Integrative Analysis of Pooled CRISPR Genetic Screens Using MAGeCKFlute

Integrative analysis of pooled CRISPR genetic screens using MAGeCKFlute

Authors: Binbin Wang[1,2,*], Mei Wang[3,*], Wubing Zhang[1,2,*], Tengfei Xiao[7], Chen-Hao Chen[4,5], Alexander Wu[4,10], Feizhen Wu[11,12], Nicole Traugh[4,5], Xiaoqing Wang[4,5], Ziyi Li[1,2], Shenglin Mei[7], Yingbo Cui[9], Sailing Shi[1,2], Jesse Jonathan Lipp[8], Matthias Hinterndorfer[8], Johannes Zuber[8], Myles Brown[5,6], Wei Li[4,5,13,#], Xiaole Shirley Liu[1,2,4,5,#]

Affiliations:

1. Clinical Translational Research Center, Shanghai Pulmonary Hospital, Tongji University, Shanghai 200433, China,
2. Department of Bioinformatics, School of Life Sciences and Technology, Tongji University, Shanghai 200092, China.
3. Department of Geriatrics, Shanghai General Hospital, Shanghai 200080, China.
4. Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute and Harvard T.H. Chan School of Public Health, Boston, MA 02215, USA.
5. Center for Functional Cancer Epigenetics, Dana-Farber Cancer Institute, Boston, MA 02215, USA.
6. Department of Medical Oncology, Dana-Farber Cancer Institute and Harvard Medical School, Boston, MA 02215, USA.
7. GV20 Oncotherapy, Shanghai 201210, China.
8. Research Institute of Molecular Pathology (IMP), Vienna Biocenter (VBC), Vienna 1030, Austria.
9. College of Computer, National University of Defense Technology, Changsha 410073, China.
10. Program in Computational Biology and Quantitative Genetics, Harvard School of Public Health, Boston, MA 02215, USA.
11. Laboratory of Epigenetics, Institute of Biomedical Science, Fudan University, Shanghai 200032, China.
12. Children's Hospital of Fudan University, Shanghai, 201102, China.
13. Center for Genetic Medicine Research, Children's National Medical Center; Department of Genomics and Precision Medicine, George Washington University, Washington, DC 20010, USA.

#Corresponding Authors: Wei Li, Xiaole Shirley Liu

*These authors contributed equally to this work

1. **KEYWORDS** CRISPR; CRISPR screen; MAGeCK; MAGeCK-VISPR; CRISPR data analysis; CRISPR pipeline; Drug-treated CRISPR screen

2. **EDITORIAL SUMMARY MAGeCKFlute is an algorithm for analysis and visualisation of CRISPR screen data. Starting from sequencing reads and a sgRNA library, results are normalised and represented as pathway enrichment classifications.**

3. **TWEET Use MAGeCKFlute to analyse and visualise CRISPR screen results, from quality control and normalization to functional enrichment**

4. **COVER TEASER Analysis of CRISPR screens with MAGeCKFlute**

5. **Please indicate up to four primary research articles where the protocol has been used and/or developed.**
    1. Li et al. MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. Genome Biology 15 (554), 2014. https://doi.org/10.1186/s13059-014-0554-4 2.
    2. Li et al. Quality control, modeling, and visualization of CRISPR screens with MAGeCKVISPR. Genome Biology 16 (1) 281, 2015. https://doi.org/10.1186/s13059-015-0843-6 3.
    3. Jeselsohn et al. Allele-Specific Chromatin Recruitment and Therapeutic Vulnerabilities of ESR1 Activating Mutations. Cancer Cell 33 (2) 173-186. https://doi.org/10.1016/j.ccell.2018.01.004.
    4. Xiao et al. Estrogen-regulated feedback loop limits the efficacy of estrogen receptor–targeted breast cancer therapy. PNAS published ahead of print July 9, 2018 https://doi.org/10.1073/pnas.1722617115.

Abstract

Genome-wide screening using CRISPR coupled with nuclease Cas9 (CRISPR/Cas9) is a powerful technology for the systematic evaluation of gene function. CRISPR/Cas9 allows targeted modification or knockout of any genomic locus, which can simultaneously be tagged for evaluation of selection within a population under a desired growth condition. Data analysis begins with raw sequencing data and an sgRNA library and interpretation of the screening results is a critical step. Statistically principled analysis is needed for the accurate identification of gene hits and associated pathways. Here, we describe a protocol to perform computational analysis of CRISPR screens using the MAGeCKFlute pipeline. MAGeCKFlute combines our previously published MAGeCK and MAGeCK-VISPR algorithms, and incorporates additional downstream analysis functionalities. MAGeCKFlute is distinguished from other currently available tools by being a comprehensive pipeline that contains a series of functions for analyzing CRISPR screen data. This protocol explains how to use MAGeCKFlute to perform quality control, 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. Running the entire MAGeCKFlute pipeline requires approximately two hours on a desktop computer running Linux or Mac OS and with R support. The MAGeCKFlute package is available at http://www.bioconductor.org/packages/release/bioc/html/MAGeCKFlute.html.

Introduction

CRISPR(Clustered Regularly Interspaced Short Palindromic Repeats)/Cas9 is a powerful technology that can target desired genomic sites for gene editing or activity modulation via specific single-guide RNAs (sgRNAs)[1]. CRISPR screening is a high-throughput technology to investigate the functions of many genes in a single experiment. In the screening experiment, single-guide RNAs (sgRNAs) are designed, synthesized and cloned into a lentivirus library, which is subsequently transduced into cells at a low multiplicity of infection (MOI) to ensure each cell takes up only one sgRNA. A sgRNA usually contains 18-20 nucleotides 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 the host cell division.

Genome-wide CRISPR screens[2,3] allow for a systematic investigation of gene functions in various contexts[4]. The screening procedure can be categorized into knockout screens[5,6,7] and CRISPR activation/inhibition screens (CRISPRa/CRISPRi), which are performed by fusing a catalytically inactive Cas9 (dCas9) to transcriptional activation and 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 example to demonstrate data analysis. CRISPR screens have been highly effective at identifying genes that function in tumorigenesis[8,9], metastasis[10], response to immunotherapy[11,12], and genes associated with drug response[13,14,15].

To compare different cell populations, one population which represents the initial sgRNA status (day 0) is harvested, and the remaining populations are 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 vehicle/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 have 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 which 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 Experimental Design). MAGeCK-VISPR is a comprehensive quality control, analysis and visualization workflow for CRISPR/Cas9 screens. It incorporates MAGeCK and VISPR which together interactively explore results and quality control in a web-based frontend. In combination, MAGeCK and MAGeCK-VISPR allow users to perform read count mapping, normalization, quality control, and to identify positively and negatively select genes in the screens.

Here, we introduce MAGeCKFlute (Figure 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 allows the user to 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 demonstrate how to conduct normalization, analysis, bias correction, hit identification, and quality control of CRISPR screens using this pipeline. We also show how to perform downstream GO term and Kyoto Encyclopedia of Genes and Genomes (KEGG)[20] pathway enrichment analyses, and how to visualize the results. Finally, 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[21], RSA[22], BAGEL[23], ScreenBEAM[24], and casTLE[25] also perform parts of the CRISPR screen analysis pipeline. RIGER[21] and RSA[22] examine the rank distribution of all sgRNAs targeting a gene with regard to selection during the screen and calculate the statistical significance of at the individual sgRNA level and at the gene level. BAGEL is based on a Bayesian supervised learning method, ScreenBEAM uses Bayesian hierarchical modeling 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 workflow. The workflow is based on MAGeCK and MAGeCK-VISPR and includes read mapping, normalization, quality control, 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 identify screening hits and perform downstream functional analysis for various CRISPR screens, such as CRISPR knockout, CRISPR activation, and CRISPR inhibition screens. MAGeCKFlute provides a function with one command line to run the entire pipeline, and can also be used to run individual functions, such as batch-effect removal, copy-number bias correction, gene hit identification, and functional enrichment analysis. 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. When CRISPR screening samples were 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[26], clusterProfiler[27] and GESA[17], to perform functional enrichment analysis, and currently it is unclear which model is most appropriate for analyzing screening results. MAGeCKFlute provides all three options mentioned above for enrichment analysis, and users are encouraged to test different models. Another limitation involves the quality of the screens. Certain samples may lose more than 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 may reduce the screening period to get higher quality screening data. Despite these limitations, our protocol provides a convenient approach to perform 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, however, a raw-read-count file which contains the reads for each sgRNA is also allowed. 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[28] and normalizes sgRNA read counts to adjust for sequencing depth. The first method used to identify gene hits is MAGeCK RRA. MAGeCK RRA allows for a comparison of two experimental conditions. It can identify sgRNAs and corresponding genes that are significantly selected between the two conditions. MAGeCK RRA ranks sgRNAs based on 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 which includes at least 3 conditions: a day 0 condition, a control condition (treated with vehicle/DMSO), and a drug-treated condition. MAGeCK MLE also models the 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 after the target is perturbed, similar to the "log fold-change" measurement in differential expression analysis. MAGeCK-VISPR further incorporates all functions of MAGeCK and performs quality control and visualization of all results using VISPR, a web-based interactive framework. MAGeCK NEST[29] adds additional features to MAGeCK-VISPR to improve hit calling. First, MAGeCK NEST can improve results using the Network Essentiality Scoring Tool[30] (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 may consider using MAGeCK NEST to improve hit calling if there are many sgRNA outliers or if a high Gini[31] index (see below) is observed in the screen data.

Quality control and read count table generation Aligning reads to a known sgRNA library and evaluating screen quality (Figure 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 quality control statistics, including:

- The numbers of mapped reads (Fig. 2a)
- The percentage of reads mapped (Fig. 2a)
- The number of sgRNAs to which zero reads mapped (Fig. 2d)
- Gini index[31] (measures the evenness of sgRNA read counts) (Fig. 2c)
- The beta score correlation among samples (Fig. 2b)

A low percentage of mapped reads may indicate errors from 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[31] to measure the evenness of sgRNA read counts. The Gini index[31] is a common measure of income inequality in economics, and is used here to measure the evenness of sgRNA read counts. A high Gini index[31] suggests that the distribution of the sgRNA read count is distributed very 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.

Batch effect removal CRISPR screens that are performed or sequenced in multiple batches may harbor batch effects. Batch effects may arise from differences in reagents, sequencing platforms, or any other unintended variations in experimental conditions. One example is a public CRISPR screen in colon cancer which has strong batch effects[8], where samples are clustered by batches instead of by conditions (Figure 3a). After correcting for batch effect in the dataset at the sgRNA count level using the ComBat[32] function (incorporated into MAGeCKFlute), the biological replicates are properly clustered together (Figure 3b). This indicates that the batch effect has been removed.

Read count normalization with negative control sequences or non-essential 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 which target a completely inert genomic location in all of the starting cell populations, such that cell proliferation was not differentially affected under any of the experimental conditions. AAVS1 is a well-validated "safe harbor" 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[33,34]. sgRNAs targeting AAVS1 have similar behavior across samples (Figure 3c), suggesting 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[35].

Similar to sgRNAs targeting the AAVS1 locus, sgRNAs targeting non-essential genes can also be used to normalize read counts. Starting from 927 non-essential genes whose knock-down has no significant 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 to 100th percentile (Supplementary Figure 1a) in 98.3% (1019 out of 1036) Cancer Cell Line Encyclopedia (CCLE)[36] cell lines (Supplementary Figure 1b). 350 out of 937 non-essential genes passed this criteria (Supplementary Figure 1). The expression distributions of these 350 genes are consistent across hundreds of cancer cell lines (Supplementary File 1). This suggests that sgRNAs targeting these genes are appropriate controls for normalization of CRISPR screens, if AAVS1-targeting sgRNAs are not available (Figure 3b).

Beta score normalization with essential genes Cells exposed to different conditions (for example, with or without a drug) may have different proliferation rates. For example, CDK4/6 inhibitors affect the cell cycle and generally reduce cell proliferation[37]. Therefore, comparing cells that have a faster doubling time to more slowly proliferating cells may lead to biases in hit identification, since 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, since many drugs affect cell proliferation (Supplementary Figure 2a). To overcome this bias, we generated a list of 625 refined, high-confidence core essential genes (Supplementary Method) to normalize the beta score. MAGeCKFlute performs normalization of gene beta scores using these genes, assuming 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 based on the median beta score of this refined set of 625 essential genes (Supplementary Figure 2b-c).

Normalization with essential genes in genome-wide screens makes the beta scores comparable across samples (supplementary Figure 2a, b).

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 for cells with high copy number regions targeted[38]. When an amplified region contains a targeted non-essential gene, the observed essentiality scores often appear more negative than expected (Supplementary Figure 3a). Therefore, false positives are introduced in essential gene identification. There is an optional method (Step 8.A.iii) presented in this protocol for correcting this copy-number related bias if the corresponding copy number file is available (Supplementary File 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 essentiality scores for all affected genes (Supplementary Figure 3b). This function had been incorporated to MAGeCKFlute pipeline and can be applied when performing MAGeCK RRA or MLE.

Differential hit identification upon cell treatment After beta score normalization, 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 cut off can be specified in the FluteMLE function, with default at 1 standard deviation from the differential beta score mean. We adopted a "quantile matching" approach to robustly estimate σ which is the standard deviation of the beta score β. σ is chosen such that the (1-p) empirical quantile of the absolute values of β matches the (1-p/2) theoretical quantile of the prior normal distribution $N\ (0, \sigma^2)$. p is set as 0.32 for 1 standard deviation and 0.05 for 2 standard deviation, which corresponds to 68% and 95% of the data falling within 1 and 2 standard deviations away of the mean, respectively. If we write the theoretical upper quantile of a normal distribution as $\mathrm{Q_N}(1-\mathrm{p})$ and the empirical upper quantile of β as $\mathrm{Q_{|\beta|}}(1-p)$, then σ 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[26,39] 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].

To explore the biological functions of screen hits, MAGeCKFlute incorporates several functional modules. We incorporated several published enrichment algorithms derived from the clusterProfiler[27], GOstats[26] and GSEA packages into the enrichment analysis. GO term and KEGG pathway annotation are also incorporated into the package. We also developed a new enrichment function, *enrich.HGT*, which tests the enrichment of KEGG pathways based on the hypergeometric distribution. The pathview package[42] is used to intuitively visualize the beta scores of genes on KEGG pathways (Figure 4). The results provided by these integrated tools allow users to gain insights into the functions of screen hits, and in drug screens may yield clues to the drug mechanism of action.

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
mageck mle	MAGeCK MLE
VISPR	Visulasation of the result of MAGeCK.
mageck-vispr	Quality control at the FASTQ and raw count level. And include all the functions of MAGeCK and VISPR
BatchRemove	Remove batch effect of CRISPR screen data at the raw read count level
mageck test/mle --cnv-norm parameter	Correct the bias caused by copy number when identify the hits using MAGeCK RRA and MAGeCK MLE.
mageck_nest.py	Improve hit identification and remove outlier sgRNAs
FluteRRA	Downstream analysis of the MAGeCK RRA results
FluteMLE	Quality control at the beta score level; normalization with essential genes; identify treatment related hits; and functional analysis

Materials

EQUIPMENT

Software

- MAGeCK v0.5.6 or newer (https://mageck.sourceforge.net/)
- MAGeCK-VISPR v0.5.3 or newer (https://bitbucket.org/liulab/mageck-vispr)
- Conda 4.5.4 or newer (https://conda.io/docs/)
- Python v3.4.3 or newer (https://www.python.org)
- R v3.5.0 or newer (https://www.r-project.org) or RStudio (https://www.rstudio.com/)
- sva package v3.7 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 preferred)

Data

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)[43] 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 provides a raw read count for each sgRNA.

EQUIPMENT SETUP

Software setup

Most of the commands given in the protocol run in a typical Linux/Mac shell prompt, and all commands should be run in the same directory as the data files. The protocol also includes R scripts. Commands meant to be executed from the Linux/Mac shell (for example, bash or csh) 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

Install MAGeCKFlute

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

```
$R
```

2. Install MAGeCKFlute, from either Liu lab using option A, or Bioconductor using Option B:

A. Install MAGeCKFlute from 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')
```

3. Test whether the MAGeCKFlute package 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.

Download and install MAGeCK and MAGeCK-VISPR

4. MAGeCK 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) Afterwards, add a bioconda channelusing the following command:

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

CRITICAL STEP: This is essential as MAGeCK and MAGeCK-VISPR depend on it. It is important to add conda-forge and bioconda in the order above to ensure that bioconda is the highest priority. This allows setup to be run properly.

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

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

(iv) Activate the environment via:

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

B. Install MAGeCK with source code

(i) MAGeCK (version 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 python setup.py using the following command:

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

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

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

(iv) Invoke python setup.py using the following command:

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

Install MAGeCK NEST (Optional)

5. 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.

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

```
$cd mageck_nest-3.0
```

7. Finally, install MAGeCK NEST using the following command:

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

Process CRISPR screen data with MAGeCK or MAGeCK-VISPR.

8. CRISPR screen data can be processed step-by-step with MAGeCK (option A) or using MAGeCK-VISPR (option B). In a typical use case, CRISPR screen data is processed with MAGeCK (option A) step-by-step. If users prefer to analyze the data using a config file and want to 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 MATERIALS section.

A. Process CRISPR screen data step-by-step with MAGeCK. TIMING: 1.5 h.

(i) Download and unzip the test data 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 with the mageck count function, by firstly 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
```

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), in which the columns 1-3 are sgRNA names, sequences, and target genes respectively (e.g. Table 2). The library file is either in .txt or in .csv format.

Table 2 | Example of an sgRNA library file

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

Run the *mageck count* command with:

```
$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
```

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 --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.

Box 1 Additional *mageck count* parameters

Several additional key parameters that may be used to run MAGeCK in step 8A(ii) are as follows:

--trim-5	Length of trimming the 5' of the reads, default AUTO (MAGeCK will automatically determine the trimming length).
--sgrna-len	Length of the sgRNA. Default AUTO (MAGeCK will automatically determine the sgRNA length. Only use this if users turn on the "--unmapped-to-file" option.
--count-n	Count sgRNAs with Ns. By default, sgRNAs containing N will be discarded.
--unmapped-to-file	Save unmapped reads to file, with sgRNA lengths specified by --sgrna-len option.

(iii) (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 text editor and saved as .txt file, and each item in this file should be separated by a tab. In this file, columns 1-3 are sample names that correspond to a raw count file, batch covariate which could be numbers that represent batches, and another covariate besides 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 and scripts in the same folder. Alternatively, provide full paths for the files and scripts in the command.

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

In this command,

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

`-batchMat` Matrix or file path of batch table, which has at least three columns, including Samples matched colname of mat, Batch, and Covariates.

`-cov` Specify the covariates besides 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 principle component analysis before and after batch removal.

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

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

`-outdir` Output directory on disk.

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

(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 which can be generated by the *mageck count* command or other alignment tools such as bowtie or bwa.

```
$mageck test -k GSC_0131.count.txt -t day23_r1,day23_r2 -c day0_r1,day0_r2
-n GSC_0131_rra
```

In this command,

`-k` Count table (Table 4, Step 4Aii), provide a tab-separated count table. Each line in the table should include sgRNA name (1st column), target gene (2nd column) and read counts (3rd column) in each sample.

`-t` Sample label or sample index (0 as the first sample according to python standard) in the count table that are to be treated as treatment experiments, separated by comma (,). If sample labels are provided (rather than a sample index), the labels must match the labels in the first line of the count table.

`-c` Sample label or sample index in the count table that are to be treated as control experiments, separated by comma (,). 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/treatment, or any control/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.

Please use command *mageck test –h* to see additional parameters.

CRITICAL STEP: Note that rather than employing MAGeCK RRA in this step, it is possible instead to use MAGeCK MLE, as described in the next steps, Steps 8AVI-VII.

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

(vi)(Optional) if an experiment contains more than two conditions, for example a 3-condition design: day0, drug treatment, DMSO treatment, we recommend using

MAGeCK MLE or MAGeCK-NEST (Box 2) instead of MAGeCK RRA to employ the previous step. To do this, access the directory of raw counts with the following command:

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

path/to/demo_data/ should point to either the demo data or to the user's own raw count data.

(vii) 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` Provide a tab-separated count table. Each line in the table should include sgRNA name (1st column), target gene (2nd column) and the read count in each sample (3rd and subsequent columns).

`--design-matrix`
Provide a design matrix (Box 3), either as a file name or a quoted string of the design matrix. An example of design matrix is shown in Table 5. The row 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 known non-essential genes, 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 parameters, *--norm-method control* and *--control-sgrna* as shown above. If these negative control sgRNAs were not included in the library, sgRNAs targeting non-essential genes can be used as negative controls for normalization, as described in the parameter specifications above. We generated a non-essential gene list which includes 350 genes as described in "Experimental Design" section. Users can use the *--control-sgrna* parameter to provide sgRNAs corresponding to these genes to perform the normalization with these non-essential 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 prepared by the user and saved as a .txt file.

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

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

```
$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 to "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' (for example, day 0 or plasmid) 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 to the initial condition.

Table 5 | An example of a design matrix file

samples	Initial condition	Mock treatment	Drug treatment
plasmid	1	0	0
D7_R1	1	1	0
D7_R2	1	1	0
PLX7_R1	1	0	1
PLX7_R2	1	0	1

(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. Both methods require a tab-delimited file containing copy number values for each gene across the cell line(s) associated with the experiment (Supplementary File 2). The copy number file contains 2 columns: gene name and copy number. 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. In order 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 copy number bias correction with MAGeCK RRA, type the following:

```
$mageck test -k rawcount.txt -t HL60.final,KBM7.final -c
HL60.initial,KBM7.initial -n rra_cnv --cnv-norm cnv_data.txt –cell-line
HL60_HAEMATOPOIETIC_AND_LYMPHOID_TISSUE
```

To perform copy number bias correction with MAGeCK MLE, type the following:

```
$mageck mle --count-table rawcount.txt --design-matrix designmatrix.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, make sure to activate the corresponding conda environment before conducting any MAGeCK-VISPR related command:

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

(ii) Initialize a new workflow, as follows.

Chose a workflow directory and initialize the workflow with the fastq or fastq.gz files that contain the raw reads. If the raw reads 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.

```
$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 raw counts. To run MAGeCK-VISPR with raw counts, users should specify the raw count file in the “config.yaml” file instead of using the parameter –reads.

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

(iv) Configure the workflow using the following command.

```
$cd workflow
```

Next, specify the path to the library file and the normalization method by changing the “config.yaml” file. The sgRNAs used for normalization and batch information must be provided as well, if data were generated from different batches. CNV correction is recommended if CNV data is available. A short video (Supplementary File 3) describing how to edit the “config.yaml” file can be found at: https://www.youtube.com/watch?v=3maSxhy1JL0.

(v) To check whether the “config.yaml” files has been configured correctly, enter the filling command line in the terminal:

```
$snakemake –n
```

(vi) Execute the workflow.

The execution of the workflow must specify how many CPU cores should be assigned to this process. In general, 3-4 cores should be sufficient, but an increase in cores will reduce the running time. The workflow can be performed using the fowling command:

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

(vii) (Optional) Visualize results with VISPR

MAGeCK-VISPR also provides a web-based visualization framework (VISPR) for an interactive exploration of CRISPR screen quality control and analysis results. Once the workflow execution has finished, visualize the generated results and quality controls by issuing 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.
```

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

Downstream analysis pipeline.

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

```
$R
```

10. Load the MAGeCKFlute package into the R environment:

```
>library(MAGeCKFlute)
```

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

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

12. 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: 1h

(i) To perform functional analysis for MAGeCK RRA results, 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 of MAGeCK RRA, which has been performed at step 8.A.v or 8.B.vi.

B. Functional analysis of MAGeCK MLE results. TIMING: 1h

(i) To perform functional analysis of MAGeCK MLE results, enter the following command:

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

The file “mle.gene_summary.txt” is included in the output files of MAGeCK MLE which has been performed at step 8.A.v or 8.B.vi. The meanings of the parameters in this command are as follows (see Box 4 for further 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 corresponding to the parameter `-ctrlname` and `-treatname`.

`-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 output file name.

After the command line finishes running, users can check the current directory to verify the existence of the files listed in Table 6. A detailed description of the output files can be found in the Anticipated Results section.

CRITICAL STEP: 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.

Box 4 Additional *FluteMLE* parameters

Additional key parameters that may be used to run *FluteMLE* in step 11.B.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 pvalue cutoff of enrichment analysis.
-adjust	One of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none".
-enrich_kegg	One of "ORT"(Over-Representing Test), "GSEA"(Gene Set Enrichment Analysis), "DAVID", "GOstats", and "HGT"(HyperGemetric 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 build-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.

Anticipated results

Results of MAGeCK count (8.A.ii).

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 is 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[31] 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. The Gini index[31] is a common measure of income inequality in economics, and is used here to measure the evenness of sgRNA read counts. We have found that good quality screen data exhibits the following qualities:

- The number of mapped reads should be greater than 300 multiplied by the total sgRNA number.
- The mapped read percentage should be greater than 65%.
- The "zero count" number should be less than 1% of total sgRNAs.
- The Gini index should be less than 0.1 for plasmid or initial state samples, and less than 0.2 for negative selection samples.

Results of MAGeCK RRA (Step 8.A.V)

The main output of the MAGeCK RRA consists of the following files (Table 6):

Table 6 | 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

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 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.

Results of MAGeCK MLE and MAGeCK NEST (8.A.VI)

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 6). 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 '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. The 'p-value' is calculated by randomly permuting sgRNA labels. The 'fdr' is the false discovery rate calculated by the Benjamini-Hochberg Procedure. Similarly, the 'wald-p-value' (and 'wald-fdr') is the p-value (and false discovery rate) calculated by the Wald test to determine whether the corresponding 'beta score' significantly differs from zero in the MAGeCK MLE model, respectively. A detailed description of the output files from MAGeCK MLE can also be found in MAGeCK documentation using the following link**:** https://sourceforge.net/p/mageck/wiki/output/

Results of MAGeCK-VISPR (Step 8.B.VI)

MAGeCK-VISPR is a comprehensive quality control (QC), data analysis, and visualization method for CRISPR screens. Both MAGeCK RRA and MLE can be run in combination with MAGeCK-VISPR. All the results 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 quality control of the reads at the sequence level for each sample, which is generated by FASTQC. The "test" subfolder contains the main results of MAGeCK RRA or MLE, including "gene_summary.txt" file, which can be used in the functional analysis in step 12.

Results of MAGeCKFlute (Step 12)

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 12.A.I)

MAGeCKFlute processes MAGeCK RRA results ("gene_summary" file) with the function *FluteRRA*. *FluteRRA* will perform functional analysis with both positively and negatively selected genes. GO enrichment analysis uses the clusterProfiler[27] package as the default

enrichment method. The Hypergeometric test is used as the default method for KEGG enrichment analysis. *FluteRRA* will output both figures (Figure 3c,d,e) and tables of the significantly enriched terms. All the result files are listed in Table 7.

Table 7 | Results of FluteRRA

Files	Description
Prefix_Flute.rra_summary.pdf	The integration of the results
bp.neg.png	The enrichment analysis of biology process using negatively selected genes.
bp.pos.png	The enrichment analysis of biology process 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.

FluteMLE (Step 12.B.I) FluteMLE uses the "gene_summary" file as its input, which is the output of MAGeCK MLE. 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 8.

Table 8 | 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 andrank plot of the treatment and control samples
Enrichment_Treat-Ctrl	Functional enrichment analysis of the genes, of which the beta scores are significantly different between treatment and control samples
Pathview_Treat_Ctrl	The KEGG map of the enriched terms

Scatter_9Square	A 9 square scatter plot of the treatment and control beta scores
Enrichment_9Square	Functional enrichment analysis of the four separate groups of genes from the 9 square scatter plot, which relate to the experimental conditions
Pathview_9Square	The KEGG map of the enriched terms

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, which has been mentioned in the Data section. The treatment samples were compared to the control samples (vehicle or DMSO treated). The cells were harvested at two time-points, 7 days and 14 days after treatment.

MAGeCKFlute performs quality controls (QC) based on the beta score to ensure that the two conditions are comparable. The QC results consist of 3 levels: distribution of the beta score for different conditions (Supplementary Figure 4a), linear fitting of the beta scores of essential genes (Supplementary Figure 4b), and MA plot (Supplementary Figure 4c). The distribution of the beta scores is in the folder "Distribution_of_BetaScores". After the normalization, the beta score of most genes should be close to zero. Therefore, the mean of the distribution should also be close to zero. The MA plot can be used to visualize the differences between the beta scores generated in two samples, by transforming the data onto M (βT-βC) and A (βT+βC) scales, the βT and βC are the beta score of treatment control samples, respectively. The MA plot can be found in the folder named "MAplot". Since the beta scores for most genes will not change drastically in treatment condition compare to control condition, the M value for the majority of the genes in the MA plot should be close to zero. The folder "Linear_Fitting_of_BetaScores" contains figures showing the linear regression results of the beta scores for treatment and control samples. A summary table of the beta scores for each normalization is also generated in the "Scatter_Treat_Ctrl" folder.

After quality control, MAGeCKFlute will identify the differences between two treatment conditions (not including plasmid or Day 0) such as samples treated with or without a particular drug. MAGeCKFlute will generate scatterplots (Figure 4a) and ranking plots (Figure 4b) to the "Scatter_Treat_Ctrl" folder. For the scatterplots, the x-axis is the beta score of control samples (specified in the *FluteMLE* function with the parameter *–c)*, and the y-axis is the beta score of treatment samples (specified with the parameter *–t)*. The two diagonal

lines represent the cutoff of +/- 1 standard deviation of the difference between the treatment and control beta scores. The red (or blue) points are genes whose essentiality decreased (or increased) after treatment compared with control, respectively. The red and blue points are used to generate as two independent gene lists for further downstream analysis. The ranking plot showing the rank of the change in the beta scores can also be found in the "Scatter_Treat_Ctrl" folder. Similar to the scatterplots, the ranking plots use the same criteria to identify the beta score differences.

The functional annotation will be performed based on the two groups of genes selected in the scatterplot and ranking plots (Figure 4c and 4d). The enrichment method and cutoff can be specified in the *FluteMLE* function. By default, MAGeCKFlute uses the Hypergeometric test to perform enrichment analysis. MAGeCKFlute also integrates several other popular tools such as GSEA[17], DAVID[39], GOstats[26] and clusterProfiler[27]. Enrichment results 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 (Figure 4e). For each selected gene, the left (or right) half is colored based on beta score of treatment (or control) samples, respectively. This approach allows users to explore the variations of the beta scores within one pathway graph. For example, *STAM* is strongly negatively selected in DMSO treatment condition and weakly positively selected in PLX treatment condition (Figure 4e). Therefore, this gene lost its essentiality upon PLX treatment, suggesting that it acts in a pathway targeted by PLX.

To identify treatment-related hits accurately, MAGeCKFlute classifies genes into 4 groups (Supplementary Figure 4d), determined by differences in the beta scores between the treatment and control samples. 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 (that is, cells whose the 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 and strongly positively selected in treatment (that is, 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 and weakly selected in the treatment. 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 and strongly negatively selected in the treatment. 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.

Troubleshooting

Troubleshooting advice can be found in Table 9.

Table 9 | Troubleshooting table

Step	Problem	Probable reason	Solution
1	Package installation failed	R version is old or dependent package cannot be installed	Update R to 3.5.0 or newer. Try to start the R session from the root folder. The PATH variable can be set by typing into a terminal: For Linux $export PATH="/usr/bin:$PATH" For MacOS $export PATH="/usr/local/bin:$PATH"
2	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 a correct format	Follow the tutorial to generate a correct design matrix. https://bitbucket.org/liulab/mageck-vispr
3	FluteMLE crashed with an error due to cannot find samples: *"Error in FluteMLE(gene_summary = 'mle.gene_summary.txt', treatname = 'treatment.beta', : No sample found!"*	The index specified with the parameters "treatname" or "ctrlname" does not fit the column names of "gene_summary.txt" file.	Ignore the suffix ".beta" when specifying the control names and treatment name with the parameter "treatname" and "ctrlname"

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 data sets with more samples or deeper sequencing runs may take longer, and timing will vary across different computers.

Steps 1-7, install MAGeCK and MAGeCK-VISPR and MAGeCKFlute. ~ 30 min

Step 8(A)ii. Generate a count table with FASTQ file. TIMING: 15 mins. ~6 million reads per min.
Step 8(A)iii. Batch effect removal. TIMING: 5 mins.
Step 8(A)iv. Identify screen hits using MAGeCK RRA. TIMING: 8 mins.
Step 8(A)v. Identify screen hits using MAGeCK MLE. TIMING: 1h
Step 8(B)vi. Process CRISPR screen data with MAGeCK-VISPR. TIMING:1.h.
Step 12(A)i. Functional analysis of MAGeCK RRA results. TIMING:30 mins
Step 12(B)i. Functional analysis of MAGeCK MLE results. TIMING:1.h.

Acknowledgments
This project was supported by the NHGRI/NIH (R01HG008728), the Breast Cancer Research Foundation, Shanghai Pulmonary Hospital (to X.S.L), the Startup fund from 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. perform 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., X.W.. All authors contributed to the discussion and writing of the final manuscript.

Competing Financial Interests
The authors declare that they have no competing financial and non-financial interests.

Availability
The source code of MAGeCKFlute (version 0.99.18) is freely available at https://bitbucket.org/liulab/mageckflute/ under the 3-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/.

Figure Legends

Figure 1. A schematic representation of CRISPR/Cas9 screen analysis using MAGeCKFlute. Procedure step numbers described in the main text are shown to the left of the corresponding box. The FASTQ files or raw read count file (Table 4), a screen library file (Table 2), and a Design matrix (Table 5) are required as input for initial analysis though the MAGeCK algorithm. The following components are optional: count table batch correction (which requires an otherwise optional batch matrix file), CNV analysis and correction, and use of MAGeCK-VISPR for processing and visualization. FluteRRA and FluteMLE are used for downstream analyses, including pathway enrichment using GO and KEGG. Outputs include the beta score distribution and beta-score scatter plots.

Figure 2. Example Quality control assessment of CRISPR/Cas9 screen data. All four samples are got from a genome-wide CRISPR screen dataset generated from patient-derived Glioblastoma GBM stem-like cells (GSCs; GEO accession number: GSE70038)[43]. These samples contain two conditions: Day0, initial time point samples; Day23, after 23 days of culture. Replicate 1 and Replicate 2 are biological replicates. These results are generated by performing the FluteRRA/MLE function with the parameter *-q*. (a) Read counts and mapping percentages. The mapped read percentage should be greater than 65%.(b) pairwise sample correlations, (c) Gini index, which measures read depth evenness within samples. The Gini index should be less than 0.1 for plasmid or initial state samples, and less than 0.2 for negative selection samples. (d) Number of missed sgRNAs. The number of missed should be less than 1% of total sgRNAs.

Figure 3. Batch effect correction and normalization of read counts and beta scores from CRISPR/Cas9 screen data. The HCT116 data are from a genome-wide CRISPR screen using HCT116 colorectal carcinoma cells[8]. The sgRNAs from several time points were harvested and sequenced. Each time point contains more than 1 replicate (A,B,C), but the replicates were generated independently. The LNCap data (Supplementary File 4) are from a genome-wide CRISPR screen data that includes 2 cell lines, LNCap95 and LNCap abl. (a) Density plot of read counts from gRNAs corresponding to negative control genes *AAVS1*, *CCR5*, and *ROSA26* (top) and to non–essential genes (bottom). (b) Beta score distribution of samples before (left) and after (right) normalization using non-essential genes. The red dashed line represents a normal distribution with a mean of zero and the same standard deviation as the original beta score. Black dashed line indicates the mean of the simulated normal distribution. After a correct normalization, the mean of beta score should be close to zero. (c) Before and (d) after batch-effect correction using ComBat[44]

Figure 4. CRISPR/Cas9 screen analysis by MAGeCKFlute. (a) Scatterplot of treatment and control beta scores. The beta scores were normalized using the median of the beta scores of pan-essential genes (Supplementary Method). The two diagonal lines indicate +/-1 standard deviation of the difference between treatment and control beta scores. Red dots (Group A) are genes whose beta score increased after treatment. Blue dots (Group B) are genes whose beta score decreased after treatment. Gray dots are genes for which the beta

score did not change significantly between different conditions. (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. Colors are as in panel a. (c) The top 10 enriched KEGG pathways with positively (Red, Group A) and (d) negatively (Blue, Group B) selected genes. The p-value was calculated with the clusterProfile package that is based on the hypergeometric distribution. (e) A visualization of treatment and control beta scores over the JAK-STAT signaling pathway generated by the Pathview package[42]. The left and right of a gene-box represent control and treatment beta scores, respectively.

Supplementary Information

Supplementary Figure 1. Selection of non-essential genes for normalization of CRIPSR screen data. (a) Distribution of expression of all non-essential genes in CCLE cell lines. The red dashed line indicates the cutoff of our non-essential genes selection. Genes with expression levels below the cutoff were excluded from non-essential gene list. (b) 350 out of 937 non-essential genes had expression that ranked between the 5th and 100th percentile in 98.3% (1019 out of 1036) Cancer Cell Line Encyclopedia (CCLE)[36] cell lines.

Supplementary Figure 2. Normalization with essential genes. Beta score of essential genes (blue dots) and all genes except essential genes (red dots) before and after normalization with essential genes. The histograms on the top and right show the beta scores of treatment and control conditions, respectively. Before normalization (a), the beta score distribution of treatment and control conditions are not comparable. After normalization (b), these two distributions are more comparable (c) The formula for normalization of the beta score using essential genes where c is an empirical value is used to scale the normalized beta score. The value of c is 0.6 and was obtained from public screen data[8].

Supplementary Figure 3. Copy number bias correction in MAGeCK-MLE. Model of the relationship between β scores and gene copy numbers before (a) and after (b) copy number correction. Without the copy number bias correlation, the beta score shows a positive correlation with copy number. This bias can be correct using MAGeCKFlute.

Supplementary Figure 4. Output figures of MAGeCKFlute. All the data are from a genome-wide CRRSPR screen with an A375 cell line (Materials) **(**a) Beta score distribution of treatment samples (PLX7_R1, PLX7_R2) and control samples (DMSO_R1, DMSO_R2). Distribution of beta score in both treatment and control conditions are consistent, making beta scores comparable within different conditions. (b) Scatterplot of beta score of two samples. The regression line (dashed line) indicates the consistency of beta score between two conditions. (c) The MA plot can be used to visualize the differences between beta score taken in two samples, by transforming the data onto M (log ratio) and A (mean average) scales., in which M= β_T-β_C, A=β_T+β_C, β_T is the beta score of treatment samples, β_C is the beta score of control samples. Blue line is M=0 and red line is the loess regression line. (d) Identification of treatment related genes. The horizontal and vertical dashed lines indicate the mean plus or minus one stand deviation of treatment and control beta score, respectively. The diagonal dashed line indicates mean plus or minus one standard deviation of the differential beta score which can be calculated using treatment beta score minus control beta score. The number in each group are the gene numbers. Top 5 genes are selected based on the beta score and labeled in each group. Genes in the green group are strongly negatively selected in the

control samples and are weakly positively or negatively selected in the treatment samples. These genes are potentially located in the pathways targeted by the treatment. The orange group contains genes that are weakly selected in the control and strongly positively selected in treatment. These are genes whose loss confers treatment resistance. Genes in the blue group are strongly positively selected in the control and weakly selected in the treatment. These genes may be either potential regulators of cell proliferation in general, or regulators of the treatment target. Genes in the purple group are weakly selected in the control and strongly negatively selected in the treatment. These genes are potentially synthetically lethal in combination with the drug treatment.

Supplementary File 1
The non-essential gene list.

Supplementary File 2
Copy number file used to perform the copy number correction.

Supplementary File 3
A video tutorial of how to edit the “config.yaml” file.

Supplementary File 4
The LNCap data which includes AAVS1, CCR5 and ROSA26 as negative control genes.

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 Jj Fau - Sabatini, D.M., Sabatini Dm Fau - Lander, E.S. & Lander, E.S. Genetic screens in human cells using the CRISPR-Cas9 system. **343**, 80-84.
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-1alpha-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 U S A* **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. Kanehisa, M. & Goto, S. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res* **28**, 27-30 (2000).
21. Luo, B.*, et al.* Highly parallel identification of essential genes in cancer cells. *Proc Natl Acad Sci U S A* **105(51)**, 20380-20385 (2008).
22. Konig, R.*, et al.* A probability-based approach for the analysis of large-scale RNAi screens. *Nat Methods* **4**, 847-849 (2007).
23. Hart, T. & Moffat, J. BAGEL: a computational framework for identifying essential genes from pooled library screens. *Bioinformatics* **17**, 164 (2016).

24. 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).
25. Morgens Dw Fau - Morgens, D.W., Deans Rm Fau - Deans, R.M., Li A Fau - Li, A. & Bassik Mc Fau - Bassik, M.C. Systematic comparison of CRISPR-Cas9 and RNAi screens for essential genes. *Nat Biotechnol* **34**, 634-636 (2016).
26. Falcon, S. & Gentleman, R. Using GOstats to test gene lists for GO term association. *Bioinformatics* **23**, 257-258 (2007).
27. Yu, G., Lg, W., Y., H. & Qy., H. clusterProfiler: an R Package for Comparing Biological Themes Among Gene Clusters. *OMICS* **16**, 284-287 (2012).
28. Shalem O Fau - Shalem, O., Sanjana Ne Fau - Sanjana, N.E. & Zhang F Fau - Zhang, F. High-throughput functional genomics using CRISPR-Cas9. *Nat Rev Genet* **16**, 299-311 (2015).
29. Chen, C.H.*, et al.* Improved design and analysis of CRISPR knockout screens. *Bioinformatics* **bty**, 450 (2018).
30. Jiang, P.*, et al.* Network analysis of gene essentiality in functional genomics experiments. *Genome Biol* **16**(2015).
31. Gini. "Concentration and dependency ratios" (in Italian). English translation in Rivista di Politica Economica. **87**, 769–789 (1997).
32. Johnson, W.E., Li C Fau - Rabinovic, A. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **8**, 118-127 (2007).
33. 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).
34. 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).
35. Morgens, D.W.*, et al.* Genome-scale measurement of off-target activity using Cas9 toxicity in high-throughput screens. *Nat Commun* **8**, 15178 (2017).
36. Barretina, J.*, et al.* The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature* **483**, 603-607 (2012).
37. Sherr, C.J. & Roberts, J.M. CDK inhibitors: positive and negative regulators of G1-phase progression. *Genes Dev* **13**, 1501-1512 (1999).
38. Aguirre, A.J.*, et al.* Genomic Copy Number Dictates a Gene-Independent Cell Response to CRISPR/Cas9 Targeting. *Cancer Discov* **6**, 914-929 (2016).
39. Huang da, W., Sherman Bt Fau - Lempicki, R.A. & 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. Luo, W. & Brouwer, C. Pathview: an R/Bioconductor package for pathway-based data integration and visualization. *Bioinformatics* **29**, 1830-1831 (2013).
43. 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).
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).