bowtie-build 30kMs.fa 30kguides
```

- Extract guide sequences with bowtie.

```
bowtie -m 1 --best --strata -S -p 16 -v 1 30kguides --max 30k_lane8_bw1MM.notUniq.fastq <(
bamToFastq -i CAR6CANXX_8_20170310B_20170311.bam -fq /dev/stdout | fastx_trimmer -Q33 -l 20 ) |
samtools view -S -F 0x0004 - > 30k_lane8_bw1MM.sam
[samopen] SAM header is present: 26936 sequences.
# reads processed: 156844748
# reads with at least one reported alignment: 141427354 (90.17%)
# reads that failed to align: 14905378 (9.50%)
# reads with alignments suppressed due to -m: 512016 (0.33%)
Reported 141427354 alignments to 1 output stream(s)
```

- Generate a shorter sam file from sequencing file containing only the information required for further processing.

```
samtools view -h filename.bam | cut -f 1,10,12,14 >filename_short.sam
```

- If multiple sequencing files are used for data analysis merge the output files from bowtie and the short_sam files, run Script 11 Merge_2files.py

```
python Merge_2files.py
enter filename_2 (path): '30k_lane7_short.sam'
enter filename_1 (path): '30k_lane8_short.sam'
enter merged file_filename: '30k_78_short.sam'
reading in file1
finished
reading in file2
finished
```

- To extract the bowtie results from the Sequencing sam file run Script 12 Merge_bowtie_shortsam.py

```
python Merge_bowtie_shortsam.py
enter bowtie result file path: '30k_1278_bw1MM.sam'
enter short (samfile columns 1,10,12,14) sam file path: 'short_1728.sam'
enter output filename (e.g. out_0_samplename.txt)'out_0_30kMsetopo.txt'
reading in bowtie file
total lines = 429453604
mm0 = 357344470 mm1 = 72109133  mm2 = 0
Processed 10 million
.
.
processed 550 million
processed 560 million
reads not assigned to guide = 134350699      reads assigned to guide = 429453603
```

- Run Script 13 make_Masterdict.py: Script asks for input and output filenames and allows variation of mismatches allowed in the index sequence and the guide sequence (in the example shown here index MM is set to 0 and guide MM to 1.

```
python make_Masterdict.py
enter input filename (e.g. out_0_30kMsEtopo.txt): 'out_0_30kMs_etopo.txt'
enter output filename (MD_30kMsEtopo_i0MMsg1MM.txt): 'MD_4lanesi0MMsg1MM.txt'
enter indexfilename path (e.g. indEtop.csv): 'indEtop.csv'
enter guidefile_filename (e.g. 30kMs_unique.tsv): '30kMs_unique.tsv'
enter number of index MM allowed (0): 0
enter number of guide MM allowed (1): 1
Index dictionaries done
reading NGS.sam file
generating MD_dict
10 million
20 million
.
.
420 million
total lines: 429453603
good lines (index and guide match <= max MM): 379554126
kicked lines: 49899477
kicked (bad index): 49899477
kicked (guide MM): 0
```

writing Masterdict

writing Masterdict finished

- The generated file with the prefix MD for ‘Masterdict’ is the starting file used for following analysis scripts. It is a tab separated text file, column content:
    - Guidename
    - Samplename (e.g. ctrl_1, treated_1)
    - Barcode Sequence (10nt)
    - Sequece read count

Step2: Negative selection, CRISPR-UMI pipeline

General notes:

The main purpose of this section of data analysis is to document and describe the scripts and calculations that were used to evaluate hits in a negative depletion setting. The key script in the analysis pipeline is CRISPR_UMI.py. It prepares input files for MAGeCK [9] for both conventional CRISPR analysis (ignoring BCs) and CRISPR-UMI analysis in parallel. By running and algorithm called POPTOP(x) prior to analysis it also allows to remove a certain number of clones (x) per guide, always removing the clones with the highest read support (ctrl and treated condition taken together), prior to analysis. This analysis allowed us to show that some of the clones with highest read support are responsible for false positive signals in conventional CRISPR-screening and that CRISPR-UMI screening is robust towards those outliers. For CRISPR-UMI individual clones are evaluated using MAGeCK to give a depletion score for guides and CRISPR-UMI.py gives median depletion (reads treated/reads ctrl) for every guide. Combining those two values for every guide allows to robustly score the effect of each guide.

The analysis starts from a tab separated text file with the columns:

- Guidename
- Sample name (e.g. ctrl_1, treated_1)
- BC Sequence (10nt)
- Sequence read count

The analysis compares 2 experiments against each other using A) a conventional approach ignoring the clonal information provided by barcodes or B) using CRISPR-UMI analysis. For the conventional approach A) the sequence read counts for the same guide and the same sample but different BC Sequences are all added together and a file with 4 columns (Guidename, Genename, ctr (reads), exp (reads)) is generated. This file serves as an input file for MAGeCK [9]. For CRISPR-UMI analysis. B) the script calculates depletion by RPM(ctrl)/RPM(treated) and determines the median depletion of clones for each guide. It generates a file with the median depletion for all guides. It also generates a file for MAGeCK analysis but with the 4 columns (UMI-name, Guidename, ctr(reads), exp(reads)). Run MAGeCK on that file and the result will be a list of all guides ranked by MAGeCKs robust ranking algorithm. Combine the median depletion of every guide with the MAGeCK neg score (the score by which MAGeCK typically ranks genes). To rank genes we rank guides by median depletion and calculate rank/(total number of guides), we then rank all guides by MAGeCK neg score and calculate rank/(total number of guides). Multiplying those two values gives a score for every guide. We combine scores for every guide using fisher's method to generate a depletion score for each gene.

Resources:

Example Sequence files:

Example BAM files from the associated publication are available via NCBI Sequence Read Archive PRJNA383356 or at the link http://mendel.imp.ac.at/SEQUENCES/Michlits/.

BAM file structure:

Read1: A 50bp sequence read, the first 20nt are the guide sequence the following 30nt are constant sequence from the scaffold. A drop in read quality after base 20 is expected.

Index1: A 10bp sequence read, the barcode BC.

Index2: A 6bp sequence read, the experimental index that was introduced with the FW-PCR primer

Scripts

- Script 14 CRISPR_UMI.py
  Required input is a tab separated file with 4 columns (Guidename, Samplename, BC Sequence, Sequence read count) (See Step1).
- Script 15 combine_guides.py
  Combines output files from MAGeCK and output files from Script4 CRISPR_UMI.py. It assigns every guide a neg score (from MAGeCK) and the median depletion from CRISPR_UMI
- Script 16 quickforfisher.py
  Input file (tab separated file columns: guidename, median_depletion, guide score. The input file needs to be sorted alphabetically. The script rearranges lists of Guide scores for every guide so that GS can easily be combined in tabular format (e.g. using Microsoft Excel).

Dependencies:

- python
- scipy
- Mageck v.0.5.5

Files and results:

- File 9 CONV_4mageckpop0_200.txt:
  Example file. That file contains the columns (guidename, genename, ctrl reads, exp reads) and is one of the output files from Script4 CRISPR_UMI.py and is used for conventional analysis.
- File 10 CONV_pop0_200.gene_summary.xlsx:
  Example file. The MAGeCK result of Conventional analysis.
- File 11 Median_intermpop0_200.txt:
  Example file. That file contains the columns (guidename, median_depletion, number of clones, average reads per clone) and is one of the output files from Script4 CRISPR_UMI.py and is used for conventional analysis.
- File 12 CrUMI_pop0_200.guide_summary_adj.xls
  Example file: That file is a modified output from MAGeCK for clonal analysis. (the output was renamed – from the standardname …name.gene_summary... to name.guide_summary) and columns not required for further were analysis deleted.
- File 13 CrUMI_comb.xls
  Example file: Modified output file from Script 5 combine_guides.py. Calculation combined Guide Scores based on ranking of median and MAGeCK neg score are carried out in Excel.
- File 14 forfisher_hitlist.txt
  That's an example of an input file for Script 6 quickforfisher.py
- File 15 CrUMI_result.xls
  That's an example of the hitlist of genes after applying fishers method to combine guide scores.

Step-by-step:

- Start from a tab separated text file containing columns (guidename, samplename, 10nt BC sequence, read count). This file we termed “Masterdictionary” (see Step1).
- Run Script 14 CRISPR-UMI.py This script produces multiple output files for both conventional analysis and CRISPR-UMI analysis. The script also allows to run an optional algorithm called poptop. E.g. poptop(2) removes the 2 clones with most read-support for every guide. The script compars ctrl sample with exp sample but also allows 2 samples combined vs 2 others combined. The output files are as follows:
    - CONV_interm_name
    - CONV_unique_name
    - CONV_4mageck_name
    - CrUMI_interm_name
    - CrUMI_unique_name
    - CrUMI_4mageck_name
    - CrUMI_median_name

Variable inputs explained:

- Input filename: The input file is a Masterdictionary (see Step1) it is a tab separated text file with the columns (Guidename, Samplename, BC Sequence, Sequence read count).
- Output foldername: The script generates a folder (may work only for mac os) and writes all result files into that folder. It is a name of the analysis that is carried out.
- Enter number of clones popped: This is the optional poptop algorithm. The number you enter is the number of clones (topranked clones by readcount in ctr+exp) that will be removed from analysis.
- Type control sample name: That’s the name of the control sample to compare to. Sample names were defined earlier (e.g. example File 8: indEtop.csv (see Step1)).
- Type control sample#2: This is optional (it allows to compare 2 samples vs 2 samples and is sometimes useful in data analysis) if only 2 samples are compared to each other type: “NO”
- Type treated sample: (same as control)
- Type treated sample #2: (same as control)
- Enter max MM for removing barcode-shadows (1): This variable is used to determine which barcodes are true independent clones and which bc are likely sequencing errors (“bc-shadows” of higher represented UMIs) we have two criteria for barcode shadows 1) the edit distance (number of MM). We recommend to remove bc with edit distance 1. 2) min fold (see next point)
- Enter min fold for removing bc-shadows (3): Min fold is the minimum fold difference in total read counts that a bc-shadow (see above) must have from its “parent bc” to be classified as bc-shadow. We recommend a min fold value of 3.
- Enter minimum reads in ctrl+exp to accept clone(3). CRISPR-UMI evaluates every clones individually and individual clone reads often have very low read count. We suggest a cutoff to only accept clones which have at least 3 reads in total.
- Enter pseudocount (added to 0 reads to calculate fold_change)(0.5): This is only relevant for clones which have at least 3 reads in total (see cutoff above) so that affects clones with read count treated-control 0-3. The idea behind 0.5 is that if you assume you would have two times more cells in the experiment the 3 counts would double to 6 and 0 counts can give a maximum of 1 If doubled which would correspond to a ration of 1 to 6 or 0.5 (pseudocount) to 3.

python CRISPR_UMI.py

enter input filename (e.g. MD_30kMsEtop_i0MMsg1MM.txt): out_2_4li0MM_MD.txt

enter output foldername (e.g.200_pop0): 200_pop0

enter number of clones popped (2) if no poptop (0): 0

type control sample name (e.g.ctrl_200clones): c_200

type control sample #2 name (e.g.ctrl_200clones) (if no 2nd sample type:NO): NO

```
type treated sample #1 name (e.g.exp_200clones): e_200
type treated sample #2 name (e.g.exp_200clones) (if no 2nd sample type:NO): NO
enter max MM for removing bc-shadows (1): 1
enter min fold for removing bc-shadows (3): 3
enter minimum reads in ctrl+exp to accept clone (3): 3
enter pseudocount (added to 0 reads to calculate fold_change)(0.5): 0.5
reading in Masterdict
reading in MD finished! - start shadow-kicking and calculations
total_reads_ctrl =102247477
total_reads_exp =86282622
popped 0 clones.
total_reads_ctrl =102247477
total_reads_exp =86282622
finished reading in MD
total UMI analysed:          12997738
total UMI trashed (shadows):          73.31024059724854%      left with3469065clones
total reads:          188530099          left with 94.59839248267727% of reads
average clones per guide:  129.24499832346038      average reads per clone:    25.79863440301322
writing result files
Median_file finished.          processed: 26691 guides
CrUMI_file finished processed: 3481722 clones
CONV_file finished!          processed: 26428 guides
finished
```

For conventional analysis:

- Run MAGeCK(0.5.5): Combines reads of multiple guides to give a score for depletion and enrichment of genes using robust ranking algorithm (RRA) [9].

```
mageck test -k CONV_4mageckpop0_200.txt -t exp -c ctr -n CONV
```

- This is the end of conventional CRISPR-screen analysis, the name of the final hitlist is then CONV.gene_summary.txt

For CRISPR-UMI (clonal) analysis:

- The previous script CRISPR-UMI.py generated an input file (CrUMI_4mageck) as required for analysis with MAGeCK v 0.5.5. Conventionally MAGeCK combines reads of multiple guides to give a score for depletion and enrichment of genes using robust ranking algorithm algorithm (RRA) [9]. For CRISPR-UMI analysis we make use of the same principle but compute scores for guides based on multiple clones (BCs). Therefor the file contains the following 4 columns: clone identifier (UMI), guide, ctrl reads, exp reads, with the required MAKeCK column headers: sgRNA, Gene, ctrl, exp.
- Run MAGeCK:

mageck test -k CrUMI_4mageckpop0_200.txt -t exp -c ctr -n CrUMI

- The MAGeCK output file CrUMI.gene_summary.txt contains negative scores, with the column header: neg score, for depletion of guides and ranks guides according to this score. Comment:Note that MAGeCK calls this file …name.gene_summary.txt by default which in the case of CRISPR-UMI is misleading because it really is a scoring of guides not genes. In order to evaluate individual guides we combined the MAGeCK neg score (file CrUMI.gene_summary.txt) with the median depletion of individual guides computed by CRISPR-UMI.py (file CrUMI_Median_pop2_200.txt)
- Run Script 15 combine_guides.py to combine the data of median fold change with the MAGeCK computed neg score

python combine_guides.py

enter filename 1 to compare guidename in column[0] (e.g. CrUMI.gene.summary.txt_):

CrUMI_pop0_200.gene_summary.txt

enter filename 2 guidename in column[0] (e.g. CrUMI_median_pop2_200.txt)

CrUMI_median_pop0_200.txt

enter filename of combined file (e.g. CrUMI_guides): CrUMI_pop0_200_comb.txt

lines in file1: 26424

lines in file2: 26302

matching guides: 26301

finished

- The generated file (CrUMI_pop0_200_comb.txt) can be plotted as a 'vulcano-like' plot where –log10(neg score) is the y axis and median (RPM(treated)/RPM(ctrl)) the x-axis.
- To generate a hitlist we rank guides by (neg score) and give every guide a value of rank / total number of guides. Then we rank guides by median depletion and also give a value of rank / total number of guides. Those values are multiplied to give a CrUMI-score (see Excel file CrUMI_pop0_200_comb.xls). With Excel generate a file that contains column: guidename, median(treated/ctrl), CrUMI-score (guides must be sorted by name)
- Run Script 16 quickforfisher.py

python quickforfisher.py

type filename: 4_fishers_hitlist.txt

- Combine CrUMI-scores using Fisher's method, or (for genes where Fisher's method leads to numerical errors) combine CrUMI-scores in tabular format (e.g. using Excel) using the formula x= $\sqrt[n]{\prod_{i=1}^{n} p_i}$
- Genes are sorted according to combined guide scores to give a hitlist for CRISPR-UMI screen.

Step 3: Positive Selection; Incidence vs abundance analysis

For positive selection screens CRISPR-UMI can be used to differentiate between Abundance (that is total read number of a guide) and Incidence (number of independent barcodes sequenced) as we demonstrated for a screen for roadblocks of reprogramming.

Resources:

Example Sequence files:

Example BAM files from the associated publication are available via NCBI Sequence Read Archive PRJNA383356 or at the link http://mendel.imp.ac.at/SEQUENCES/Michlits/.

BAM file structure:

Read1: A 50bp sequence read, the first 20nt are the guide sequence the following 30nt are constant sequence from the scaffold. A drop in read quality after base 20 is expected.

Index1: A 10bp sequence read, the barcode BC.

Index2: A 6bp sequence read, the experimental index that was introduced with the FW-PCR primer

Scripts

- Script 17 IncidenceAbundance.py

  Required input is a tab separated file with 4 columns (Guidename, Samplename, BC Sequence, Sequence read count) (See Step1). The script counts all reads of a guide (adding up all barcodes) to return a value for abundance and counts the incidence (that is number of independent BCs) for every guide.

Dependencies:

- python
- scipy

Files and results:

- File 16 expindicesSE.csv: index file contains experimental barcodes and labelling of the samples

Step-by-step:

- Start from a tab separated text file containing columns (guidename, samplename, 10nt barcode sequence, read count). That file we termed “Masterdictionary” (see Step1).
- Run the python script 17 IncidenceAbundance.py: This script eliminates clones to differentiate between reads that are true independent clones and reads that may stem from introduced PCR single bp mutations or errors in sequencing. The output contains following files:
  - out_3_name_MDclean.txt: A tab separated file (guidename, samplename, 10nt barcode sequence, read count) without the clones that were eliminated.
  - out_3_name_MDkicked.txt: A tab separated file (guidename, samplename, 10nt BC sequence, read count) with only the clones that were eliminated.
  - out_4_name_conv.txt: contains data for conventional analysis.
  - out_5_name_IncAbu.txt: contains information of both incidence (number of selection events) and abundance (total read number) per guide for every

experiment. Can be used for plotting incidence vs abundance of positive selection events

- out_6_name_CrUMI.txt: contains read tables for individual clones

```
python 3_IncidenceAbundance.py
enter filename Masterdict (e.g. iPSC_MD): 'MD_iPSC_30k'
enter max MM for removing bc-shadows (1): 1
enter min fold for removing bc-shadows (10): 10
enter minimum reads in ctrl+exp to accept clone (10): 10
enter new sample name (e.g. iPSC_1MM10X10min): 'iPSC_1MM_10X_10min'
enter indexfilename (e.g. ind_file.csv): 'expindicesSE.csv'
reading in Masterdict
reading in done
finding bc-shadows done
shadows eliminated: 10918746    lines processed:11475111
Index dictionaries done
wrote 182406 guides in classical output file
writing read tables
write output_5 per clone analysis
finished
```

- Open file out_5_name_IncAbu in Excel and calculate enrichment based on Abundance (conventional way) or based on incidence (only possible with CRISPR-UMI). For calculation of enrichment based on abundance divide RPM(exp)/RPM(ctr) and then determine the median of 4 experiments. For enrichment based on incidence divide incidence(exp) by RPM(ctrl). The results can be plotted as enrichment based on abundance vs enrichment based on incidence.

Step 4: Clonal size estimation in reprogramming screen

This Section describes scripts used for estimating colony size in a positive selection screen of reprogramming. Mouse embryonic fibroblasts were infected with CRISPR-UMI library and reprogrammed to induced pluripotent stem cells. In the example data set samples are labelled C1-C4 for controls (biological replica, MEFs from 4 mice) and E1-E4 for experiment (reprogramming of MEFs from those 4 mice), E1A and E1B, are technical replicas. In this section of the Step-by-step protocol the term UMI is used to solely describe the 10nt barcode and not the combination of a guide-barcode, this is not coherent with the rest of the Step-to-Step protocol or the aossicated publication. The colony size is reflected in reads per UMI and colony number in the number of different UMI per guide. The Analysis described here is an estimation of average colony size depending on the gene knocked out with CRISPR-UMI.

Scripts:

- Script 18 bam2fastq_map_UMIgroup.sh
    - convert unaligned bam to fastq with UMI sequence appended to read id
    - map reads without mismatches to guide sequences using bowtie; sort and index aligned bam
    - group similar UMIs mapping to the same guide using umi_tools
    - produce count table of reads per guide and UMI using UMI group information
    - example usage: bam2fastq_map_UMIgroup.sh -v 10 -I E1_A.bam -O E1_A -D results -R guides.fa
- Script 19 BUILD.ref.sh
    - build index for reference set of guide sequences
- Script 20 MAP.bam2fastq_map_UMIgroup.sh
    - produce count tables for all bam files defined in parameters.txt by starting bam2fastq_map_UMIgroup.sh
- Script 21 counts2libsize.R
    - calculate scaling factor using count-files and based on parameters.txt
- Script 22 plot_colony_size_estimation.R
    - import guide_umi count-files for experiments defined in parameters.txt
    - filter: keep counts of 5 and above; for each guide keep count values up to the cumulative percentage of 90%
    - normalize counts for total library size; median scale
    - plot boxplot of normalized scaled counts across all experiments for selected guides

Dependencies

- the shell-scripts above use samtools/1.3.1, bowtie/1.2 and UMI-tools: 0.4.4
- the R-scripts use dplyr and ggplot2

Input Files Used in Example:

- Example BAM files from the associated publication available via Sequence Read Archive PRJNA383356 or at the link http://mendel.imp.ac.at/SEQUENCES/Michlits/.
- File 17 targets.txt: A space separated text-file defining path to input bam files, with corresponding experiment and replicate-name without a header line
- File 18 guides.fa: fasta file with guide sequences to align to
- File 19 guideids2plot.txt: text file containing guide ids to be plotted

Example run:

- Set variables for input data used by shell and R-scripts

```
export WANTEDGUIDES_FILE=data/guideids2plot.txt
export TARGETS_FILE=data/targets.txt
export REF=data/guides.fa
export RESULTS_DIR=results
```

- Run shell scripts for preprocessing / mapping / count-summarization
- Example uses qsub to process samples in parallel

```
./Script 19 BUILD.ref.sh
./Script 20 MAP.bam2fastq_map_UMIgroup.sh
```

- Run R-scripts to calculate library size and plot colony size estimates

```
R --vanilla --slave < Script 21 counts2libsize.R
R --vanilla --slave < Script 22 plot_colony_size_estimation.R
```

Step-by-step:

- The entry point to the analysis are unaligned bam files, with one bam file per sample containing the sequenced reads together with a barcode. Then, for each sample/bam-file the observed colony sizes are estimated based on the number of barcodes mapped to each guide, also referred to guide_umi_count below. Counting is performed independently for each sample. Count data are normalized and scaled for all replicates of a condition (experiment and control in this case) before plotting values for all replicates in a boxplot to illustrate the distribution of observed colony sizes for each guide.
- An unaligned bam file is first converted to fastq with the UMI incorporated in the readname as the last string after ":". This is done as UMI-tools is used at a later step to model sequencing errors in the UMIs, and the software assumes that the UMI is incorporated in the read name.

```
samtools view sample.bam |  awk '{if(!match(substr($14,6,10),/N/)){print
"@"$1":"substr($14,6,10)"\n"substr($10,1,20)"\n+\n"substr($11,1,20)}}' > sample.fastq
```

- The resulting fastq file should looks like the example below in which the umi (only the barcode here) is GACATAGTTT and the read-sequence is the sequenced guide.

```
# @7001253F:400:H5F3TBCXX:1:1108:1417:2064#E1A_exp_9:GACATAGTTT
# CTCACAGGCCAAGAACTACA
# +
# DDDDDGHHHGIIIIIIIIII
```

- The obtained fastq file is mapped to the reference file of guide sequences, disallowing mismatches, keeping only aligned reads, followed by sorting and indexing of the aligned bam file. The reference file of guide sequences is a fasta file of 20nt long guide sequences and needs to be indexed prior to this alignment step as illustrated in 1_BUILD.ref.sh.

  bowtie -p 10 -m 1 --best --strata -S -v 0 guides.fa sample.fastq | samtools view -S -F 0x0004 - |

  samtools view -ht sample.fai - | samtools sort --threads 10 --output-fmt BAM -o sample.bowtie.bam -

  samtools index sample.bowtie.bam

- umi_tools group with default settings is used to produce a flatfile listing all mapped reads with corresponding read group information- that is reads mapping to the same guide and having similar UMI keys are assigned to one read group identified by a read group id (unique_id).

  umi_tools group --umi-separator=: --in-sam -I sample.bowtie.bam --group-out=sample_UMI_group.txt

- A sample output looks like this where a read-group can be identified by the unique_id column, or in this particular case by combining the contig+final_umi columns as all reads align at the same position of the guide reference sequence- its beginning:

  # read_id contig position umi umi_count final_umi final_umi_count unique_id

  # 7001253F:400:H5F3TBCXX:1:2209:13158:2198#E1B_exp_25:CCCTGGGGGG

  Alkbh2_1_dnaRepair 0 CCCTGGGGGG 177 CCCTGGGGGG 182 313

  # 7001253F:400:H5F3TBCXX:1:2209:4518:14317#E1B_exp_25:CCCTGGGGGG

  Alkbh2_1_dnaRepair 0 CCCTGGGGGG 177 CCCTGGGGGG 182 313

  # 7001253F:400:H5F3TBCXX:1:2209:6348:34026#E1B_exp_25:CCCTGGGGGG

  Alkbh2_1_dnaRepair 0 CCCTGGGGGG 177 CCCTGGGGGG 182 313

  # 7001253F:400:H5F3TBCXX:1:2209:8409:54303#E1B_exp_25:CCCTGGGGGG

  Alkbh2_1_dnaRepair 0 CCCTGGGGGG 177 CCCTGGGGGG 182 313

  # 7001253F:400:H5F3TBCXX:1:1203:12217:82031#E1B_exp_25:CACTGGGGGG

  Alkbh2_1_dnaRepair 0 CACTGGGGGG 2 CCCTGGGGGG 182 313

- The individual columns listed in the group flatfile are:

  # read_id

  # contig

  # position

  # umi = raw umi

  # umi_count = how many times was this umi observed at the same alignment coordinates

  # final_umi = the error corrected umi

  # final_umi_count = how many times was the umi observed at the same alignment coordinates, inc. error correction

  # unique_id = the unique identifier for this group

- Particularly in case of extreme colony size we have observed relatively high umi_counts also for UMIs similar but distinct from the most frequent final_umi in a read group. We want to keep those reads with an assumed UMI sequencing error, but only in cases with high confidence indicated by the umi_count. The group flatfile is therefor parsed to keep only reads where:
    - reads show a UMI identical to the most frequent primary final_umi in a read group: the umi column equals to the error corrected final_umi
    - reads with UMIs that are secondary in the read group, but could be derived by a sequencing error based on the overall read counts of the read group: where the minor similar UMI in the read group is confident (found in more than 10 reads), and the read group shows high counts and higher counts than the secondary UMI supporting the notion of a possible sequencing error

```
awk '{if ( ($4==$6) || ($5>10 && $7>100 && $7>($5+10)) ){ print } }' sample_UMI_group.txt | awk '{print $2,$6,$5,$4}' | sort | uniq | grep -v final_umi > sample_UMI_group_highconf.txt
```

- The above step results in a count table were for each retained guide (col 1) with its observed umis (col 4) one can derive the number of reads with a specific guide_umi combination in column 3 (guide_umi_count not accounting for UMI errors), and the error corrected final_umi sequence in column 2.

```
# contig final_umi umi_count umi
#
# Alkbh2_1_dnaRepair CCCTGGGGGG 177 CCCTGGGGGG
# Alkbh2_1_dnaRepair CGAGGGGCTT 1 CGAGGGGCTT
# Alkbh2_1_dnaRepair CGTTCCAGCG 1 CGTTCCAGCG
# Alkbh2_1_dnaRepair CTATTATGTA 1 CTATTATGTA
```

- The final colony size count table which is accounting for UMI errors is obtained by summing up the umi_counts of remaining guide_umi combinations in the read group. A read group is hereby identified by combining the contig+final_umi columns, and the counts are obtained as the summed up umi_counts.

```
awk '{arr[$1" "$2]+=$3} END {for (i in arr) {print i" "arr[i]}}'  sample_UMI_group_highconf.txt > sample_guide_UMI_counts.txt
```

- The resulting file lists the number of reads (col 3) supporting a observed combination of guide and error free UMI (col 1 and 2). This count reflects the colony size of the colony containing the particular guide and identified by the specified error-free UMI.

```
# Alkbh2_1_dnaRepair CCCTGGGGGG 177
# Alkbh2_1_dnaRepair CGAGGGGCTT 1
# Alkbh2_1_dnaRepair CGTTCCAGCG 1
# Alkbh2_1_dnaRepair CTATTATGTA 1
```

- The above procedure produces a single count file per sample. In order to estimate the distribution of colony sizes across experiments the count tables are merged, count data are normalized for library size and the data are plotted using R.
- The R script 3_counts2libsize.R imports the obtained colony size tables for the 8 separate samples, sums up all reads per sample to obtain the library size and determines a scale factor to normalize all samples relative to this maximal observed library size.
- The R script 4_plot_colony_size_estimation.R is used to import the the obtained colony size tables for the 8 separate samples. For each experiment only colonies (=guide+UMI combinations) supported by more than 5 reads are kept. Guide_umi read counts are also filter based on cumulative frequency of colony sizes observed, normalize based on the determined scaling factor, and a median-scaled value is determined for each colony within an experiment. Than colony-size/guide_umi counts for experimental replicates are merged and the distribution is evaluated in a boxplot.

Step 5: Comparisons of CRISPR-UMI vs conventional CRISPR screening

General notes:

We use two approaches to evaluate and quantify screen-quality. Both quality checks are carried out on guide level. One is signal to noise ratio, where we plot all guides in a volcano plot and define signal as distance from the origin and noise as the standard deviation among non-targeting ctrl guides. The other method ranks all guides and calculates for how many guides per gene are found among the top 5,10,20,30, and so on guides.

Resources:

Scripts

- Script 23 Guide_frequency.py

Files, intermediate files and results

- File 20 SNR estimation.xlsx

Step-by-step:

Signal to noise ratio

- Estimation of signal to noise ratio was carried out in Excel. See File 20 SNR estimation.xlsx as an example.
- The hitlist was reduced to non-targeting Ctrl guides and guides against genes of the non-homologous end joining (NHEJ) pathway (Lig4, Nhej1, Xrcc4, Xrcc5, Xrcc6).
- Signal was defined as distance from the origin of the volcano plot using the following equations. Both depletion and significance axis are normalized to the strongest guide in the set.

$$d_i = {RPM_{treated}}/{RPM_{Control}}$$

d_i……..depletion of guide I (Conventional)
RPM.....reads per million

$$S_i = \sqrt{\left(\frac{1-d_i}{1-d_{min}}\right)^2 + \left(\frac{\log_{10}(p_i)}{\log_{10}(p_{min})}\right)^2}$$

S_i……...signal of guide i.
d_i…….depletion of guide i (Conventional),
median depletion of clones for guide i (CRISPR-UMI)
d_{min}…….depletion of strongest depleting guide in the comparison (Conventional),
lowest median depletion of all guides in the comparison (CRISPR-UMI)
p_i……p-value for depletion for guide i (Conventional),
mageck neg score for guide i (CRISPR-UMI)
p_{min}…….lowest p-value of all guides in the comparison (Conventional),
lowest mageck neg score for all guides in the comparison (CRISPR-UMI)

- Signal to noise ratio is calculated using the formula

$$SNR = \frac{\overline{S}_{NHEJ}}{\sigma_{CTRL}}$$

SNR……..signal to noise ratio.

$\overline{S}_{NHEJ}$…….average signal of all guides of NHEJ pathway

σ_{CTRL}…….Standard deviation of signal from all control guides

Efficiency comparison

- Rank guides according to their Guide Scores using the formula. (see Step2 Negative Seletction, CRIPSR-UMI pipeline, and data files)

$$\mathrm{GS} = \frac{rank_{depletion}}{N_{total\ guides}} \times \frac{rank_{p-value}}{N_{total\ guides}}$$

GS……..…..Guide score - combined score, evaluation of guides

$rank_{depletion}$……rank of guides by depletion (Conventional Analysis)
rank of guides by median depletion (CRISPR-UMI Analysis)

$rank_{p\text{-}value}$……..rank of guides by p-value (Conventional Analysis)
rank of guides by mageck neg score (CRISPR-UMI Analysis)

$N_{total\ guides}$……..…total guides in experiment

- Run Script 23 guide_frequency.py

python guide_frequency.py

type filename (.csv)ranklist_dil_conv.csv

- Plot data in Excel: rank is x, y-axis the average number of guides per gene among the top x guides.

Costs and benefits of CRISPR-UMI

The following table gives a short summary comparing the costs and benefits of CRISPR-UMI versus standard pooled CRISPR screen methodology.

	Conventional	*CRISPR-UMI*	*cost/benefit*
Library cloning	1-step library cloning	2-step library cloning	For CRISPR-UMI the library is cloned into a different backbone. Costs are about one week more cloning time to generate a UMI-library backbone.
Screening	Standard procedure	Standard procedure or Standard procedure with limiting dilution	A screen with limiting dilution takes about 3-5 days longer depending on the number of clones and cells per clone used. But CRISPR-UMI also worked without limiting dilution. No costs
Sequencing	SR50, single indexing	SR50, dual indexing	Same Illumina kit, No costs
Data Analysis	Mageck or other ready to use pipelines	CRISPR-UMI pipeline. A series of scripts including Mageck	The CRISPR-UMI pipeline at the time of publication is a very 'raw' custom pipeline written by a 'wet-lab' PhD student. It should not be a main challenge for a trained computational biologist to analyze a CRISPR-UMI screen. Analysis runtime of CRISPR-UMI takes about 1-2h longer than conventional analysis. Costs: 30min – 4days depending on skill level.
Result	The screen identified strongest hits (top5), but weaker hits (6^{th}-17^{th}) contained about 50% putative false positives. The screen missed 2 putative true hits discovered only with CRISPR-UMI	CRISPR-UMI is robust towards clonal outliers. The strongest (top13) hits do not contain a single putative false positive hit. CRISPR-UMI identified 2 genes not identified by conventional CRISPR screening	The main lesson learned from comparing conventional screening with CRISPR-UMI is that single clones with very high read number strongly affect scoring at guide level in a conventional screen leading to both false positives and false negatives. CRISPR-UMI on the other hand is very robust towards this outliers or artefacts. This lead us to suggest that while CRISPR-UMI is showing improvements in 'easy' screening settings as we demonstrated, the true power of the method may be its robustness towards clonal outgrowth in difficult screening scenarios such as organoid or in-vivo screens. Never the less the improvements in 'easy' screening scenarios come at essentially no costs. Furthermore, the use of a CRISPR-UMI library does not interfere with conventional analysis, but rather just offers a second layer of analysis in addition.

Table1. Costs and Benefits of CRISPR-UMI vs Conventional CRISPR-screening

References:

1. Chen, B. *et al.* Dynamic imaging of genomic loci in living human cells by an optimized CRISPR/Cas system. *Cell* **155,** 1479–1491 (2013).
2. Doench, J. G. *et al.* Rational design of highly active sgRNAs for CRISPR-Cas9-mediated gene inactivation. *Nature Biotechnology* **32,** 1262–1267 (2014).
3. Rosenbloom, K. R. *et al.* The UCSC Genome Browser database: 2015 update. *Nucleic Acids Res.* **43,** D670–81 (2015).
4. Finn, R. D. *et al.* The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Res.* **44,** D279–85 (2016).
5. Kuscu, C., Arslan, S., Singh, R., Thorpe, J. & Adli, M. Genome-wide analysis reveals characteristics of off-target sites bound by the Cas9 endonuclease. *Nature Biotechnology* **32,** 677–683 (2014).
6. SQ, T. *et al.* GUIDE-seq enables genome-wide profiling of off-target cleavage by CRISPR-Cas nucleases. *Nature Biotechnology* **33,** 187–197 (2014).
7. Wang, T., Wei, J. J., Sabatini, D. M. & Lander, E. S. Genetic Screens in Human Cells Using the CRISPR-Cas9 System. *Science* **343,** 80–84 (2014).
8. Elling, U. *et al.* Forward and reverse genetics through derivation of haploid mouse embryonic stem cells. *Cell Stem Cell* **9,** 563–574 (2011).
9. Li, W. *et al.* MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. *Genome Biol.* **15,** 554 (2014).