nature protocols

PROTOCOL

https://doi.org/10.1038/s41596-018-0113-7

# Integrative analysis of pooled CRISPR genetic screens using MAGeCKFlute

Binbin Wang[1,12], Mei Wang[2,12], Wubing Zhang[1,12], Tengfei Xiao[3,10], Chen-Hao Chen[3], Alexander Wu[3,4], Feizhen Wu[5,6], Nicole Traugh[3], Xiaoqing Wang[3], Ziyi Li[1], Shenglin Mei[1], Yingbo Cui[7], Sailing Shi[1], Jesse Jonathan Lipp[8], Matthias Hinterndorfer[8], Johannes Zuber[8], Myles Brown[3,9], Wei Li[3,11]* and X. Shirley Liu[1,3]*

**Genome-wide screening using CRISPR coupled with nuclease Cas9 (CRISPR–Cas9) is a powerful technology for the systematic evaluation of gene function. Statistically principled analysis is needed for the accurate identification of gene hits and associated pathways. Here, we describe how to perform computational analysis of CRISPR screens using the MAGeCKFlute pipeline. MAGeCKFlute combines the MAGeCK and MAGeCK-VISPR algorithms and incorporates additional downstream analysis functionalities. MAGeCKFlute is distinguished from other currently available tools by its comprehensive pipeline, which contains a series of functions for analyzing CRISPR screen data. This protocol explains how to use MAGeCKFlute to perform quality control (QC), normalization, batch effect removal, copy-number bias correction, gene hit identification and downstream functional enrichment analysis for CRISPR screens. We also describe gene identification and data analysis in CRISPR screens involving drug treatment. Completing the entire MAGeCKFlute pipeline requires ~3 h on a desktop computer running Linux or Mac OS with R support.**

## Introduction

CRISPR (clustered regularly interspaced short palindromic repeats)–Cas9 is a powerful technology for targeting desired genomic sites for gene editing or activity modulation via specific single-guide RNAs (sgRNAs)[1]. CRISPR screening is a high-throughput technology capable of investigating the functions of many genes in a single experiment. In a screening experiment, sgRNAs are designed, synthesized and cloned into a lentivirus library, which is subsequently transduced into cells at a low multiplicity of infection to ensure that only one sgRNA copy is integrated per cell. A sgRNA usually contains 18–20 nt complementary to its target and guides the Cas9 enzyme to a specific DNA location where Cas9 induces a double-strand break. The repair of such a break by the cell often leads to a knockout of the targeted gene. Cells are cultured under different experimental settings, and the sgRNAs incorporated into the host genome are replicated with host-cell division.

Genome-wide CRISPR screens[2,3] allow systematic investigation of gene functions in various contexts[4]. The screening procedure can be categorized into knockout screens[5–7] and CRISPR activation or inhibition screens (CRISPRa/CRISPRi), which are performed by fusing a catalytically inactive Cas9 (dCas9) to transcriptional activation or repression domains, respectively. Data analysis for each type of CRISPR screen is similar in principle. For simplicity, within this protocol, we will refer to CRISPR knockout and CRISPR activation/inhibition screens as 'CRISPR screens', and use CRISPR knockout screens as an example to demonstrate data analysis. CRISPR screens have been highly effective in identifying genes that function in tumorigenesis[8,9], metastasis[10], and response to immunotherapy[11,12] as well as genes associated with drug response[13–15].

[1]Shanghai Key Laboratory of Tuberculosis, Clinical Translational Research Center, Shanghai Pulmonary Hospital, School of Life Sciences and Technology, Tongji University, Shanghai, China. [2]Department of Geriatrics, Shanghai General Hospital, Shanghai, China. [3]Center for Functional Cancer Epigenetics, Department of Data Sciences, Dana-Farber Cancer Institute, Harvard T. H. Chan School of Public Health, Boston, MA, USA. [4]Program in Computational Biology and Quantitative Genetics, Harvard School of Public Health, Boston, MA, USA. [5]Laboratory of Epigenetics, Institutes of Biomedical Sciences, Fudan University, Shanghai, China. [6]Key Laboratory of Birth Defects, Children's Hospital of Fudan University, Shanghai, China. [7]School of Computer, National University of Defense Technology, Changsha, China. [8]Research Institute of Molecular Pathology (IMP), Vienna Biocenter (VBC), Vienna, Austria. [9]Department of Medical Oncology, Dana-Farber Cancer Institute and Harvard Medical School, Boston, MA, USA. [10]Present address: GV20 Oncotherapy Shanghai, China. [11]Present address: Center for Genetic Medicine Research, Children's National Medical Center, Department of Genomics and Precision Medicine, George Washington University, Washington, DC, USA. [12]These authors contributed equally: Binbin Wang, Mei Wang, Wubing Zhang. *e-mail: wli2@childrensnational.org; xsliu@jimmy.harvard.edu

To identify essential genes in a cell population, cells with CRISPR perturbation can be collected in two conditions, one representing the initial sgRNA status (day 0), and the other representing a cell population allowed to proliferate under certain experimental conditions for a set amount of time. To study gene–drug interactions, CRISPR screens can be conducted using three different cell populations: the day 0 population, a drug-treated population (treatment) and a control population (mock-drug control, typically treated with a vehicle such as DMSO). At the end of the screen, genomic DNA from the transduced cells is extracted, and the sgRNA-encoded regions where the virus had integrated into the host genome are sequenced using high-throughput sequencing. The read count of each sgRNA is a proxy for the proliferation characteristics of the cell with that specific knockout.

For many research groups, data analysis is the most challenging aspect of CRISPR screens. The primary goal of data analysis is to identify genes whose disruption leads to phenotype change (e.g., cell growth) under certain screening conditions, relative to a predefined control condition (e.g., before screening starts or cells without drug treatment). A secondary goal is to infer biological insights from those hits using functional analysis approaches, including gene ontology (GO), pathway enrichment analysis or gene set enrichment analysis (GSEA)[16,17]. We previously developed two algorithms[18,19] to analyze CRISPR screen data: MAGeCK (model-based analysis of genome-wide CRISPR–Cas9 knockout)[18] and MAGeCK-VISPR (visualization for CRISPR)[19]. Both algorithms use a negative binomial distribution to model variances of sgRNA read counts. MAGeCK RRA and MAGeCK MLE are the two main functions of MAGeCK that can be used for identifying CRISPR screen hits. MAGeCK RRA uses robust rank aggregation (RRA) and MAGeCK MLE utilizes a maximum-likelihood estimation (MLE) for robust identification of CRISPR screen hits (see further discussion in the 'Experimental design' section). MAGeCK-VISPR is a comprehensive QC, analysis and visualization workflow for CRISPR–Cas9 screens. It incorporates MAGeCK and VISPR, which together interactively explore results and QC in a web-based front end. In combination, MAGeCK and MAGeCK-VISPR allow users to perform read-count mapping, normalization and QC, as well as to identify positively and negatively selected genes in the screens.

## Overview of the protocol

Here, we describe how to use MAGeCKFlute (Fig. 1), a comprehensive CRISPR screen analysis pipeline that applies either MAGeCK or MAGeCK-VISPR to identify gene hits and then performs downstream functional analyses using FluteRRA or FluteMLE. MAGeCKFlute has functions that perform batch effect removal, normalization and copy-number correction. We chose the name MAGeCKFlute to invoke a pipeline, and as a metaphorical reference to the successful completion of a series of tests in Mozart's popular opera of the same name. We give users the option of performing a step-by-step analysis of screen data with MAGeCK (Step 7A(i–ix)), or a comprehensive workflow with MAGeCK-VISPR (Step 7B(i–vii)), which also includes visualization of the results. Within MAGeCK, we demonstrate how to identify gene hits with MAGeCK RRA (Step 7A(v)) using publicly available data from a CRISPR screen performed in glioblastoma (GBM) stem-like cells (GSCs)[20]. We also show how to use MAGeCK MLE (Step 7A(vi and vii)) to identify gene hits from a screen with multiple conditions, using an dataset for a melanoma cancer cell line, A375, treated with the BRAF protein kinase inhibitor vemurafenib (PLX)[7]. We also demonstrate how to remove batch effects using a CRISPR screen dataset that was generated in different batches[20] (Step 7A(iv)). In the final steps of the analysis, we show how to use FluteMLE to perform QC at the beta score level for MAGeCK MLE results and how to use FluteRRA or FluteMLE to perform downstream GO term and Kyoto Encyclopedia of Genes and Genomes (KEGG)[21] pathway enrichment analyses for MAGeCK RRA and MLE results (Step 11). With our example, we describe how to identify genes involved in drug pathways by comparing CRISPR screen results from different drug treatments.

## Comparison with other methods

Other algorithms such as RIGER[22], RSA[23], BAGEL[24], ScreenBEAM[25] and casTLE[26] also perform parts of the CRISPR screen analysis pipeline. RIGER[22] and RSA[23] examine the rank distribution of all sgRNAs targeting a gene with regard to selection during the screen and calculate the statistical significance of all sgRNAs targeting a gene. BAGEL is based on a Bayesian supervised learning method; ScreenBEAM uses a Bayesian hierarchical model to assess gene-level activity from all relevant measurements; and casTLE combines measurements from multiple targeting reagents to estimate a maximum effect size. MAGeCKFlute differs from these algorithms because it uses a negative binomial model to address off-target sgRNAs and provides a comprehensive CRISPR screen

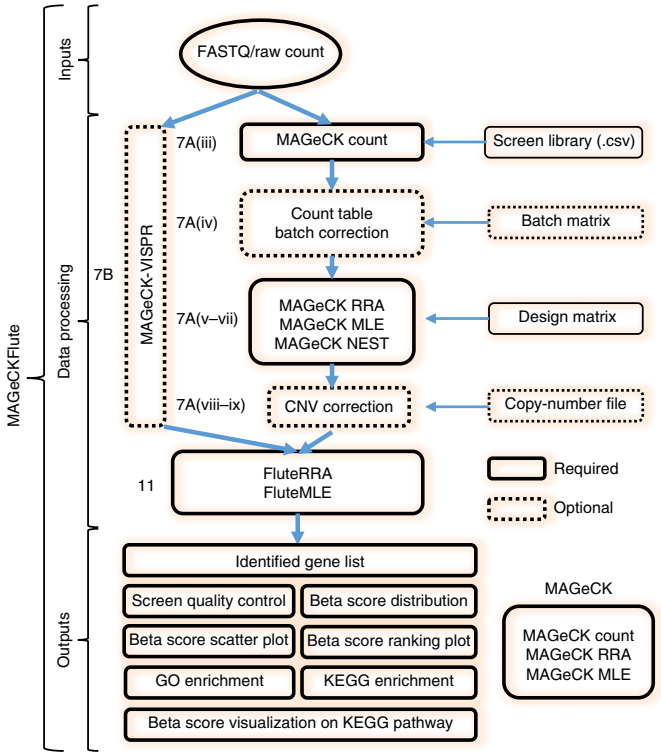


**Fig. 1 | Schematic representation of CRISPR–Cas9 screen analysis using MAGeCKFlute.** Procedure step numbers are shown to the left of the corresponding boxes. The FASTQ files or raw read-count files (Table 4), a screen library file (Table 2) and a design matrix (Table 5) are required as input for initial analysis by both MAGeCK and MAGeCK-VISPR. The following input components are optional: count table batch correction (which requires an otherwise optional batch matrix file) and CNV analysis and correction. Users have the option of analyzing CRISPR screen data step by step with the individual MAGeCK modules (Step 7A, right branch) or with MAGeCK-VISPR, which combines all MAGeCK modules and additional quality control and visualization functions in a single script (Step 7B, left branch). FluteRRA and FluteMLE use the results generated by MAGeCK or MAGeCK-VISPR for downstream analyses, including pathway enrichment using GO and KEGG. Outputs of FluteRRA or FluteMLE include the beta score distribution and beta score scatter plots.

workflow. The workflow is based on MAGeCK and MAGeCK-VISPR, and includes read mapping, normalization, QC, hit identification and functional analysis. This protocol aims to enable wet-lab and computational investigators to analyze their CRISPR screen data, and to aid in understanding the biology behind the screen results.

## Applications

MAGeCKFlute can be applied to remove batch effects, correct copy-number bias, identify screening hits and perform downstream functional analysis for various CRISPR screens, such as CRISPR knockout, CRISPR activation and CRISPR inhibition screens. MAGeCKFlute can map raw reads onto a CRISPR library, normalize read counts to allow comparison between different samples, identify genes that are positively or negatively selected under the screening conditions, and explore enriched GO terms and KEGG pathways for those selected genes. For CRISPR screening samples that have been treated with a drug, MAGeCKFlute can also be used to identify drug-associated genes. MAGeCKFlute generates figures and tables representing the results of CRISPR screen data analysis and can be paused at any step.

## Limitations

Although the current protocol provides an integrated solution for analyzing pooled CRISPR screens, there are still limitations. For example, there are many different approaches, such as GOstats[27], clusterProfiler[28] and GESA[17], that can be used to perform functional enrichment analysis, and currently it is unclear which model is most appropriate for analyzing screening results. MAGeCK-Flute provides all three options mentioned above for enrichment analysis, and users are encouraged

**Table 1 | Main functions of MAGeCKFlute**

| Functions | Description of functions |
|---|---|
| mageck count | Map the raw FASTQ data to reference library file and count the reads for each sgRNA |
| mageck test | MAGeCK RRA (identifying CRISPR screen hits by calculating the RRA enrichment score to indicate the essentiality of a gene) |
| mageck mle | MAGeCK MLE (identifies CRISPR screen hits by calculating a 'beta score' for each targeted gene to measure the degree of selection after the target is perturbed) |
| VISPR | Visualization of the results from MAGeCK |
| mageck-vispr | Quality control at the FASTQ and raw-count level; includes all the functions of MAGeCK and VISPR |
| BatchRemove | Removes batch effects of CRISPR screen data at the raw read-count level |
| mageck test/mle, `--cnv-norm` parameter | Corrects the bias caused by copy number when identifying hits with MAGeCK RRA and MAGeCK MLE |
| mageck_nest.py | Improves hit identification and removes outlier sgRNAs |
| FluteRRA | Downstream analysis of the MAGeCK RRA results |
| FluteMLE | Quality control at the beta score level; normalization with essential genes; identification of drug treatment-related hits; functional analysis |

to test different models. Another limitation involves the quality of the screens. Certain samples may lose >50% of the cell population, perhaps due to a long culture time or high-dose drug treatment that leads to strong selection. Such samples should be avoided, as they will influence the accurate identification of negatively selected hits. Alternatively, users can reduce the screening period to obtain higher-quality screening data. Despite these limitations, our protocol provides a convenient approach for performing a comprehensive computational analysis for CRISPR screens.

## Experimental design

### The basics of CRISPR screen data analysis with MAGeCK and MAGeCK-VISPR

MAGeCKFlute performs reads mapping and hit identification using `mageck count` and `mageck test/mle`, respectively, which are the main functions of MAGeCK and MAGeCK-VISPR (Table 1). The typical input of MAGeCKFlute is a FASTQ file or a raw read-count table in which columns are samples and rows are sgRNAs. CRISPR screen analysis usually contains two parts: sgRNA-level and gene-level analysis. The sgRNA-level analysis models read counts of individual sgRNAs independently. The fold change and *P* value are calculated for each sgRNA, which is similar to RNA-seq analysis. The gene-level analysis integrates the sgRNA-level fold change and *P* values to identify interesting gene hits. MAGeCK first maps sequencing reads to the sgRNA design library[29] and then normalizes sgRNA read counts to adjust for sequencing depth.

### Quality control and read-count table generation

Alignment of reads to a known sgRNA library and evaluation of screen quality (Fig. 2) are required before the identification of hits. MAGeCK and MAGeCK-VISPR align reads to a sgRNA library file, count the read number for each sgRNA and output a set of QC statistics, including the following.

- The numbers of mapped reads (Fig. 2a);
- The percentage of reads mapped (Fig. 2a);
- The read count correlation between samples (Fig. 2b);
- The Gini index[30] (measures the evenness of sgRNA read counts) (Fig. 2c);
- The number of sgRNAs to which zero reads are mapped (Fig. 2d).

A low percentage of mapped reads may indicate errors in oligonucleotide synthesis, sequencing errors or contaminated samples. A high mapping rate suggests success in sample preparation and sequencing. A low number of missing sgRNAs is also a good indicator of high-quality samples. MAGeCK and MAGeCK-VISPR use the Gini index, a common measure of income inequality in economics[30] to measure the evenness of sgRNA read counts. A high Gini index suggests that the sgRNA read count is distributed heterogeneously across the target genes. This is potentially caused by unevenness in CRISPR oligonucleotide synthesis, low-quality viral library packaging, poor efficiency in viral transfection or over-selection during the screens.

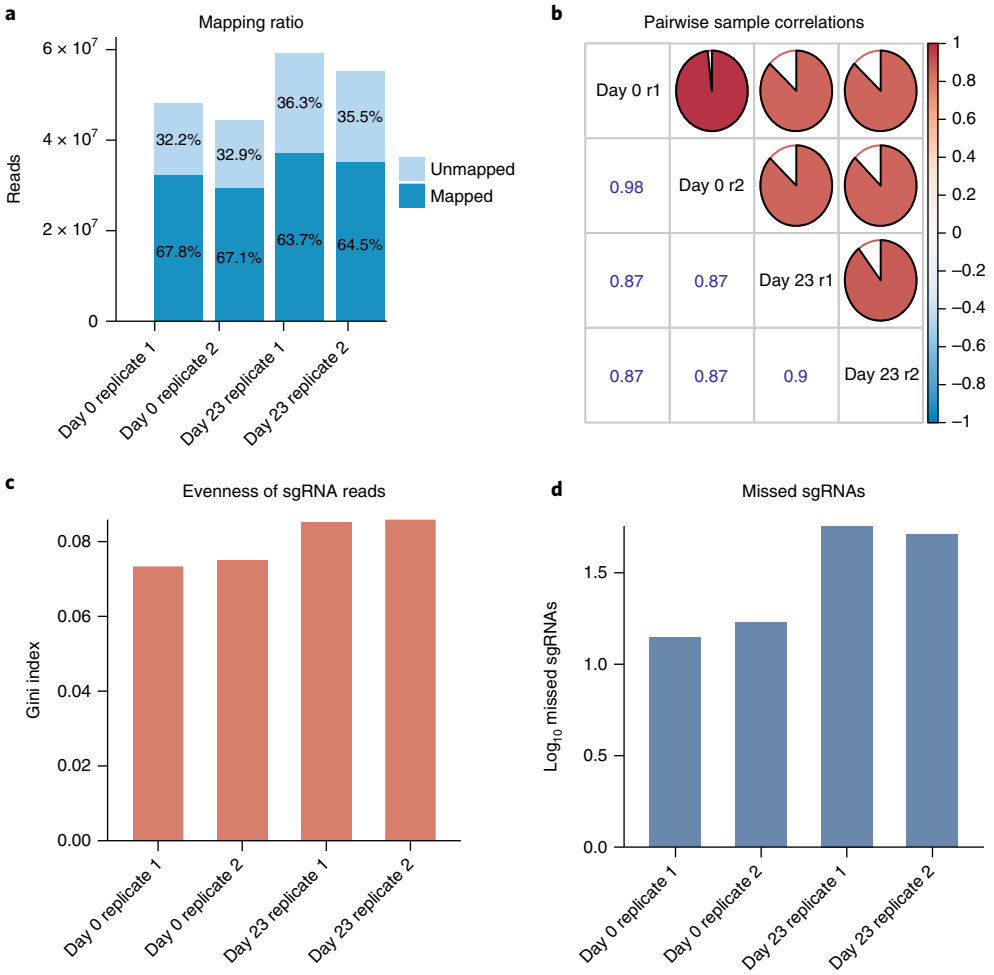


**Fig. 2 | Example quality control assessment of CRISPR–Cas9 screen data.** All four samples analyzed here are from a genome-wide CRISPR screen dataset generated from patient-derived GBM stem-like cells (GSCs)[20]. These samples represent two conditions: day 0, the initial time point of the screen, and day 23, after 23 d of culture. Replicates 1 and 2 are biological replicates. These results are generated by performing MAGeCK count. **a**, Read counts and mapping percentages. The mapped read percentage should be >65%. **b**, Pairwise sample correlations of read count. **c**, The Gini index, which measures read depth evenness within samples. **d**, Number of missed sgRNAs.

### Batch effect removal

CRISPR screens that are performed or sequenced in multiple batches may harbor batch effects. If CRISPR screen data were generated with different reagents or sequencing platforms, at different times or with any other unintended variation in experimental conditions, batch effects could be observed in these data. In such cases, batch effect removal becomes a necessary step for data analysis. One example is a public CRISPR screen in colon cancer, which has strong batch effects[20], in which samples are clustered by batches instead of by conditions (Fig. 3a). After correcting for batch effect in the dataset at the sgRNA count level using the ComBat[31] function (incorporated into MAGeCKFlute), the biological replicates are properly clustered together (Fig. 3b). This indicates that the batch effect has been removed.

### Screen hit identification

The first method used to identify gene hits is MAGeCK RRA. MAGeCK RRA allows comparison of two experimental conditions. It can identify sgRNAs and corresponding genes that are substantially selected between the two conditions. MAGeCK RRA ranks sgRNAs on the basis of their *P* values calculated from the negative binomial model and uses a modified RRA algorithm named α-RRA to identify positively or negatively selected genes. MAGeCK RRA uses the RRA enrichment score to indicate the essentiality of a gene.

An alternative method that can model complex experimental designs is MAGeCK MLE, which can be used to analyze data from screens with multiple conditions, such as a typical drug screen that

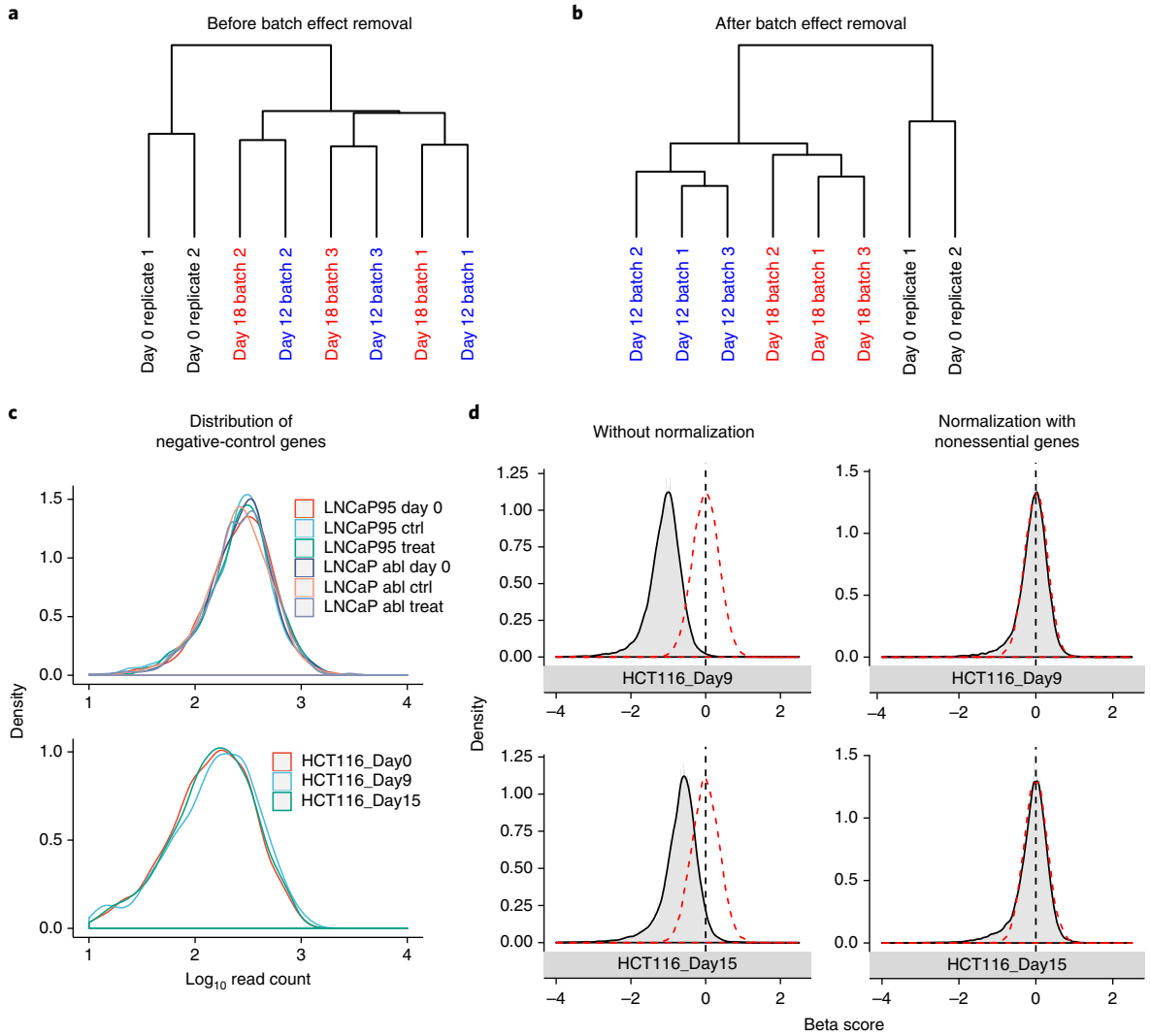


**Fig. 3 | Batch effect correction and normalization of read counts and beta scores from CRISPR screen data.** The data analyzed here are from a genome-wide CRISPR screen using HCT116 colorectal carcinoma cells[8] collected at several time points. Days 0, 12 and 18 were selected to demonstrate batch effects. Another two time points, days 9 and 15, were selected to demonstrate negative normalization with nonessential genes. **a**, Before and (**b**) after batch effect correction of HCT116 CRISPR screen data using ComBat[44]. The sgRNAs from several time points (days 0, 12 and 18) were collected and sequenced. Each time point contains more than one replicate, but the replicates were generated independently. Different batches are shown in different colors, and replicates are marked by the numbers 1, 2, and 3. **c**, Density plot of read counts from gRNAs corresponding to negative-control genes *AAVS1*, *CCR5* and *ROSA26* (top) and to nonessential genes (bottom). Data shown here are from the HCT116 genome-wide CRISPR screen and an LNCap dataset (Supplementary Data 4), which are from a genome-wide CRISPR screen and include two cell lines, LNCap95 and LNCap abl. **d**, Beta score distribution of HCT116 CRISPR screen samples before (left) and after (right) normalization using nonessential genes. Samples were collected at two data points (days 9 and 15). The red dashed lines represents a normal distribution with a mean of zero and the same standard deviation as the original beta score. Black dashed lines indicates the mean of the simulated normal distribution. After a correct normalization, the mean of the beta score should be close to zero. ctrl, control; treat, treatment.

includes at least three conditions: a day 0 condition, a control condition (treated with vehicle such as DMSO) and a drug-treated condition. MAGeCK MLE also models sgRNA knockout efficiency, which may vary depending on different sequence contents and chromatin structures. MAGeCK MLE calculates a 'beta score' for each targeted gene to measure the degree of selection upon gene perturbation, similar to the 'log fold-change' measurement in differential expression analysis.

MAGeCK-VISPR further incorporates all functions of MAGeCK and performs QC and visualization of all results using VISPR, a web-based interactive framework. MAGeCK NEST[32] adds features to MAGeCK-VISPR to improve hit calling. First, MAGeCK NEST can improve results using the Network Essentiality Scoring Tool[33] (NEST) to integrate information from protein–protein interaction networks. Second, MAGeCK NEST adopts a maximum-likelihood approach to remove sgRNA outliers, which often have higher G-nucleotide counts. Users should consider using MAGeCK NEST

to improve hit calling if there are many sgRNA outliers or if a high Gini[30] index is observed in the screen data.

**Read-count normalization with negative-control sequences or nonessential genes**

It is often desirable to compare read counts between different conditions in a single experiment. For the purpose of normalization between different growth conditions, an ideal standard would be sgRNAs that target a completely inert genomic location in all the starting cell populations, such that cell proliferation will not be differentially affected under any of the experimental conditions. *AAVS1* is a well-validated locus that can be used to host an exogenous gene sequence. It has an open chromatin structure and is transcription competent. Most importantly, there are no known adverse effects on the cell resulting from the insertion or deletion of the *AAVS1* locus[34,35]. sgRNAs targeting *AAVS1* have similar behavior across samples (Fig. 3c), suggesting that *AAVS1*-targeting sgRNAs may be appropriate controls for read-count normalization. Using *AAVS1*-targeting sgRNAs as controls also mitigates the nuclease-induced toxicity of Cas9 and reduces the overall false-positive rate.

Similar to sgRNAs targeting the *AAVS1* locus, sgRNAs targeting nonessential genes can also be used to normalize read counts. We compiled a list of nonessential genes (Supplementary Data 1) for normalization of CRISPR screens, if *AAVS1*-targeting sgRNAs are not available. Starting from 927 nonessential genes whose knockdown has no substantial effect in multiple CRISPR screens[8], we removed genes that are expressed at low levels in multiple cell lines. We selected genes whose expression ranked in the 5th–100th percentile (Supplementary Fig. 1a) in 98.3% (1,019 out of 1,036) of Cancer Cell Line Encyclopedia (CCLE)[36] cell lines (Supplementary Fig. 1b). 350 out of 937 nonessential genes passed these criteria (Supplementary Fig. 1). The expression distributions of these 350 genes are consistent across hundreds of cancer cell lines (Supplementary Data 1). This suggests that sgRNAs targeting these genes are appropriate controls for normalization of CRISPR screens, if *AAVS1*-targeting sgRNAs are not available (Fig. 3d). MAGeCKFlute also supports read normalization using sgRNAs from a list of nonessential genes, and we suggest including at least 200 nonessential genes in the library to ensure efficient normalization.

**Copy-number bias correction**

The process of inducing double-strand breaks at targeted genomic sites in CRISPR screens triggers DNA damage–response mechanisms and may cause cell-cycle arrest, especially in cells with high copy-number regions targeted[37]. When an amplified region contains a targeted nonessential gene, the observed beta scores often appear more negative than expected (Supplementary Fig. 2a). A negative beta score indicates that knockout of this gene may inhibit cell proliferation or cause cell death. Therefore, false positives are introduced in essential gene identification. We present an optional method (Step 7A(viii)) in this protocol for correcting this copy number–related bias if the corresponding copy-number file is provided by users (the corresponding copy-number file for the example data is provided as Supplementary Data 2). In this method, the relationship between genomic copy number and observed essentiality is quantitatively modeled for each gene in each experiment. The copy-number bias is then adjusted from the observed outcomes, generating corrected beta scores for all affected genes (Supplementary Fig. 2b). This function had been incorporated into the MAGeCKFlute pipeline and can be applied when performing MAGeCK RRA or MLE.

**Beta score normalization with essential genes**

Cells exposed to different conditions (with or without drug treatment) may have different proliferation rates. For example, CDK4/6 inhibitors affect the cell cycle and generally reduce cell proliferation[38]. Therefore, comparing cells that have a faster doubling time to more slowly proliferating cells may lead to biases in hit identification, as genes will appear to have a stronger selection in cell populations that are proliferating more rapidly. This is often the case when comparing samples with and without drug treatment, because many drugs affect cell proliferation. The 'beta score' for each gene indicates the type of selection a gene is undergoing: a positive beta score indicates positive selection, and a negative beta score indicates negative selection. When different samples are cultured for the same time in CRISPR screens, those with shorter doubling times will have more cell cycles of selection; thus, genes in faster-growing cells tend to yield higher absolute beta scores (Supplementary Fig. 3a). To overcome this bias, we generated a list of 625 refined, high-confidence core essential genes (Supplementary Data 3) that can be used to normalize the beta score (for details, see the Supplementary Methods). MAGeCKFlute performs normalization of gene beta scores using the list of core essential genes (Step 11B), assuming that they are equally negatively selected between two samples, even if the two samples have a different baseline proliferation rate. The beta scores of all genes are

normalized on the basis of the median beta score of this refined set of 625 essential genes. After normalization, the slope and *x*-intercept of the regression line of two samples are close to 1 and 0, respectively, which indicates that normalization with essential genes in genome-wide screens makes the beta scores comparable across samples (Supplementary Fig. 3b). For CRISPR screens with a certain treatment, we recommend that users conduct normalization with essential genes to make the beta score comparable between treatment and control samples.

**Differential hit identification upon cell treatment**

After beta score normalization with essential genes, the next step is to identify differential hits between treatment and control conditions by subtracting their beta scores. This differential beta score is used to identify treatment-related screen hits. The cutoff can be specified in the FluteMLE function, with the default at 1 s.d. from the differential beta score mean. We adopted a 'quantile matching' approach to robustly estimate $\sigma$, which is the standard deviation of the beta score $\beta$. $\sigma$ is chosen such that the $(1-p)$ empirical quantile of the absolute values of $\beta$ matches the $(1-p/2)$ theoretical quantile of the prior normal distribution $N(0,\sigma^2)$, where $p$ stands for the quantile of the beta score $\beta$. $p$ is set to 0.32 for 1 s.d. and 0.05 for 2 s.d., which corresponds to 68% and 95% of the beta score falling within 1 and 2 s.d. of the mean, respectively. If we write the theoretical upper quantile of a normal distribution as $Q_N(1-p)$ and the empirical upper quantile of $\beta$ as $Q_{|\beta|}(1-p)$, then $\sigma$ is calculated as

$$\sigma = \frac{Q_{|\beta|}(1-p)}{Q_N\left(1-\frac{p}{2}\right)}$$

**Functional analysis of screen hits**

The functional analysis of screen hits provides information about the biology of the cell system that was queried in the design of the screen. Several types of functional analyses have been widely used, including GO enrichment analysis[27,39] and GSEA analysis[17]. As expected, in simple proliferation screens, core components of housekeeping pathways (e.g., ribosome and spliceosome) are typically negatively selected[5–7], and components of pathways predicted to be cell type–specific have been found to be essential in the predicted cell types[40,41].

MAGeCKFlute incorporates several functional modules that can be used to explore the biological functions of screen hits. We included published enrichment functions derived from the clusterProfiler[28], GOstats[27] and GSEA packages, and added enrich.HGT to test the enrichment of molecular signatures on the basis of the hypergeometric distribution. These functions allow users to specify the size of genes annotated by GO terms, KEGG pathways, MSigDB gene set collections or user-defined gene sets, and then test for their statistical overrepresentation in the screen hits. In some cases, users might be interested in the strong selection of protein complexes or pathways with a small number of genes, so limiting the gene set size would allow such enrichment to be detected rather than being overwhelmed by weak selection of big pathways.

## Materials

### Equipment

**Software**

- MAGeCK v.0.5.6 or newer (https://sourceforge.net/projects/mageck/)
- MAGeCK-VISPR v.0.5.3 or newer (https://bitbucket.org/liulab/mageck-vispr)
- Conda v.4.5.4 or newer (https://conda.io/docs/)
- Python v.3.4.3 or newer (https://www.python.org)
- R v.3.5.0 or newer (https://www.r-project.org) or RStudio (https://www.rstudio.com/)
- sva package v.3.7 or newer (http://bioconductor.org/packages/release/bioc/html/sva.html)

**Hardware**

- A 64-bit computer running either Linux or Mac OS X; ≥4 GB of RAM (16 GB is preferred)

**Data**

We selected five CRISPR screen datasets as example datasets; these are accessible at http://cistrome.org/MAGeCKFlute/

- A CRISPR screen dataset generated from patient-derived GBM sGSCs (Gene Expression Omnibus (GEO) accession number: GSE70038)[20] (Dataset 1). In this screen, cells were collected at two time

points: day 0 (initial time point of the screen) and day 23 (after 23 d of culture). Replicate 1 and replicate 2 are biological replicates. This dataset is in FASTQ format. These data are used to demonstrate how to analyze screen data using MAGeCK RRA.

- The second dataset is a CRISPR screen in a melanoma cancer cell line, A375, treated with PLX[7](Dataset 2). In this case, the cells were collected at two time points, 7 and 14 d after treatment, and were compared to a control (DMSO-treated) condition. The A375 dataset provides a raw read-count table for each sgRNA. These data contain three conditions (day 0, DMSO treatment and drug treatment) and are used to demonstrate how to analyze a screen with more than two conditions using MAGeCK MLE.
- The HCT116 dataset is a genome-wide CRISPR screen using HCT116 colorectal carcinoma cells[8]. The sgRNAs from several time points were collected and sequenced. This dataset was generated in different batches, and a read-count table is included in the demo data. We use this dataset to demonstrate how to perform batch effect removal.
- The HL60 dataset is a genome-wide CRISPR screen using the acute myelocytic leukemia HL60 cell line[42] with copy-number variation (CNV) information. Cells were collected at two time points: HL60_initial (initial time point of the screen) and HL60_final (after 12 doubling times). We use this screen to demonstrate copy-number bias correction.
- The LNCap dataset (Supplementary Data 4) is a genome-wide CRISPR screen dataset from two cell lines, LNCap95 and LNCap abl[43]. Both contain three conditions (day 0, DMSO treatment and drug treatment) and include *AAVS1*-targeting sgRNAs as negative controls, so they are used to demonstrate the normalization with AAVS1-targeting sgRNAs.

## Equipment setup

### Software setup

Most of the commands given in the protocol run in a typical Linux or Mac shell prompt, and all commands should be run in the same directory as that of the data files. The protocol also includes R scripts. Commands meant to be executed from the Linux or Mac shell (e.g., `bash` commands) are prefixed with a '$' character. Commands meant to be run from either an R script or at the R interactive shell are prefixed with a '>' character.

## Procedure

### Installation of MAGeCKFlute ● Timing ~30 min

1 Start an R session with a terminal or an integrated development environment, such as Rstudio:

```
$R
```

Install MAGeCKFlute, from either the Liu lab, using option A, or Bioconductor, using option B:

(A) **Install MAGeCKFlute from the Liu lab**

(i) Install MAGeCKFlute using the following commands:

```
>install.packages("devtools")
>library('devtools')
>install_bitbucket("liulab/MAGeCKFlute")
```

(B) **Install MAGeCKFlute from Bioconductor**

(i) Install MAGeCKFlute using the following commands:

```
>source('http://www.bioconductor.org/biocLite.R')
>biocLite('MAGeCKFlute')
```

2 Test whether the MAGeCKFlute package (http://www.bioconductor.org/packages/release/bioc/html/MAGeCKFlute.html) was installed successfully, using the following command:

```
>library('MAGeCKFlute')
```

If no error occurs in the loading of the package, it means MAGeCKFlute was installed successfully.

**? TROUBLESHOOTING**

### Downloading and installation of MAGeCK and MAGeCK-VISPR ● Timing 20 min

3 MAGeCK and MAGeCK-VISPR can be installed with either conda (option A) or source code (option B).

(A) **Install MAGeCK and MAGeCK-VISPR with conda**

(i) To install MAGeCK and MAGeCK-VISPR, installation of the Python variant included in the Miniconda Python distribution is required. Download Miniconda (http://conda.pydata.org/miniconda.html) and locate the download directory, then install Miniconda by executing the following command in a terminal:

```
$bash path/to/file/Miniconda3-latest-Linux-x86_64.sh
```

where 'path/to/file' is the directory containing the Miniconda installation file. When the question below appears, answer 'yes':

```
Do you wish the installer to prepend the Miniconda3 install
location to PATH ...? [yes|no]
```

**! CAUTION** Python 2 is incompatible with MAGeCK and MAGeCK-VISPR.

(ii) Afterward, add a bioconda channel, using the following command:

```
$conda config --add channels conda-forge
$conda config --add channels bioconda
```

**▲ CRITICAL STEP** Addition of this channel is essential, as MAGeCK and MAGeCK-VISPR depend on it. It is important to add conda-forge and bioconda in the order specified above to ensure that bioconda has the highest priority. This allows the setup to be run properly.

(iii) Then create an isolated software environment for MAGeCK-VISPR by executing the following command in a terminal:

```
$conda create -n mageck-vispr mageck mageck-vispr python=3
```

(iv) Activate the environment with the following command:

```
$source activate mageck-vispr
```

(B) **Install MAGeCK and MAGeCK-VISPR with source code**

(i) MAGeCK (v.0.3 and later) supports a standard Python installation procedure, with compiling and installation of the software. First, download the source code from the website (https://sourceforge.net/p/mageck/wiki/Home/) and locate the download location. Then unzip the files and go into the unzipped directory with the following commands:

```
$tar xvzf mageck-0.5.6.tar.gz
$cd mageck-0.5.6
```

(ii) Invoke the Python setup.py file, using the following command:

```
$python3 setup.py install
```

(iii) MAGeCK-VISPR can be installed with source code. First, download the source code and go into the source code directory using the following command:

```
$git clone git@bitbucket.org:liulab/mageck-vispr.git
$cd mageck-vispr
```

(iv) Invoke the Python setup.py file, using the following command:

```
$python3 setup.py install
```

**Table 2 | Example of an sgRNA library file**

| ID | sgRNA | Gene_symbol |
|---|---|---|
| s_1 | ACCTGTAGTTGCCGGCGTGC | A1BG |
| s_10 | CCCACAGACGCCTCAGTCTC | A2M |
| s_100 | CCGTGAGCAGGCAGTTCCGC | AATK |

### (Optional) Installation of MAGeCK NEST ● Timing 5 min

4 Download the source code from https://bitbucket.org/liulab/mageck_nest. After the 'mageck_nest-3.0.tar.gz' file is downloaded, unzip the source code file by typing

```
$cd path/to/file/
$tar -zxvf mageck_nest-3.0.tar.gz
```

The `path/to/file/` is the path for the 'mageck_nest-3.0.tar.gz' file.

5 Then change the work directory to 'mageck_nest-3.0' as follows:

```
$cd mageck_nest-3.0
```

6 Finally, install MAGeCK NEST, using the following command:

```
$python3 setup.py install
```

### Processing of CRISPR screen data with MAGeCK or MAGeCK-VISPR

7 CRISPR screen data can be processed with MAGeCK (option A) or using MAGeCK-VISPR (option B). In a typical use case, CRISPR screen data are processed with MAGeCK (option A) step by step. If users want to perform QC and visualize the results, we recommend MAGeCK-VISPR (option B) instead. To illustrate the procedure, we use the two test CRISPR screen datasets described in the 'Equipment' section.

(A) **Process CRISPR screen data step by step with MAGeCK ● Timing 1.5 h**

(i) Download and unzip the test data for both datasets, using the following commands:

```
$ wget http://cistrome.org/MAGeCKFlute/demo.tar.gz
$ tar zxvf demo.tar.gz
$ cd demo_data
```

(ii) Generate a count table for Dataset 1 with the `mageck count` function, by first changing the working directory to a directory that contains raw .fastq data and is able to store the output of `mageck count` as follows:

```
$cd path/to/demo_data/mageck_count
```

▲ **CRITICAL STEP** The command `mageck count` aligns reads onto a sgRNA library and generates a read-count table. The count table can be used directly in the downstream analysis. The command requires a known sgRNA library file (library.csv, included in Dataset 1), in which columns 1–3 are sgRNA names, sequences and target genes, respectively. The library file is either in .txt or .csv format. For an example sgRNA library file, see Table 2.

(iii) To run the `mageck count` on Dataset 1, type the following command:

```
$mageck count -l library.csv -n GSC_0131 --sample-label day0_r1,
day0_r2,day23_r1,day23_r2 --fastq GSC_0131_Day0_Rep1.fastq.gz
GSC_0131_Day0_Rep2.fastq.gz GSC_0131_Day23_Rep1.fastq.gz GSC_0131_
Day23_Rep2.fastq.gz
```

**Box 1 | Additional mageck count parameters**

Several additional key parameters that can be used to run MAGeCK in Step 7A(iii) are as follows:

| | |
|---|---|
| `--trim-5` | Sets the length for trimming of the 5′-ends of the reads; the default is `AUTO` (MAGeCK will automatically determine the trimming length). |
| `--sgrna-len` | Sets the length of the sgRNA; the default is `AUTO` (MAGeCK will automatically determine the sgRNA length). Use this only if you have turned on the '`--unmapped-to-file`' option. |
| `--count-n` | Counts sgRNAs with Ns. By default, sgRNAs containing N will be discarded. |
| `--unmapped-to-file` | Saves unmapped reads to a file, with sgRNA lengths specified by the `--sgrna-len` option. |

**Table 3 | Example of a batch matrix file**

| Sample name | Batch | Conditions |
|---|---|---|
| HCT116_2_T18A | 1 | 1 |
| HCT116_2_T18B | 2 | 1 |
| HCT116_2_T18C | 3 | 1 |
| HCT116_2_T12A | 1 | 2 |
| HCT116_2_T12B | 2 | 2 |
| HCT116_2_T12C | 3 | 2 |
| HCT116_1_T0 | 1 | 3 |
| HCT116_2_T0 | 2 | 3 |

The meanings of the parameters in this command are as follows (see Box 1 for further parameters that could be used or see all the parameters by typing the command `mageck count -h`):

| | |
|---|---|
| `-l` | The provided sgRNA library file, including the sgRNA ID, the sequence and the gene it is targeting (Table 2). |
| `-n` | The prefix of the output files. |
| `--sample-label` | The sample labels, separated by a comma (,). Must be equal to the number of samples provided (in the `--fastq` option). Default '`sample1, sample2, ...`'. |
| `--fastq` | The sample .fastq files (or .fastq.gz files), separated by a space; use a comma (,) to indicate technical replicates of the same sample. |

(iv) *(Optional) Batch effect removal.* If any portion of the screen was performed in batches (at separate times or with different reagents), we recommend running Combat in the sva package[44] to remove possible batch effects, as follows. To run the package, a BatchMatrix file (see Table 3 for format) is required. This file can be generated using a text editor and saved as .txt file, and items in this file should be separated by tabs. In this file, columns 1–3 are sample names that correspond to a raw count table; batch covariate, which could be numbers that represent batches; and another covariate in addition to batch (optional), respectively. Most commonly, the additional covariate in column 3 would be a number representing an experimental condition. The easiest approach is to put all needed files in the same folder. Alternatively, provide full paths for the files and scripts in the command. Here, using the HCT116 dataset as an example, we demonstrate how to remove batch effects from any screen that has been performed in batches. The batch effect–removed count table can be used as the input for MAGeCK RRA or MLE. To run the batch effect–removal package, initiate R and type the following commands:

```
$R
> library(MAGeCKFlute)
> BatchRemove(mat = "rawcount.txt", batchMat = "BatchMatrix.
txt", prefix = "BatchCorrect", -pca = T, -cluster = T, -outdir =
".")
```

**Table 4 | Example of count table**

| sgRNA | Gene | day0_r1 | day0_r2 | day23_r1 | day23_r2 |
|---|---|---|---|---|---|
| s_48202 | *RPL* | 44 | 45 | 44 | 29 |
| s_47147 | *RHBDD* | 477 | 472 | 445 | 560 |
| s_48746 | *RWDD2B* | 487 | 405 | 644 | 587 |

In this command, the meanings of the parameters are as follows:

`-mat` Matrix, or file path of data.

`-batchMat` Matrix or file path of the batch table, which has at least three columns: 'Samples', 'Batch' and 'Covariates'.

`-cov` Specifies the covariates in addition to batch, such as treatment condition, which can be used to model the outcome.

`-log2trans` Boolean, specifying whether to do $\log_2$ transition before batch removal.

`-pca` Boolean, specifying whether to do principal component analysis before and after batch removal.

`-cluster` Boolean, specifying whether to do cluster analysis before and after batch removal.

`-prefix` Character, specifying prefix of output figures; it is needed only if `cluster/pca` is `TRUE`.

`-outdir` Output directory on disk.

(v) *Identify screen hits using MAGeCK RRA*. Use the `mageck test` subcommand to perform MAGeCK RRA for comparison between two conditions, such as an initial condition versus cells cultured for a period of time. The input of `mageck test` is a count table that can be generated by the `mageck count` command (Step 7A(iii)) or other alignment tools, such as bowtie[45] or bwa[46]. To identify screen hits from Dataset 1, type the following:

```
$mageck test -k GSC_0131.count.txt -t day23_r1,day23_r2 -c
day0_r1,day0_r2 -n GSC_0131_rra --remove-zero both --remove-
zero-threshold 0
```

In this command, the meanings of the parameters are as follows:

`-k` Count table (Table 4, Step 7A(iii)); provides a tab-separated count table. Each line in the table should include sgRNA name (first column), target gene (second column) and read counts (third column) for each sample.

`-t` Sample labels or sample indexes (0 as the first sample, according to Python standard) in the count table that are to be treated as treatment experiments, separated by commas (,). If sample labels are provided (rather than a sample index), the labels must match the labels in the first line of the count table. This parameter is required, which means at least one sample should be assigned to this parameter.

`-c` Sample labels or sample indexes in the count table that are to be treated as control experiments, separated by commas (,). If no samples are specified by this parameter, controls will be defined as all the samples not specified by the `-t` parameter.

`-n` The prefix of the output files.

`--remove-zero` Remove sgRNAs whose mean value is zero in control, treatment, both control and treatment or any control or treatment sample. Default: both (remove those sgRNAs that are zero in both control and treatment samples).

`--remove-zero-threshold` sgRNA-normalized count threshold to be considered removed in the `--remove-zero` option. Default = 0.

Use the command `mageck test -h` to see additional parameters.

**Box 2 | Use of MAGeCK NEST to improve the accuracy of hit calling**

MAGeCK NEST provides additional features to MAGeCK MLE to improve hit calling. First, MAGeCK NEST can improve results using the Network Essentiality Scoring Tool (NEST) to integrate information from protein-protein interaction networks. Second, MAGeCK NEST adopts a maximum-likelihood approach to remove sgRNA outliers, which often have higher G-nucleotide counts. If the Gini index of the read count is high (such as >0.2) or there are a lot of sgRNA outliers in the screen data, users can consider using MAGeCK NEST to improve hit calling. The input and output files of MAGeCK NEST are as same as for MAGeCK MLE. MAGeCK NEST uses the parameter `-e` to specify negative-control genes. To identify screen hits with MAGeCK NEST, use the following command:

```
$mageck_nest.py nest -k rawcount.txt -d designmatrix.txt -n nest_res --norm-method control -e negative_control_genes.txt
```

**Box 3 | Format of design matrix file**

The design matrix file (Table 5) is a binary matrix indicating which sample (contained in the first column) is affected by which condition (contained in the second and subsequent columns). Values under the headers are binary. The element in the design matrix, $d_{ij}$, equals '1' if sample $i$ is affected by condition $j$, and 0 if it is not. Each column of the design matrix file should be separated by a tab character. This file can be created with a text-editing software and saved as a plain-text file.

The following rules apply to the design matrix file:

- The design matrix file must include a header line of condition labels.
- The first column consists of the sample labels, which must match the sample labels in the read-count file.
- The non-header values in columns 2 and beyond must be either '0' or '1'.
- The second column defines an initial condition that affects all samples and must be '1' for all rows (except the header row).
- The design matrix file must contain at least one sample representing the 'initial state' (e.g., day 0) that has only a single '1' in the corresponding row. That single '1' must be in the 'initial condition' column (the second column). MAGeCK MLE will calculate the beta score by comparing the other conditions.

▲ **CRITICAL STEP** Note that rather than using MAGeCK RRA in this step, it is possible instead to use MAGeCK MLE, as described in the next steps, Step 7A(vi and vii).

(vi) *(Optional) Identify screen hits using MAGeCK MLE.* If an experiment contains more than two conditions, for example, a three-condition design: day 0, drug treatment and DMSO treatment, we recommend using MAGeCK MLE or MAGeCK-NEST (Box 2) instead of MAGeCK RRA to perform the previous step. To do this, access the directory of raw counts (generated with `MAGeCK count` in Step 7A(ii)) with the following command:

```
$ cd path/to/demo_data/mageck_mle
```

`path/to/demo_data/` should point to either the demo data (Step 7A(i)) or to the user's own raw count data. In the following step (Step 7A(vii)), we use MAGeCK MLE to obtain gene hits from the screen in Dataset 2 (Equipment).

▲ **CRITICAL STEP** The input for the MAGeCK MLE function should be a raw count table in which columns are samples and rows are sgRNAs. Generally, this raw count table is generated from a FASTQ file using the MAGeCK count function in Step 7A(iii). Because Dataset 2 provides the raw count table instead of the FASTQ file, we used it directly as the input for MAGeCK MLE.

(vii) (Optional) Run MAGeCK MLE with the following command:

```
$mageck mle --count-table rawcount.txt --design-matrix designmatrix.txt --norm-method control --control-sgrna nonessential_ctrl_sgrna_list.txt --output-prefix braf.mle
```

Here is a description of the key parameters of `mageck mle` (to see all the parameters, use the command `mageck mle -h`):

`--count-table` Provides a tab-separated count table. Each line in the table should include an sgRNA name (first column), a target gene (second column) and the read count in each sample (third and subsequent columns).

**Table 5 | An example of a design matrix file**

| Samples | Initial condition | Mock treatment | Drug treatment |
|---|---|---|---|
| Day 0 | 1 | 0 | 0 |
| D7_R1 | 1 | 1 | 0 |
| D7_R2 | 1 | 1 | 0 |
| PLX7_R1 | 1 | 0 | 1 |
| PLX7_R2 | 1 | 0 | 1 |

`--design-matrix` Provides a design matrix (for instructions for generating the design matrix, see Box 3), either as a file name or a quoted string of the design matrix. An example of a design matrix is shown in Table 5. The rows of the design matrix must match the order of the samples in the count table (if `--include-samples` is not specified), or the order of the samples is specified by the `--include-samples` option.

`--norm-method{none, median, total, control}` Method for normalization, including 'none' (no normalization), 'median' (median normalization, default), 'total' (normalization by total read counts), 'control' (normalization by control sgRNAs specified by the `--control-sgrna` option).

`--control-sgrna` A list of control sgRNAs.

`--output-prefix` The prefix of the output file(s). Default is 'sample1'.

▲ **CRITICAL STEP** We recommend that sgRNAs targeting negative control loci, such as *AAVs1*, *CCR5* and *ROSA26*, be included in a custom sgRNA library. These sgRNAs can be used as negative controls to normalize screen data. If the library includes such sgRNAs, they can be specified in the command line with the parameters `--norm-method control` and `--control-sgrna`, as shown above. If these negative-control sgRNAs were not included in the library, sgRNAs targeting nonessential genes can be used as negative controls for normalization, as described in the parameter specifications above. We generated a nonessential gene list that includes 350 genes as described in the 'Experimental design' section. Users can use the `--control-sgrna` parameter to provide sgRNAs corresponding to these genes to perform normalization with these nonessential genes. The control sgRNA file should be a tab-separated table, with only a single column that includes the IDs of sgRNAs. It needs to be prepared by the user and saved as a .txt file. The nonessential_ctrl_sgrna_list.txt file contains sgRNAs targeting the 350 nonessential genes specified in the 'Experimental design', which we use to perform negative-control normalization.

**? TROUBLESHOOTING**

(viii) *(Optional) Correct copy-number bias.* MAGeCK RRA and MAGeCK MLE contain an optional method to correct copy-number biases in the calculated RRA scores and beta scores, respectively. We recommend that users perform copy-number bias correction if the CNV information is available for the cell line. Both MAGeCK RRA and MAGeCK MLE require a tab-delimited file containing copy-number values for each gene across the cell line(s) associated with the experiment. The copy-number file contains two columns: gene name and copy number (Supplementary Data 2). The data analyzed here are from an HL60 cell line (see 'Equipment'). The name of this file is incorporated into the analysis with the parameter `--cnv-norm`. For MAGeCK RRA, an additional parameter, `--cell-line`, is required to specify the name of the cell line (from the copy-number data file) to be used in the bias correction method. To perform copy-number correction in MAGeCK MLE, the name of the cell line (from the copy-number data file) must match sample labels in the design matrix file. To perform MAGeCK RRA with copy-number bias correction, type the following:

```
$mageck test -k rawcount.txt -t HL60.final -c HL60.initial -n rra_
cnv --cnv-norm cnv_data.txt –cell-line HL60_HAEMATOPOIETIC_AND_
LYMPHOID_TISSUE
```

(ix) (Optional) We use the HL60 dataset (see 'Equipment') to demonstrate how to perform copy-number correction with MAGeCK MLE. To perform MAGeCK MLE with copy-number bias correction, type the following:

```
$mageck mle --count-table rawcount.txt --design-matrix designma-
trix.txt --cnv-norm cnv_data.txt
```

(B) **Process CRISPR screen data with MAGeCK-VISPR** ● **Timing 1.5 h**

(i) Activate the MAGeCK-VISPR environment as follows. If MAGeCK-VISPR was installed with conda (Step 3A(i–iii)), make sure to activate the corresponding conda environment (Step 3A(iv)) before conducting any MAGeCK-VISPR-related command:

```
$source activate mageck-vispr
```

(ii) Choose a workflow directory and initialize the workflow with the .fastq or .fastq.gz files that contain the raw reads (downloaded in Step 7A(i)). If the raw read files are not in the working directory, remember to include a path when specifying the raw read files. MAGeCK will install a 'README' file, a config file. 'config.yaml'. and a Snakemake workflow definition (a 'Snakefile') to the given directory. Initialize the workflow as follows:

```
$mageck-vispr init workflow --reads path/to/file/*.fastq*
```

Here, the term '`workflow`' can be changed to any name. The parameter `--reads` is used to specify the .fastq or .fastq.gz files to be analyzed. The `path/to/file` is the directory containing the .fastq or .fastq.gz files.

(iii) (Optional) Alternatively, MAGeCK-VISPR supports analysis with a raw count table (e.g., the rawcount.txt file in Dataset 2). To run MAGeCK-VISPR with raw counts, specify the raw count file in the 'config.yaml' file instead of using the parameter `–reads`. A short video (Supplementary Video 1) describing how to edit the 'config.yaml' file can be found at https://www.youtube.com/watch?v=3maSxhy1JL0.

```
$mageck-vispr init workflow
```

(iv) Configure the workflow, using the following command:

```
$cd workflow
```

Next, specify the path to the library file (Table 2) and the normalization method by changing the 'config.yaml' file. The sgRNAs used for normalization and batch information must be provided as well, if the data were generated from different batches. CNV correction is recommended if CNV data are available.

(v) To check whether the 'config.yaml' files have been configured correctly, enter the following command line into the terminal:

```
$snakemake –n
```

**? TROUBLESHOOTING**

(vi) Execute the workflow.

The execution of the workflow must specify how many CPU cores should be assigned to this process. In general, three to four cores should be sufficient, but an increase in cores will reduce the running time. The workflow can be performed, in this case with eight cores, using the following command:

```
$snakemake --cores 8
```

(vii) (Optional) Visualize the results with VISPR.

MAGeCK-VISPR also provides a web-based visualization framework (VISPR) for an interactive exploration of CRISPR screen QC and analysis of results. Once the workflow

**Box 4 | Additional FluteMLE parameters**

Additional key parameters that can be used to run FluteMLE in Step 11B(ii):

| | |
|---|---|
| `-top` | An integer, specifying the number of top-selected genes labeled in rank figure. |
| `-bottom` | An integer, specifying the number of bottom selected genes labeled in rank figure. |
| `-interstGenes` | A character vector, specifying the genes of interest labeled in rank figure. |
| `-pvalueCutoff` | A numeric, specifying the false discovery rate (FDR) cutoff of enrichment analysis. |
| `-adjust` | One of 'holm', 'hochberg', 'hommel', 'bonferroni', 'BH', 'BY', 'fdr', 'none'. |
| `-enrichkegg` | One of 'ORT' (over-representing test), 'GSEA' (gene set enrichment analysis), 'DAVID', 'GOstats' and 'HGT' (hypergeometric test), or index from 1 to 5, specifying the enrichment method used for KEGG enrichment analysis. |
| `-gsea` | A Boolean value that indicates whether GSEA analysis is performed. |
| `-posControl` | A file path or a character vector, specifying the positive control genes used for cell cycle normalization, if NULL, use built-in essential gene list. |
| `-loess` | Boolean, specify whether to include loess normalization in the pipeline. |
| `-view_allpath` | Boolean, specify if all pathway view figures are output. |
| `-outdir` | Output directory on disk. |

execution has finished, visualize the generated results and QCs by typing the following command into the terminal:

```
$vispr server results/*.vispr.yaml
```

This will generate the following output:

```
Loading data.
Starting server.
Open: go to http://127.0.0.1:5000 in your browser.
Note: Safari and Internet Explorer are currently unsupported.
Close: hit Ctrl-C in this terminal.
```

You can then copy and paste the link (http://127.0.0.1:5000) into a browser (Chrome or Firefox is recommended) to visualize the results.

### Downstream analysis pipeline ● Timing ~1 h

8 Open a new plotting script in the terminal or use the R interactive shell as follows:

```
$R
```

9 Load the MAGeCKFlute package into the R environment:

```
>library(MAGeCKFlute)
```

10 Set the working directory to the directory of the output files for MAGeCK, for example:

```
>setwd('path/to/file/')
```

where 'path/to/file/' is the location of the output files for MAGeCK.

11 Functional analysis can be carried out using results obtained from MAGeCK RRA (option A) or from MAGeCK MLE (option B).

(A) **Functional analysis for MAGeCK RRA results** ● Timing **1 h**

(i) To perform functional analysis for MAGeCK RRA results (from Dataset 1, generated in Step 7A(v)), enter the following command in the R environment:

```
>FluteRRA(gene_summary = "path/to/file/rra.gene_summary.txt",
prefix="FluteRRA", organism="hsa")
```

The file 'rra.gene_summary.txt' is included in the output files from MAGeCK RRA, which was performed at Step 7A(v) or 7B(vi).

(B) **Functional analysis of MAGeCK MLE results** ● **Timing 1 h**

(i) To perform essential gene normalization and functional analysis of MAGeCK MLE results (from Dataset 2, generated in Step 7A(vii)), enter the following command:

```
>FluteMLE(gene_summary="path/to/file/mle.gene_summary.txt",
ctrlname="dmso", treatname="plx", organism="hsa", prefix="FluteMLE",
-pathway_limit = c(3,50))
```

The file 'mle.gene_summary.txt' is included in the output files from MAGeCK MLE, which was performed at Step 7A(vii) or 7B(vi). FluteMLE performs essential gene normalization automatically, and a custom essential gene list can be specified by the parameter `-posControl`. The meanings of the parameters in this command are as follows (see Box 4 for additional parameters that could be used, or see all the parameters with the command `help("FluteMLE")` in R):

| | |
|---|---|
| `-gene_summary` | Either a file or a data frame whose column name contains 'Gene', 'dmso.beta' and 'plx.beta', which correspond to the parameters `-ctrlname` and `-treatname`. In this specific MAGeCK MLE result (from Dataset 2, generated in Step 7A(vii)), 'dmso.beta' and 'plx.beta' correspond to the control (DMSO) and drug treatment conditions (PLX), respectively. |
| `-ctrlname` | A character vector, specifying the name of control samples. |
| `-treatname` | A character vector, specifying the name of treatment samples. |
| `-organism` | A character, specifying an organism, such as 'hsa' or 'Human' (default), and 'mmu' or 'Mouse'. |
| `-prefix` | A character, indicating the prefix of the output file name. |
| `-pathway_limit` | A two-length vector (default: `c(3, 50)`), specifying the minimal and maximal size of gene sets for enrichment analysis. |

▲ **CRITICAL STEP** After the command line finishes processing, and if there is no error occurrence, users can check the current directory to verify the existence of the files listed in the 'Anticipated results' section. Before running FluteMLE, ensure that the 'gene_summary.txt file' is in the current working directory, or that the file path name is inserted into the command.

**? TROUBLESHOOTING**

## Troubleshooting

Troubleshooting advice can be found in Table 6.

**Table 6 | Troubleshooting table**

| Step | Problem | Possible reason | Solution |
|---|---|---|---|
| 2 | Package installation failed | R version is old or dependent packages cannot be installed | Update R to v.3.5.0 or newer. Try to start the R session from the root folder. The PATH variable can be set by typing the following into a terminal: For Linux `$export PATH="/usr/bin:$PATH"` For MacOS `$export PATH="/usr/local/bin:$PATH"` |
| 7A(vii) | MAGeCK crashed with an error related to the design matrix, such as "Error parsing the design matrix: 0 row sums for some samples." | Design matrix was not in the correct format | Follow the 'pipeline' tutorial at https://bitbucket.org/liulab/mageck-vispr to generate a correct design matrix |
| 7B(v) | MAGeCK-VISPR crashed with an error due to inability to find input files: "Error in configuration file (key=library, entry=xxx): File does not exist." | File names or paths of the input files are incorrect | Double-check the names and paths of the input files and make sure that these files can be accessed |
| 11B (i) | FluteMLE crashed with an error due to inability to find samples: "Error in FluteMLE(gene_summary = 'mle.gene_summary.txt', treatname = 'treatment.beta',: No sample found!" | The index specified with the parameter `'treatname'` or `'ctrlname'` does not fit the column names of the 'gene_summary.txt' file | Ignore the suffix '.beta' when specifying the control names and treatment names with the parameters `'treatname'` and `'ctrlname'` |

**Table 7 | Results of MAGeCK RRA**

| File name | Description |
|---|---|
| sgrna_summary.txt | The sgRNA rank results |
| gene_summary.txt | The gene rank results |
| log | The log information during the run |
| R | The R code that can be used to plot summary figures of the results |

## Timing

Running this protocol on the example data provided will take ~3 h on a machine with eight processing cores and at least 8 GB of RAM. However, larger datasets with more samples or deeper sequencing runs may take longer, and timing will vary across different computers.
Steps 1 and 2, installation of MAGeCKFlute: ~30 min
Step 3, downloading and installation of MAGeCK and MAGeCK-VISPR: 20 min
Steps 4–6, (optional) installation of MAGeCK NEST: 5 min
Step 7A, processing of CRISPR screen data step by step with MAGeCK: 1.5 h
Step 7B, processing of CRISPR screen data with MAGeCK-VISPR: 1.5 h
Steps 8–11A(i), functional analysis of MAGeCK RRA results: ~1 h
Steps 8–11B(i), functional analysis of MAGeCK MLE results:~1 h

## Anticipated results

### Results of MAGeCK count (Step 7A(iii))

The main output of `mageck count` includes a raw count table 'count.txt', a normalized count table 'count_normalized.txt' and a summary table of the mapping results 'countsummary.txt'. The raw count table contains the raw sequencing read counts of each sgRNA for each sample, and the normalized count table records the normalized count, using the default median normalization method or an alternative method specified by the user. The 'countsummary.txt' file includes the total reads, mapped reads, mapped percentage, zero-count number and Gini index[30] of each sample. The 'zero count' number indicates the number of sgRNAs that have a read count of 0 in that sample. Screens with a large number of zero-count sgRNAs in the initial condition, after transfection, or after selection might indicate insufficient cell representation of the library complexity. MAGeCK count analysis on Dataset 1 yields ~65% mapped reads for both replicates collected at days 0 and 23 (Fig. 2a). The correlations between replicates are >0.9 (Fig. 2b). The Gini indices are <0.1 (Fig. 2c), and the missing sgRNAs are <1% (Fig. 2d).

### Results of MAGeCK RRA (Step 7A(v))

The main output of the MAGeCK RRA consists of the files listed in Table 7.

The most important output of MAGeCK RRA is the file 'gene_summary.txt'. For each gene, MAGeCK RRA outputs a score for both negative selection and positive selection. In either case, lower scores indicate a higher level of selection. MAGeCK RRA also outputs a *P* value or false-discovery rate (FDR) for the scores of each gene.

MAGeCK RRA directly performs some basic analysis at the sgRNA level and outputs the result to 'sgrna_summary.txt'. This file contains normalized read counts for each sample, the mean counts in control and treatment sample(s), and the log-fold-change, *P* value and FDR of each sgRNA in the comparison. The details of each file can be found in the MAGeCK documentation, using the following link: https://sourceforge.net/p/mageck/wiki/output/.

### Results of MAGeCK MLE and MAGeCK NEST (Step 7A(vi) and Box 2)

MAGeCK MLE and MAGeCK NEST generate files that are similar to MAGeCK RRA, such as a 'log' file, a 'gene_summary' file (including gene beta scores) and a 'sgrna_summary' file (including sgRNA efficiency probability predictions) (Table 7). The 'gene_summary' file includes the beta scores of the conditions specified in the design matrix, except for the initial condition, and the associated statistics. The 'p-value' is calculated by randomly permuting sgRNA labels. The 'fdr' is the FDR calculated by

the Benjamini–Hochberg procedure. Similarly, the 'wald-p-value' and 'wald-fdr' are the *P* value and FDR, respectively, calculated by the Wald test to determine whether the corresponding 'beta score' differs substantially from zero in the MAGeCK MLE model. A detailed description of the output files from MAGeCK MLE can also be found in the MAGeCK documentation at the following link: https://sourceforge.net/p/mageck/wiki/output/.

### Results of MAGeCK-VISPR (Step 7B)

All the results from MAGeCK-VISPR will be written into the 'result' folder. If there are no errors running MAGeCK-VISPR, users will see three subfolders in the 'result' folder. The 'count' subfolder includes all of the outputs from `mageck count`: raw count, normalized count and the summary of count files. The 'QC' subfolder includes the QC of the reads at the sequence level for each sample, which is generated by FASTQC (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/). MAGeCK-VISPR also performs QC at the read-count level, including mapping ratio (Fig. 2a), correlation between samples (Fig. 2b), evenness of sgRNA reads (Fig. 2c) and number of missing sgRNAs (Fig. 2d). The QC result can be visualized by VISRP (Step 7B(vii)). The 'test' subfolder contains the main results of MAGeCK RRA or MLE, including the 'gene_summary.txt' file, which can be used in the functional analysis in Step 11.

### Results of MAGeCKFlute (Step 11)

All pipeline results are written into a local directory 'Prefix_Flute_Results/', and all figures are integrated into a single report file 'Prefix_Flute.mle_summary.pdf'.

#### FluteRRA (Step 11A(i))

MAGeCKFlute processes MAGeCK RRA results ('gene_summary' file) with the function `FluteRRA`, which compares two conditions, such as day 23 versus day 0 in Dataset 1. `FluteRRA` will perform functional analysis with both positively and negatively selected genes. GO enrichment analysis uses the clusterProfiler[28] package as the default enrichment method. The hypergeometric test (HGT) is used as the default method for KEGG enrichment analysis. `FluteRRA` will output both figures similar to Fig. 4c–e and tables of the substantially enriched terms. All the result files are listed in Table 8.

#### FluteMLE (Step 11B(i))

FluteMLE uses the 'gene_summary' file, which is the output of MAGeCK MLE, as its input. The results of FluteMLE will be written into a directory with the name 'prefix_Flute_results' where 'prefix' is the prefix that users specify in the `FluteMLE` function. All the result files are listed in Table 9.

To illustrate the utility of MAGeCKFlute in the downstream analysis of MAGeCK MLE results, we used example data from a public CRISPR screen melanoma cancer cell line, A375 (see 'Equipment'). We analyzed the raw count using the `mageck mle` function, performed normalization with non-essential genes with the parameter `--norm-method` and specified the nonessential gene list (Supplementary Data 1) with the parameter `--control-sgrna`. Normalization with the core essential genes (Supplementary Data 2) was performed using the FluteMLE function by default.

FluteMLE performs QC based on the beta score to ensure that the two conditions, control and treatment, are comparable. The QC results consist of three levels: distribution of the beta score for different conditions (Supplementary Fig. 4a), linear fitting of the beta scores of essential genes (Supplementary Fig. 4b) and an MA plot (Supplementary Fig. 4c). After the normalization, the beta scores of most genes should be close to zero. Therefore, the mean beta score of all the genes should also be close to zero. We observed that the distributions of the beta scores in both treatment and control conditions were similar, making beta scores comparable between different conditions (Supplementary Fig. 4a,b). The MA plot can be used to visualize the differences between the beta scores generated in two samples, by transforming the data onto the intensity difference $M$ ($\beta$T − $\beta$C) and the average intensity $A$ ($\beta$T + $\beta$C); $\beta$T and $\beta$C are the beta scores of the treatment and control samples, respectively. Because the beta scores for most genes will not change markedly in the treatment condition compared to the control condition, the $M$ value for the majority of the genes in the MA plot should be close to zero (Supplementary Fig. 4c). The distribution of the beta scores is in the folder 'Distribution_of_BetaScores'. The folder 'Linear_Fitting_of_BetaScores' contains figures showing the linear regression results of the beta scores for the treatment and the control samples. A summary table of the beta scores for each normalization is also generated in the 'Scatter_Treat_Ctrl' folder.

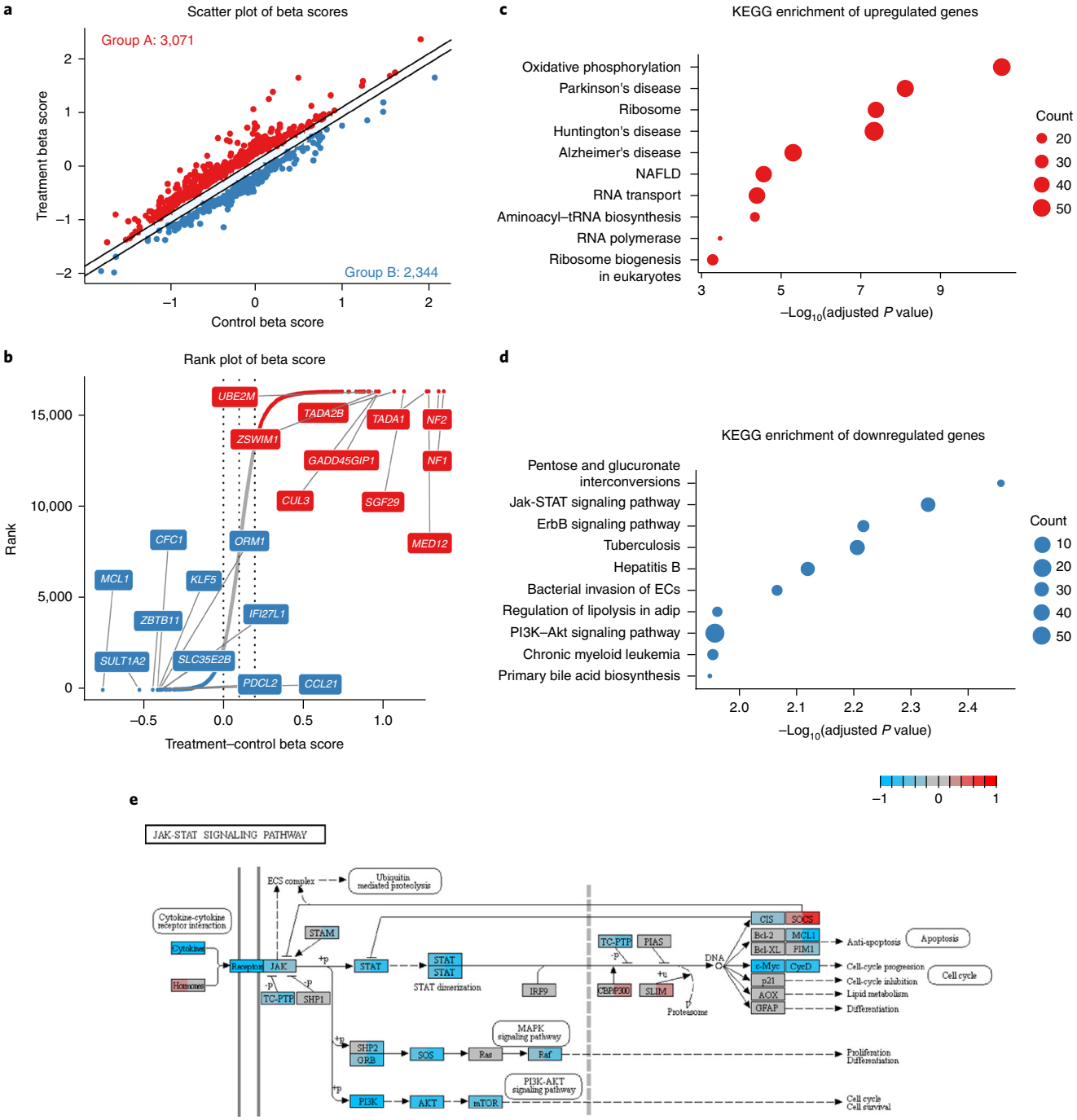


**Fig. 4 | CRISPR–Cas9 screen analysis by MAGeCKFlute.** The data analyzed here are from a CRISPR screen in a melanoma cancer cell line, A375, treated with the BRAF protein kinase inhibitor vemurafenib (PLX)[7]. Data were processed with FluteMLE. **a**, Scatter plot of treatment and control beta scores. The beta scores were normalized using the median of the beta scores of the core essential genes we compiled (Supplementary Data 2, Supplementary Methods). The two diagonal lines indicate ±1 s.d. of the difference between treatment and control beta scores. Red dots (group A) represent genes whose beta score increased after treatment. Blue dots (group B) represent genes whose beta score decreased after treatment. **b**, The genes are sorted based on the differential beta score, which is calculated by subtracting the control beta score from the treatment beta score. The color scheme is the same as in **a**. **c**,**d**, The top ten enriched KEGG pathways with positively (**c**, red, group A) and negatively (**d**, blue, group B) selected genes. The *P* values were calculated with the clusterProfile package, which is based on the hypergeometric distribution. The size of each circle indicates the number of genes that are enriched in the corresponding function. adip, adipocytes; ECs, epithelial cells; NAFLD, non-alcoholic fatty liver disease. **e**, A visualization of treatment and control beta scores over the JAK-STAT signaling pathway generated by the Pathview package[47]. The left and right portions of a gene box represent control and treatment beta scores, respectively. Red indicates a positive beta score; blue indicates a negative beta score; and gray marks genes that are neither positively nor negatively selected. The dashed vertical line in this specific pathway indicates the nuclear membrane.

**Table 8 | Results of FluteRRA**

| Files | Description |
|---|---|
| Prefix_Flute.rra_summary.pdf | The integration of the results |
| bp.neg.png | The enrichment analysis of biological processes using negatively selected genes |
| bp.pos.png | The enrichment analysis of biological processes using positively selected genes |
| kegg.neg.png | The enrichment analysis of KEGG using negatively selected genes |
| kegg.pos.png | The enrichment analysis of KEGG using positively selected genes |

**Table 9 | Results of FluteMLE**

| Files | Description |
|---|---|
| Prefix_Flute.mle_summary.pdf | The integration of the results |
| Distribution_of_BetaScores | The distribution of the beta score: all genes and essential genes under different normalization methods |
| Linear_Fitting_of_BetaScores | The linear fitting of the beta score with different normalization methods |
| MAplot | MA plot of the beta score |
| Scatter_Treat_Ctrl | Scatter plot and rank plot of the treatment and control samples |
| Enrichment_Treat-Ctrl | Functional enrichment analysis of the genes for which the beta scores are substantially different between treatment and control samples |
| Pathview_Treat_Ctrl | The KEGG map of the enriched terms |
| Scatter_9Square | A nine-square scatter plot of the treatment and control beta scores |
| Enrichment_9Square | Functional enrichment analysis of the four separate groups of genes from the nine-square scatter plot, which relate to the experimental conditions |
| Pathview_9Square | The KEGG map of the enriched terms |

After QC, MAGeCKFlute will identify the differences between two treatment conditions (not including day 0), such as samples treated with or without a particular drug. MAGeCKFlute will generate scatter plots (Fig. 4a) and ranking plots (Fig. 4b). After performing FluteMLE on Dataset 2, we observed 3,071 genes with decreased essentiality (group A, red dots) and 2,344 genes with increased essentiality (group B, blue dots) after drug treatment as compared with the control. These groups were used to generate two independent gene lists for further downstream analysis. A ranking plot shows the changes of beta score between treatment and control conditions and uses the same criteria as the scatter plots to identify beta score differences (Fig. 4b). Scatter plots and ranking plots can be found in the 'Scatter_Treat_Ctrl' folder.

The functional annotation will be performed on the basis of the two groups of genes selected in the scatter plot and ranking plot. In the FluteMLE function, the enrichment method and cutoff can be specified with the parameters `-enrich_kegg` and `-pvalueCutoff`, respectively. In this protocol, we performed all the enrichment analysis using the default enrichment method 'HGT' and used the default cutoff of 0.1. Enrichment results (Fig. 4c,d) will be generated in the folder 'Enrichment_Treat-Ctrl'.

MAGeCKFlute utilizes the Pathview package to perform data integration and visualization. The figures generated with Pathview are included in the files 'Pathview_9Square' and 'Pathview_-Treat_Ctrl'. MAGeCKFlute maps and renders data onto relevant pathway graphs, where each gene is colored with two colors in a single pathway graph, based on the beta score under different conditions (Fig. 4e). For each selected gene, the left half is colored based on the beta score of the control samples and the right half is colored according to the beta score of the treatment samples. This approach allows users to explore the variations of the beta scores within one pathway graph. For example, *STAM* is weakly negatively selected in the DMSO control condition and strongly negatively selected in the PLX drug treatment condition (Fig. 4e). Therefore, this gene increases its essentiality upon PLX treatment, suggesting that it has a potentially synthetic effect with PLX.

To identify treatment-related hits accurately, FluteMLE classifies genes into four groups (Supplementary Fig. 4d), determined by differences in the beta scores between the treatment and

control samples (MAGeCK MLE results of Dataset 2). Gene groups that exhibit different beta scores between treatment and control samples are colored to represent these differences. Genes in the green group are strongly negatively selected (i.e., cells whose gene is disrupted are under-represented) in the control samples and are weakly selected (either positively or negatively) in the treatment samples. These genes lost their essentiality after treatment and are potentially located in the pathways targeted by the treatment. The orange group contains genes that are weakly selected in the control samples and strongly positively selected in the treatment sample (i.e., cells whose gene is disrupted are over-represented). These are genes whose loss confers treatment resistance. Genes in the blue group are strongly positively selected in the control sample and weakly selected in the treatment sample. These genes may be either potential regulators of cell proliferation in general or regulators of the treatment target (if the treatment target is an essential gene). Genes in the purple group are weakly selected in the control samples and strongly negatively selected in the treatment samples. These genes are potentially synthetically lethal in combination with the drug treatment. Figures and tables are located in the folder 'Scatter_9Square'. The functional analysis and pathway visualization of the four groups are also performed by MAGeCKFlute, and the results are located in the folders named 'Enrichment_9Square' and 'Pathview_9Square', respectively.

## Reporting Summary

Further information on research design is available in the Nature Research Reporting Summary.

## Data availability

The source code of MAGeCKFlute (version 0.99.18) is freely available at https://bitbucket.org/liulab/mageckflute/ under the three-clause Berkeley Software Distribution (BSD) open-source license. Questions or comments can be submitted through the MAGeCK Google group: https://groups.google.com/d/forum/mageck. The datasets used in this paper are presented in http://cistrome.org/MAGeCKFlute/.

## References


1. Cong, L. et al. Multiplex genome engineering using CRISPR/Cas systems. *Science* **339**, 819–823 (2013).
2. Gilbert, L. A. et al. Genome-scale CRISPR-mediated control of gene repression and activation. *Cell* **159**, 647–661 (2014).
3. Konermann, S. et al. Genome-scale transcriptional activation by an engineered CRISPR-Cas9 complex. *Nature* **517**, 583–588 (2015).
4. Mali, P. et al. RNA-guided human genome engineering via Cas9. *Science* **339**, 823–826 (2013).
5. 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).
6. Koike-Yusa, H., Li, Y., Tan, E. P., Velasco-Herrera Mdel, C. & Yusa, K. Genome-wide recessive genetic screening in mammalian cells with a lentiviral CRISPR-guide RNA library. *Nat. Biotechnol.* **32**, 267–273 (2014).
7. Shalem, O. et al. Genome-scale CRISPR-Cas9 knockout screening in human cells. *Science* **343**, 84–87 (2014).
8. Hart, T. et al. High-resolution CRISPR screens reveal fitness genes and genotype-specific cancer liabilities. *Cell* **163**, 1515–1526 (2015).
9. Wang, T. et al. Identification and characterization of essential genes in the human genome. *Science* **350**, 1096–1101 (2015).
10. Chen, S. et al. Genome-wide CRISPR screen in a mouse model of tumor growth and metastasis. *Cell* **160**, 1246–1260 (2015).
11. Manguso, R. T. et al. In vivo CRISPR screening identifies Ptpn2 as a cancer immunotherapy target. *Nature* **547**, 413–418 (2017).
12. Burr, M. L. et al. CMTM6 maintains the expression of PD-L1 and regulates anti-tumour immunity. *Nature* **549**, 101–105 (2017).
13. Kurata, M. et al. Using genome-wide CRISPR library screening with library resistant DCK to find new sources of Ara-C drug resistance in AML. *Sci. Rep.* **6**, 36199 (2016).
14. Han, K. et al. Synergistic drug combinations for cancer identified in a CRISPR screen for pairwise genetic interactions. *Nat. Biotechnol.* **35**, 463–474 (2017).
15. Shi, J. et al. Discovery of cancer drug targets by CRISPR-Cas9 screening of protein domains. *Nat. Biotechnol.* **33**, 661–667 (2015).
16. Mootha, V. K. et al. PGC-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat. Genet.* **34**, 267–273 (2003).
17. Subramanian, A. et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA* **102**, 15545–15550 (2005).

18. Li, W. et al. MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. *Genome Biol.* **15**, 554 (2014).
19. Li, W. et al. Quality control, modeling, and visualization of CRISPR screens with MAGeCK-VISPR. *Genome Biol.* **16**, 281 (2015).
20. Toledo, C. M. et al. Genome-wide CRISPR-Cas9 screens reveal loss of redundancy between PKMYT1 and WEE1 in glioblastoma stem-like cells. *Cell Rep.* **13**, 2425–2439 (2015).
21. Kanehisa, M. & Goto, S. KEGG: Kyoto Encyclopedia of genes and genomes. *Nucleic Acids Res.* **28**, 27–30 (2000).
22. Luo, B. et al. Highly parallel identification of essential genes in cancer cells. *Proc. Natl. Acad. Sci. USA* **105**, 20380–20385 (2008).
23. Konig, R. et al. A probability-based approach for the analysis of large-scale RNAi screens. *Nat. Methods* **4**, 847–849 (2007).
24. Hart, T. & Moffat, J. BAGEL: a computational framework for identifying essential genes from pooled library screens. *Bioinformatics* **17**, 164 (2016).
25. Yu, J., Silva, J. & Califano, A. ScreenBEAM: a novel meta-analysis algorithm for functional genomics screens via Bayesian hierarchical modeling. *Bioinformatics* **32**, 260–267 (2016).
26. Morgens, D. W., Deans, R. M., Li, A. & Bassik, M. C. Systematic comparison of CRISPR-Cas9 and RNAi screens for essential genes. *Nat. Biotechnol.* **34**, 634–636 (2016).
27. Falcon, S. & Gentleman, R. Using GOstats to test gene lists for GO term association. *Bioinformatics* **23**, 257–258 (2007).
28. Yu, G., Lg, W., H., Y. & Qy., H. clusterProfiler: an R package for comparing biological themes among gene clusters. *OMICS* **16**, 284–287 (2012).
29. Shalem, O., Sanjana, N. E. & Zhang, F. High-throughput functional genomics using CRISPR-Cas9. *Nat. Rev. Genet.* **16**, 299–311 (2015).
30. Gini, C. "Concentration and dependency ratios" (in Italian). *Rev. Pol. Econ.* **87**, 769–789 (1997).
31. Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **8**, 118–127 (2007).
32. Chen, C. H. et al. Improved design and analysis of CRISPR knockout screens. *Bioinformatics* **34**, 4095–4101 (2018).
33. Jiang, P. et al. Network analysis of gene essentiality in functional genomics experiments. *Genome Biol.* **16**, 239 (2015).
34. DeKelver, R. C. et al. Functional genomics, proteomics, and regulatory DNA analysis in isogenic settings using zinc finger nuclease-driven transgenesis into a safe harbor locus in the human genome. *Genome Res.* **20**, 1133–1142 (2010).
35. Hockemeyer, D. et al. Efficient targeting of expressed and silent genes in human ESCs and iPSCs using zinc-finger nucleases. *Nat. Biotechnol.* **27**, 851–857 (2009).
36. Barretina, J. et al. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature* **483**, 603–607 (2012).
37. Aguirre, A. J. et al. Genomic copy number dictates a gene-independent cell response to CRISPR/Cas9 targeting. *Cancer Discov.* **6**, 914–929 (2016).
38. Sherr, C. J. & Roberts, J. M. CDK inhibitors: positive and negative regulators of G1-phase progression. *Genes Dev.* **13**, 1501–1512 (1999).
39. Huang, D. W., Sherman, B. T. & Lempicki, R. A. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat. Protoc.* **4**, 44–57 (2009).
40. Wang, T. et al. Gene essentiality profiling reveals gene networks and synthetic lethal interactions with oncogenic Ras. *Cell* **168**, 890–903 (2017).
41. Tzelepis, K. et al. A CRISPR dropout screen identifies genetic vulnerabilities and therapeutic targets in acute myeloid leukemia. *Cell Rep.* **17**, 1192–1205 (2016).
42. 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).
43. Chen, C.H., et al. Improved design and analysis of CRISPR knockout screens. *Bioinformatics* **34**, 4095-4101 (2018).
44. Leek, J. T., Johnson, W. E., Parker, H. S., Jaffe, A. E. & Storey, J. D. The sva package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics* **28**, 882–883 (2012).
45. Langmead, B., Trapnell, C., Pop, M. & Salzberg, S. L. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* **10**, R25 (2009).
46. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25**, 1754–1760 (2009).
47. Luo, W. & Brouwer, C. Pathview: an R/Bioconductor package for pathway-based data integration and visualization. *Bioinformatics* **29**, 1830–1831 (2013).


## Acknowledgements

This project was supported by the National Institutes of Health (R01 HG008927), the National Key Research and Development Program of China (2017YFC0908500 to X.S.L), the Breast Cancer Research Foundation, the Department of Defense (PC140817P1 to M.B. and X.S.L), and the start-up fund of the Center for Genetic Medicine Research and the Gilbert Family Neurofibromatosis Institute (to W.L.).

## Author contributions

W.L. and X.S.L. developed the original MAGeCK and MAGeCK-VISPR algorithm. B.W., M.W. and W.Z. developed the R package MAGeCKFlute. B.W. and W.Z. performed the data analysis; B.W., M.W., F.W., W.L. and X.S.L. wrote the manuscript with the help of Z.L., N.T. and X.W. W.L., X.S.L., B.W., M.W., W.Z., F.W., Z.L., N.T., X.W., T.X., C.-H.C., A.W., S.M., Y.C., S.S., J.J.L., M.H., J.Z. and M.B. contributed to the discussion and writing of the final manuscript.

## Competing interests

T.X. and X.S.L are co-founders and M.B. and X.S.L. are on the Scientific Advisory Board of GV20 Oncotherapy. The authors declare no competing interests.

## Additional information

**Supplementary information** is available for this paper at https://doi.org/10.1038/s41596-018-0113-7.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Correspondence and requests for materials** should be addressed to W.L. or X.S.L.

**Publisher's note:** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Published online: 1 February 2019

## Related links

**Key references using this protocol**

Li, W. et al. *Genome Biol.* **15**, 554 (2014): https://doi.org/10.1186/s13059-014-0554-4
Li, W. et al. *Genome Biol.* **16**, 281 (2015): https://doi.org/10.1186/s13059-015-0843-6
Jeselsohn, R. et al. *Cancer Cell* **33**, 173–186 (2018): https://doi.org/10.1016/j.ccell.2018.01.004
Xiao, T. et al. *Proc. Natl. Acad. Sci. USA* **115**, 7869–7878 (2018): https://doi.org/10.1073/pnas.1722617115

**Key data used in this protocol**

Toledo, C. M. et al. *Cell Rep.* **13**, 2425–2439 (2015): https://doi.org/10.1016/j.celrep.2015.11.021
Hart, T. et al. *Cell* **163**, 1515–1526 (2015): https://doi.org/10.1016/j.cell.2015.11.015
Shalem, O. et al. *Science* **343**, 84–87 (2014): https://doi.org/10.1126/science.1247005
Wang, T., Wei, J. J., Sabatini, D. M. & Lander, E. S. *Science* **343**, 80–84 (2014): https://doi.org/10.1126/science.1246981
Chen, C.-H. et al. *Bioinformatics* **34**, 4095–4101 (2018): https://doi.org/10.1093/bioinformatics/bty450

nature research

Corresponding author(s): Xiaole Shirley Liu

# Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

## Statistical parameters

When statistical analyses are reported, confirm that the following items are present in the relevant location (e.g. figure legend, table legend, main text, or Methods section).

| n/a | Confirmed | |
|---|---|---|
| ☐ | ☒ | The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement |
| ☐ | ☒ | An indication of whether measurements were taken from distinct samples or whether the same sample was measured repeatedly |
| ☐ | ☒ | The statistical test(s) used AND whether they are one- or two-sided<br>*Only common tests should be described solely by name; describe more complex techniques in the Methods section.* |
| ☐ | ☒ | A description of all covariates tested |
| ☐ | ☒ | A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons |
| ☐ | ☒ | A full description of the statistics including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals) |
| ☐ | ☒ | For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted<br>*Give P values as exact values whenever suitable.* |
| ☐ | ☒ | For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings |
| ☐ | ☒ | For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes |
| ☐ | ☒ | Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated |
| ☐ | ☒ | Clearly defined error bars<br>*State explicitly what error bars represent (e.g. SD, SE, CI)* |

*Our web collection on statistics for biologists may be useful.*

## Software and code

Policy information about availability of computer code

| | |
|---|---|
| Data collection | No software was used |
| Data analysis | MAGeCK v0.5.6 or newer (https://mageck.sourceforge.net/)<br>MAGeCK-VISPR v0.5.3 or newer (https://bitbucket.org/liulab/mageck-vispr)<br>Conda 4.5.4 or newer (https://conda.io/docs/)<br>Python v3.4.3 or newer (https://www.python.org)<br>R v3.5.0 or newer (https://www.r-project.org) or RStudio (https://www.rstudio.com/)<br>sva package v3.7 or newer (http://bioconductor.org/packages/release/bioc/html/sva.html) |

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers upon request. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

## Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

We selected two CRISPR screen data sets as example data sets, which are accessible at http://cistrome.org/MAGeCKFlute/. One is a CRISPR screen dataset generated from patient-derived Glioblastoma GBM stem-like cells (GSCs; GEO accession number: GSE70038)42 This is in FASTQ format. The second dataset is a CRISPR screen in a breast melanoma cancer cell line, A375, treated with the BRAF protein kinase inhibitor vemurafenib (PLX)7. The A375 dataset uses provide a raw read count for each format which reads previously mapped to the sgRNA library.

# Field-specific reporting

Please select the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/authors/policies/ReportingSummary-flat.pdf

# Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

| | |
|---|---|
| Sample size | No sample-size calculation was performed, we selected all the available samples to develop and test out pipeline. |
| Data exclusions | No data were excluded |
| Replication | All attempts at replication were successful |
| Randomization | We didn't do any randomization, instead of, we use all the available samples in our study. |
| Blinding | There is no experiment in our study, so blinding is not relevant. |

# Reporting for specific materials, systems and methods

### Materials & experimental systems

| n/a | Involved in the study |
|---|---|
| ☒ | ☐ Unique biological materials |
| ☒ | ☐ Antibodies |
| ☒ | ☐ Eukaryotic cell lines |
| ☒ | ☐ Palaeontology |
| ☒ | ☐ Animals and other organisms |
| ☒ | ☐ Human research participants |

### Methods

| n/a | Involved in the study |
|---|---|
| ☒ | ☐ ChIP-seq |
| ☒ | ☐ Flow cytometry |
| ☒ | ☐ MRI-based neuroimaging |