CRISPRcleanR^{WebApp}: an interactive web application for processing genome-wide pooled CRISPR-Cas9 viability screen

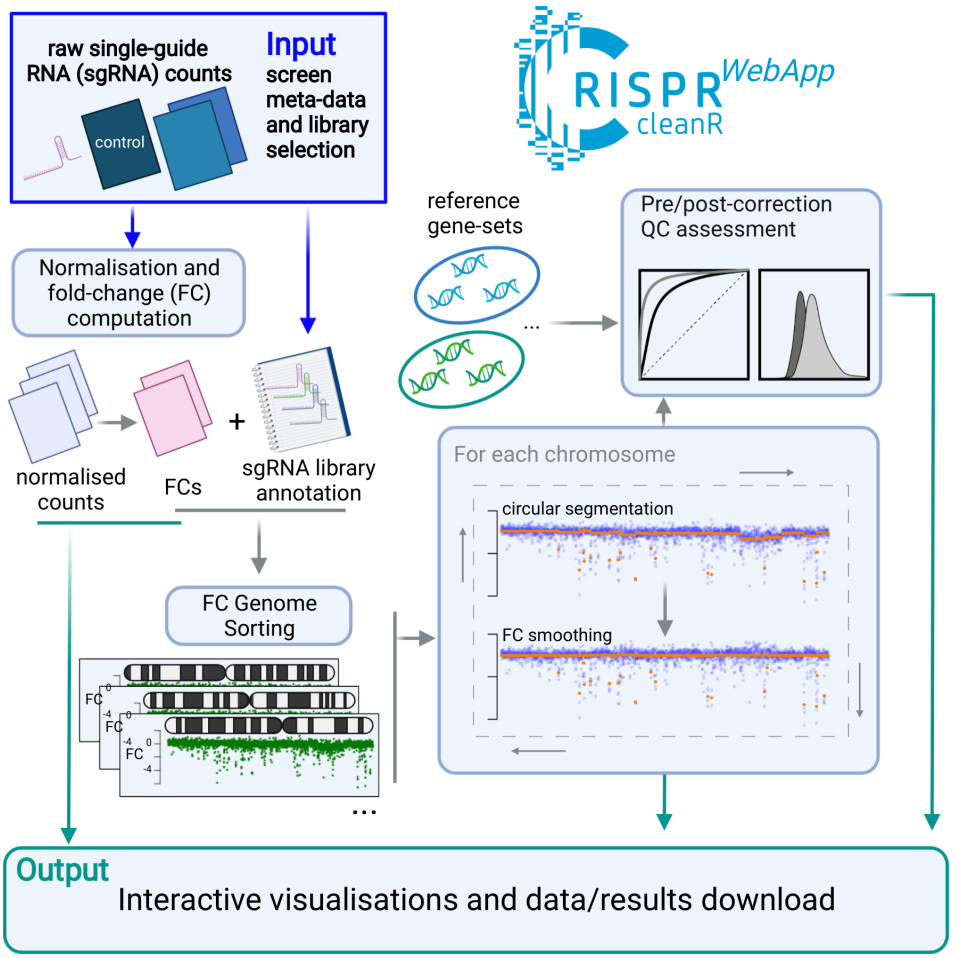
Alessandro Vinceti[1,*], Riccardo Roberto de Lucia[1,*], Paolo Cremaschi[1], Umberto Perron[1], Emre Karacok[2], Luca Mauri[1], Carlos Fernandez[1], Krzysztof Henryk Kluczynski[1], Daniel Stephen Anderson[1], Francesco Iorio[1,2,+]

[1] Human Technopole, Viale Rita Levi-Montalcini, 1 - 20157, Milano, Italy

[2] Wellcome Sanger Institute, Wellcome Genome Campus, Hinxton, Cambridge, CB10 1SA, UK

* Equally contributing authors

+Correspondence: francesco.iorio@fht.org



Summary

A major limitation of pooled CRISPR-Cas9 viability screens is the high false-positive rate in detecting *essential genes* arising from copy number-amplified (CNA) regions of the genome. To solve this issue, we developed *CRISPRcleanR*: a widely-used computational method implemented as R/python package and in a dockerized version. CRISPRcleanR detects and corrects biased gene-independent responses to CRISPR-Cas9 targeting in an unsupervised fashion, and it accurately reduces false-positive signals, while maintaining sensitivity in identifying relevant genetic dependencies. Here we present *CRISPRcleanR*^{WebApp}, a web-based application that enables access to CRISPRcleanR through an intuitive graphical web interface. CRISPRcleanR^{WebApp} removes the complexity of low-level R/python-language user interactions, including a range of interactively explorable plots. In addition, it supports a wider range of CRISPR guide RNAs' libraries with respect to the original package. CRISPRcleanR^{WebApp} is freely available and open to all users at: https://iorio-apps.fht.org/ccr.

Highlights

- CRISPR-Cas9 screens are widely used for the identification of cancer dependencies
- False-positives in CRISPR-Cas9 screens arise from targeting copy-number (CN) amplified genomic regions
- CRISPRcleanR corrects this bias in an unsupervised fashion
- CRISPRcleanR^{WebApp} is a web user-friendly front-end for CRISPRcleanR
- CRISPRcleanR^{WebApp} supports a wider range of CRISPR libraries and supports interactive results' explorations

Keywords

CRISPR-Cas9 screens, Copy number, Bias correction, Cancer dependency, Unsupervised analysis, Web-application

Introduction

The last decade has seen the advent of genome editing methods based on the Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR) system, which has revolutionised the way molecular biology is investigated and new therapeutic targets are discovered and prioritised (Behan et al. 2019b; Konstantinos Tzelepis et al. 2016; Mali et al. 2013; Cong et al. 2013). Particularly, in the last few years, the CRISPR-Cas9 system has been increasingly employed in functional genetic recessive screens, and iteratively optimised reaching unprecedented levels of efficacy, specificity and scalability, thus becoming the state-of-the-art tool for gene inactivation experiments (Smith et al. 2017; Barrangou et al. 2015; Evers et al. 2016). One of the main applications of this technology has consisted in probing each gene's potential in selectively reducing the viability of cancer cells upon inactivation (K. Tzelepis et al. 2016; T. Hart et al. 2015; T. Wang et al. 2014; Tim Wang et al. 2015), and large panels of immortalised tumour cell lines have been CRISPR-screened with the aim of identifying genomic-context-specific cancer vulnerabilities that might be exploited therapeutically (Behan et al. 2019a; Tsherniak et al. 2017; Dempster et al. 2019; Kurata et al. 2018; Martinez-Lage et al. 2018). In such experimental efforts, the screened models are engineered to achieve a constitutive or transient expression of the Cas9 endonuclease, then are transfected with a library of pooled single-guide RNAs (sgRNAs) targeting individual genes, at a genome scale.

A typical library contains multiple sgRNAs (usually from 2 to 10 (Shalem et al. 2014; T. Wang et al. 2014; Sanjana, Shalem, and Zhang 2014; T. Hart et al. 2015; K. Tzelepis et al. 2016; J. G. Doench et al. 2016; Traver Hart et al. 2017; Sanson et al. 2018)) targeting the same gene in different regions in order to increase inactivation efficiency and reduce possible topological biases (Haeussler et al. 2016). In addition, each sgRNA is co-delivered together with a resistance gene to a drug-selectable marker, allowing selection of successfully transfected cells, and a tag sequence allowing counting the number of cells to which an individual sgRNA has been successfully delivered via next generation sequencing.

After post-selection, expansion and genome sequencing of the transfected pool of cells, gene essentiality is usually quantified by differential representation analysis of the targeting sgRNAs. The counts derived from the DNA harvested at the end of the assay are contrasted against a control - for example, the counts derived from the transfected plasmid DNA (Imkeller et al. 2020; John G. Doench 2018) - and quantified through a depletion fold-change (FC).

The efficiency of the CRISPR-Cas9 system stems from its mode-of-action: the induction of DNA double strand breaks (DSBs) inflicted by the Cas9 enzyme on the genomic region

matched by a given sgRNA (Hsu, Lander, and Zhang 2014). DSBs are repaired by non-homologous end joining: an error prone mechanism causing small insertions and deletions, resulting in premature stop codons, thus efficient gene silencing (Gomez and Hergovich 2016; Symington and Gautier 2011; Panier and Durocher 2013). One of the major downsides of this system is that when used to target genomic copy number (CN) amplified regions, the Cas9 enzyme may cause a large number of DSBs, resulting in a highly cytotoxic effect that is independent from the targeted gene's function or its expression, and leads to false positive essential gene calls (Meyers et al. 2017; Iorio et al. 2018; Gonçalves et al. 2019; Dempster et al. 2021).

A number of computational methods have been developed by us and others to address this problem *in-silico*, from the analysis of sgRNA counts and FCs (Meyers et al. 2017; Iorio et al. 2018; de Weck et al. 2018; Gonçalves et al. 2019). We developed *CRISPRcleanR* (Iorio et al. 2018): the first tool working in an unsupervised way, i.e. not requiring in input any information on the copy-number-alteration profiles of the screened models, and not making any initial assumption on which are the DNA local topological properties to which the gene-independent responses to CRISPR-Cas9 targeting are due. CRISPRcleanR is implemented as R (https://github.com/francescojm/CRISPRcleanR) and Python package (https://github.com/cancerit/pyCRISPRcleanR) and it is usable in docker and cloud environments (https://dockstore.org/containers/quay.io/wtsicgp/dockstore-pycrisprcleanr).

The core algorithm of CRISPRcleanR operates in two steps: it first identifies biased genomic regions unsupervisedly, using a circular segmentation method (Venkatraman and Olshen 2007; Olshen et al. 2004), then it corrects the depletion FCs of all the sgRNAs whose targeted sequence falls in such regions. The effectiveness of CRISPRcleanR, combined with its simplicity and easy outcomes' interpretability have made it a widely used tool (Dede et al. 2020; Kim and Hart 2021; Lord, Quinn, and Ryan 2020; Allen et al. 2019; Gonçalves et al. 2020; Picco et al. 2019; Chai et al. 2020). However, the proficient use of CRISPRcleanR in its current implementations requires solid competences in computer programming, and this represents a limitation for the overall accessibility and effectiveness of this tool.

In order to allow the usage of CRISPRcleanR to a wider audience, we have developed *CRISPRcleanR*^{WebApp}: a web-based, user-friendly and interactive application that enables accessing CRISPRcleanR functionalities through an intuitive graphical user interface. This application provides a wrapper around the native R package avoiding all the low-level programming language interactions, while providing the same capabilities in terms of processing and data analysis as the original package, plus novel interactive data/result exploration modalities.

Finally, CRISPRcleanR^{WebApp} natively supports a larger set of CRISPR sgRNA libraries than the original package, encompassing six of the most popular and widespread ones (Meyers

et al. 2017; J. G. Doench et al. 2016; Ong et al. 2017; Sanjana, Shalem, and Zhang 2014; Aguirre et al. 2016; Gonçalves et al. 2021; K. Tzelepis et al. 2016; Tim Wang et al. 2015). Here we provide an overview of CRISPRcleanR^{WebApp} functionalities and data exploration modalities. Furthermore, we report results from a comparison of CRISPRcleanR^{WebApp} outcomes obtained by applying it to data from CRISPR-Cas9 screening the same cell line with the different supported libraries. The high level of concordance across these outcomes shows the effective compatibility of CRISPRcleanR^{WebApp} with the different libraries.

Design

Overview

CRISPRcleanR^{WebApp} is a client-server web app (**Fig. 1**) designed to use an underlying recent release of the CRISPRcleanR R package - v2.2.1, with v0.5.0 originally presented in (Iorio et al. 2018) - through a user-friendly interactive browser interface. CRISPRcleanR^{WebApp} provides a complete user experience offering the same analytic possibilities of the native version of CRISPRcleanR, in terms of workflows usage and interactions, while enriching it with novel intuitive and interactive data exploration and visualisation modalities. It consists of a web browser client Single Page Application (SPA) for user interaction, plus a backend server providing data storage and data processing. It also includes a user authentication/authorization mechanism, implemented through a login system, to protect submitted data and related results, ensuring a high level of privacy (STAR *Methods*). A set of video tutorials are included in the home page of CRISPRcleanR^{WebApp} and they guide the user through every step: from job submission, to results identification and access, up to results exploration and download. In addition, screen responsiveness is ensured, allowing a pleasant and effective user interaction on a wide variety of screen sizes, including tablets and smartphones. The application is served through a containerized architecture that takes full advantage of the Kubernetes cluster infrastructure available at Human Technopole (the hosting research institute). This makes CRISPRcleanR^{WebApp} highly stable, easy to maintain and scalable.

The core function of CRISPRcleanR^{WebApp} applies a circular binary segmentation algorithm (Venkatraman and Olshen 2007; Olshen et al. 2004) to patterns of sgRNAs' depletion fold-changes (FC) on a per chromosome basis. Particularly, it identifies genomic regions containing sgRNA clusters with sufficiently equal depletion FCs which are, on average, significantly different from those in the flanking genomic regions. Since it is very unlikely to observe the same fitness effect when targeting a large number of contiguous genes the FCs

in regions characterised by stretch of essential genes (3 in the default settings) are deemed as biased due to local features of the genomic segment (e.g. copy-number amplification), and they are corrected either via mean- or median-centering. On top of that, CRISPRcleanR^{WebApp} includes a suite of tools to measure, assess and visualise the effect of the performed correction, and to assess the performances of a depletion FC rank-based classification of priori known sets of essential/nonessential genes pre/post correction. CRISPRcleanR^{WebApp} also implements an inverse transformation function through which corrected sgRNA counts can be derived from corrected depletion FCs. This makes the output of CRISPRcleanR^{WebApp} compatible with other tools used to call depletion significance based on mean-variance modelling, like MAGeCK (Li et al. 2014), or bayesian statistics, like BAGEL (Traver Hart and Moffat 2016; Kim and Hart 2021).

Interface and workflow setup

CRISPRcleanR is equipped with two main analytical workflows: the first one pre-processes and corrects sgRNAs' read counts and depletion FCs, whereas the second one performs data quality control (QC) assessment and visualisation of results. The first workflow encompasses the complete set of CRISPRcleanR function calls, which are performed to obtain normalised sgRNAs' counts, gene-level depletion FCs and their corrected versions. Different setups can be customised to manage these functions: for instance, sgRNA raw counts can be normalised either by scaling sample-wise, based on the total number of reads, or by applying a median ratios normalisation; the user can upload a count file including reads from multiple control conditions and specify their number, or explicitly chose the minimal value of read counts a sgRNA should have in the control sample in order to be included in the pre-processing follow up analyses. The second workflow encompasses all the steps needed to assess the quality of a CRISPR-Cas9 screen and visualise related results. For instance, common quality checks are based on the computation of the Area Under Receiver Operating Characteristic and Precision-Recall curves (AUROC and AUPRC, respectively) when considering the computed profile of gene depletion FCs as a rank based classifier of two built-in sets of prior known essential and nonessential genes, as positive and negative control sets, respectively. All the results are summarised in data plots that can be queried through interactive visualisations or downloaded in different graphic formats.

These workflows are seamlessly integrated within CRISPRcleanR^{WebApp}. Indeed, the user, after secure login, is only asked to fill out the initial parameters in a job submission form in

the home page (**Fig. 2A** and **Table 1**), and upload a (properly formatted) delimiter-separated plain text file containing sgRNA counts (**Fig. 2B**) to run the analyses.

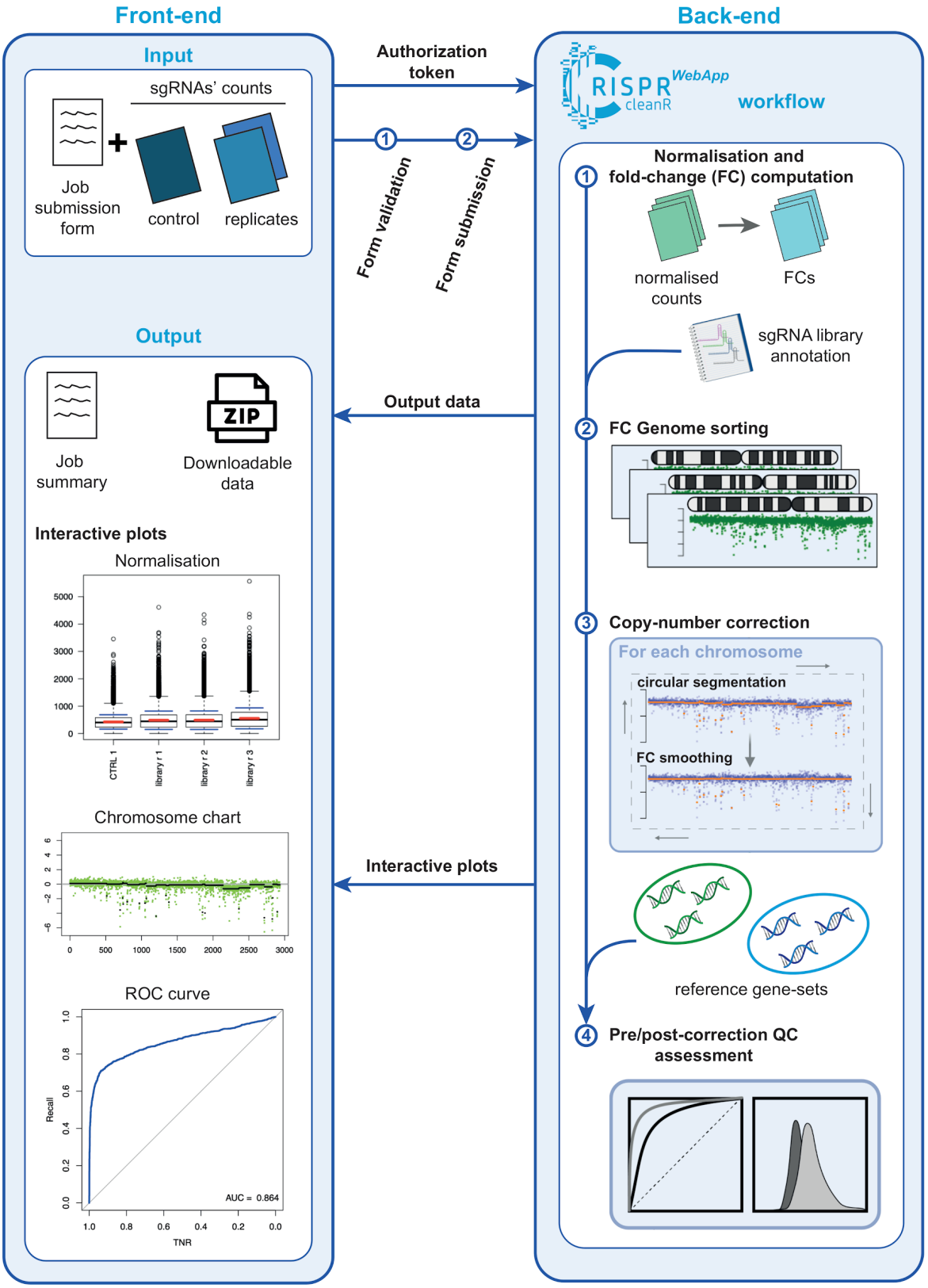


Fig. 1 - Overview of CRISPRcleanR^{WebApp} design. From the front-end, the user fills out a job submission form and uploads a plain text file containing single-guide RNA raw read counts derived from a genome-wide CRISPR-Cas9 screen, with one of the supported libraries. The form is then validated and submitted to the backend, where the data is processed following the CRISPRcleanR workflow. Results are then made available to the front-end and they are explorable through a set of interactive plots in a dedicated results page.

Field	Type	Description
Job title	Mandatory	Used to identify the output in the results section once the job is completed.
Email address	Optional	In case, the user receives the notification of the job outcome upon completion.
Data label	Mandatory	Specify a title for all the plots generated from the R package.
Library selector	Mandatory	Select between the supported CRISPR-Cas9 libraries (AVANA, Brunello, GeCKOv2, KY, MiniLibCas9 or Whitehead)
Minimal nº reads in the control sample	Mandatory (default is 30)	Minimal nº reads that each sgRNA should have in the plasmid count (or on average across multiple plasmid counts) in order to be included in the analysis.
Nº plasmids	Mandatory (default is 1)	Nº of plasmids used in the screen.
Normalisation procedure	Mandatory	Normalisation method for the read counts (counts per million or median ratios).
Notes	Optional	Additional notes for the job.

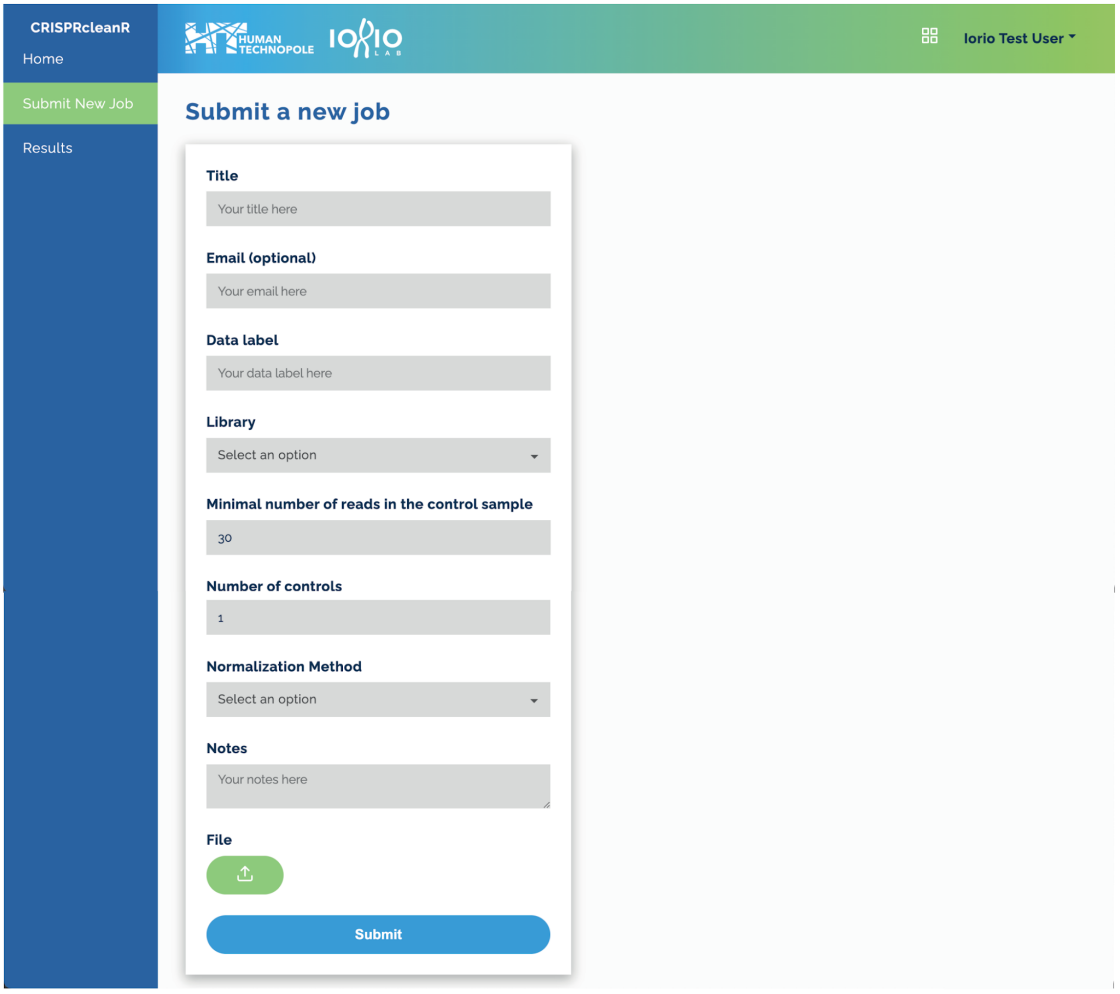
Table 1 - Fields to be filled in by the user in the job submission form.

The “Upload file” button allows the user to upload their input file containing the raw read counts. The accepted format is a comma- or tab-separated plain text with csv and tsv extensions. In the raw read counts file (**Fig. 2B**), the first two columns must include unique sgRNA identifiers and their targeted gene’s symbol, respectively, followed by one or more (in case of multiple controls) columns containing plasmid read counts. The remaining columns must contain replicates of the sgRNAs’ counts obtained post-selection and amplification, in the CRISPR screen. Before submission, the entire form is checked for possible wrong file formats and/or missing parameter specifications, in order to prevent incomplete or inconsistent input data. After the job submission, the user gets a notification assessing the outcome of the process (i.e. successful submission or submission error).

In the home page, users can download an example input file to test the app. This file contains sgRNAs’ read counts obtained by screening the HT-29 cell line with the Sanger KY library (K. Tzelepis et al. 2016) in a study by Behan et al (Behan et al. 2019a), and it

includes one control sample, i.e. read counts from the plasmid DNA, and three post-selection/amplification read counts' replicates (fourth, fifth and sixth columns).

A



B

sgRNA IDs	Gene symbols	Plasmid counts	Replicate counts		
A1BG_CCDS12976.1_ex3_19:58862927-58862950:-_5-1	A1BG	1122	1354	1568	1529
A1BG_CCDS12976.1_ex4_19:58863655-58863678:+_5-2	A1BG	580	639	500	833
A1BG_CCDS12976.1_ex4_19:58863697-58863720:-_5-3	A1BG	803	1235	900	1116
A1BG_CCDS12976.1_ex4_19:58863866-58863889:+_5-4	A1BG	424	353	283	394
A1BG_CCDS12976.1_ex5_19:58864367-58864390:-_5-5	A1BG	368	346	547	517

Fig. 2 - CRISPRcleanR^{WebApp} workflow setup. A. CRISPRcleanR^{WebApp} job submission form: the entry point for starting any new job request, after secure login. As illustrated, there are fields which the user is asked to fill out before submitting the job. **B.** Example of tab-separated file containing single-guide RNA raw read counts. The file is derived from the analysis of the HT-29 cell line screened with the Sanger KY library. First column contains sgRNA unique identifiers, the second one targeted gene symbols, then raw read counts of the plasmid DNA, followed by raw read counts obtained after 14 days post-transfection and selection of the library, in three replicates.

Results exploration and interactive plots

Once a job has been submitted, the CRISPRcleanR^{WebApp} back-end server starts an offline computation sending an email message to the user once finished, if an email address has been specified in the job submission form (this is not compulsory). A new results' entry is then immediately listed in the results page (**Fig. 3A**), where a data table offers a configurable pagination size for splitting large sets of entries into smaller chunks. In this table, each row refers to a job submitted by the logged user, and it shows main details such as submission date and time, email address, job title, and job status (succeed, failed or pending). Furthermore, the table allows for jobs filtering and sorting according to any of the column fields. Through this page the user can access and interactively explore the results obtained from each of their job submissions (**Fig. 3B**). The job results page shows a detailed summary that recapitulates the parameters specified by the user in the job submission form, and a series of panels allowing to download data and results, and to access all the interactive plots outputted by CRISPRcleanR^{WebApp}. In the Downloads panel, has access to links to all the results (and the original input file), which can be downloaded as zipped folders, as well as to all the plots, downloadable as static images in pdf or png format.
All the plots can be also visualised interactively and are equipped with a tooltip providing detailed information when hovering the mouse pointer on a graphic component. The job results page includes also a summary *gene-signature* plot, while the other plots are collected within expandable image portfolios, containing clickable thumbnails, and partitioned in three different panels (**Fig. 3B**): normalisation, chromosome charts and QC assessment. The gene-signature plot interactively shows results from the normalisation and the depletion FCs and count correction, i.e. the *chromosome plots*, all sections are supplied with a vertical/horizontal scrollbar: here, the user can create a grey rectangle of variable height/width and use it to zoom in a particular portion of the plot. On the left is provided a plot of all screened genes with coordinates indicating depletion FCs (x-axis) and depletion rank position (y-axis), respectively; on the right, the user can select one among 7 different signatures of control genes, i.e. prior known essential gene sets like proteasome, spliceosome, DNA replication, ribosomal proteins, RNA polymerase, and BAGEL essential and nonessential gene-sets (Traver Hart et al. 2017, 2014)), and explore how they rank based on their depletion FCs. Apart from the BAGEL gene-sets, the other 5 signatures are derived from the Molecular Signature Database (MSigDB) (Subramanian et al. 2005; Iorio et al. 2018). The red line indicates the rank position above which a false discovery rate (FDR) of non-essential genes is < 5%, and it is determined using the FC distributions of the BAGEL essential and nonessential genes (STAR *Methods*).

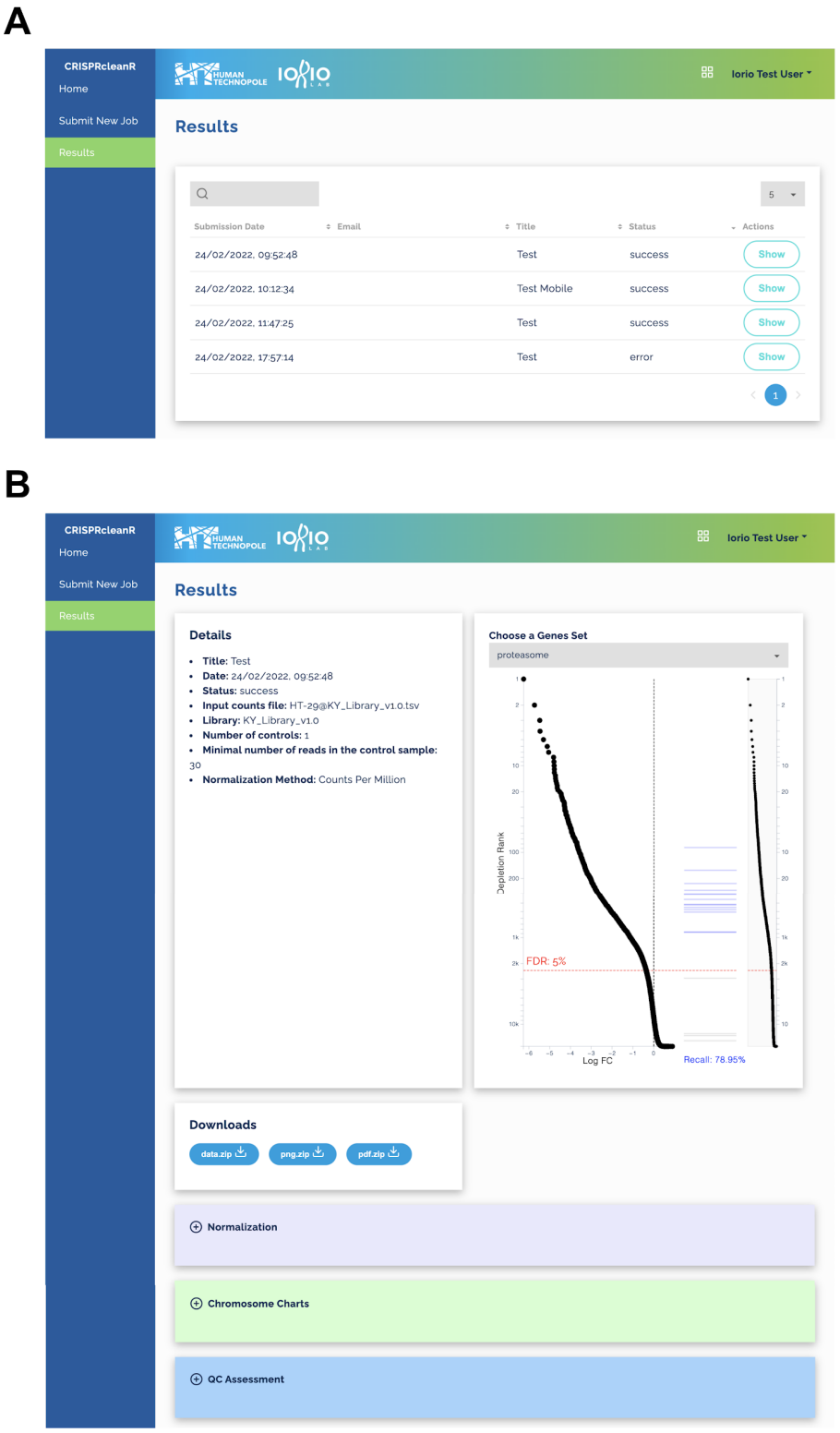


Fig. 3 - Results section of CRISPRcleanR^{WebApp}. A. The results list page: here the user will find all the results obtained from their submitted jobs. **B.** Job results accessible by selecting a job ID in the results list page. The page is made of several sections including a detailed description of the job, an interactive summary plot where all the screened genes are ranked based on their depletion fold-changes with overlaid signatures or control genes. Three expandable image portfolios contain clickable thumbnails for opening the related interactive plot rendered by the web application. Finally, the Downloads section includes links to all plots as static images (in pdf or png format), as well as to both input and processed data and results (as unique zipped folder of plain text files).

The “normalisation” panel contains two interactive plots: the first shows the comparison between raw and normalised sgRNA read counts (**Fig. 4A**), and the second shows the uncorrected FCs (**Fig. 4C**), across samples in the input file. The user can toggle between raw and normalised read counts by clicking the switch button in the upper left corner of the interactive plot. The tooltip provides several information on a particular point, i.e. guide ID, exon, gene, and raw or normalised read count. The same functionalities, with the exception of the switch button, are accessible in the interactive plot showing the uncorrected FCs (**Fig. 4C**).

The “QC assessment” panel contains a series of quality check plots obtained from the sgRNA/gene-level FCs corrected by CRISPRcleanR^{WebApp} (**Fig. 4BD**). They include visualisations of ROC and Precision/Recall curves (**Fig. 4BD**) computed on sgRNA- or gene-level corrected FC profiles, when they are considered as rank based classifiers of two built in reference sets of a priori known essential and nonessential genes. The tooltip provides information regarding the FC threshold at a certain level of recall (for the AUROC plot), or precision (for the AUPRC plot).

The “chromosome charts" panel shows one plot per chromosome, with segments of sgRNAs’ with sufficiently equal depletion FCs on the x-axis, and the corresponding average FC pre-/post-CRISPRcleanR-correction on the y-axis. The first 22 charts correspond to the autosomal chromosomes, the 23rd chart corresponds to the X chromosome, and the 24th chart to the Y chromosome. Each chromosome chart has two checkboxes, in the upper left corner, allowing the user to focus just on the segments (black lines) or the guide FCs (green dots) components. A switch button, in the upper right corner, allows to toggle between uncorrected and corrected FCs. Also in this case, the segment tooltip shows relevant information such as chromosome number, segment ID, start and end position of the segment location on the genome and average FC; on the other hand, the sgRNA tooltip provides information about chromosome number, targeted gene, start and end position of the gene portion targeted by the sgRNA, and depletion FC (uncorrected or corrected both).

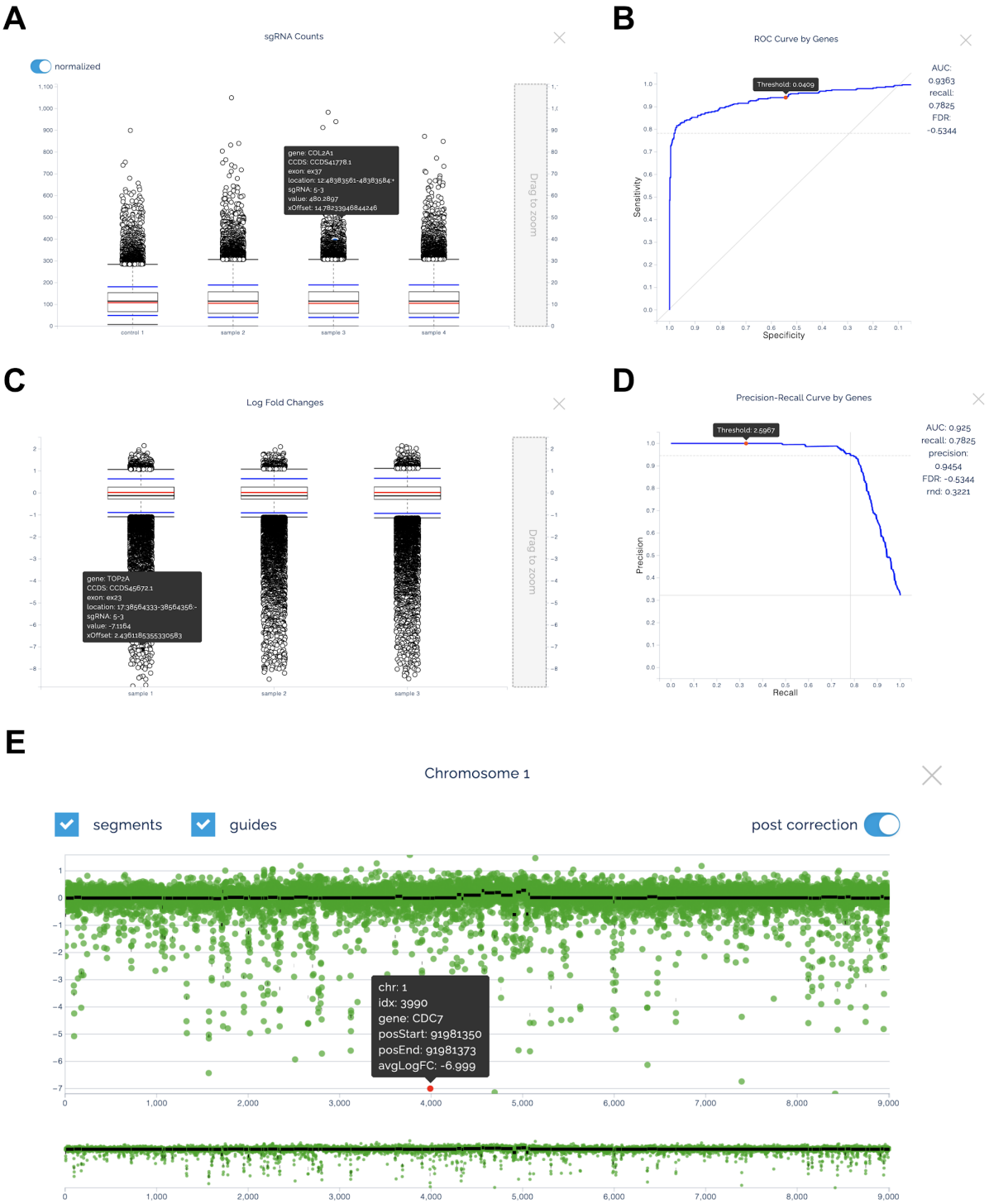


Fig. 4 - Interactive plots rendered by CRISPRcleanR^{WebApp}. AC. Overview of the normalisation plots. **A**. boxplots for raw and normalised single-guide RNA (sgRNA) read counts across samples in the input file. **C.** CRISPRcleanR uncorrected fold-changes (FCs). Both plots have a vertical scrollbar on the right that allows zooming in on specific portions of the plot. When overing on each point, a tooltip shows information regarding the corresponding sgRNA, i.e. guide ID, exon, gene, and read count/ depletion FC. **BD.** Quality control (QC) assessment plots. **BD.** Examples of interactive Receiver Operating Characteristic (ROC) and Precision Recall Curve (AUPRC) plots, obtained from profiles of corrected depletion FCs when considering them as rank based classifiers of two sets of a negative/positive control genes, i.e. priori known essential and nonessential genes. **E.** Interactive plot showing the sgRNA FCs (green dots) obtained targeting genes in the chromosome 1. The FCs are clustered within segments of equal copy-number (black lines), which are corrected by CRISPRcleanR. Each point is a sgRNA and when moving the mouse over it a tooltip will show related information such as chromosome number, guide ID, start and end position and depletion FC when hovering the pointer on it.

Results

Extended sgRNA libraries' support

The original version of CRISPRcleanR (v0.5.0) supported only the AVANA (J. G. Doench et al. 2016) and Sanger KY (K. Tzelepis et al. 2016; Behan et al. 2019a) CRISPR-Cas9 sgRNA libraries. CRISPRcleanR^{WebApp} builds upon and uses CRISPRcleanR v2.2.1, which we have extended from previous versions to fully support the following additional genome-wide libraries: Brunello (Sanson et al. 2018), GeCKOv2 (Shalem et al. 2014; Sanjana, Shalem, and Zhang 2014), Whitehead (T. Wang et al. 2014; Tim Wang et al. 2015; Park et al. 2017), and the recent MiniLibCas9 (Gonçalves et al. 2021) library of minimal size (with only two targeting sgRNA per gene). With the exception of GeCKOv2, the (publicly available) annotations of all these libraries include genomic coordinates of the targeted sequences for all the sgRNAs. This information is needed by CRISPRcleanR to genome-sort sgRNA depletion FCs prior correction. For the GeCKOv2 library, this information was not available. For this reason we remapped the GeCKOv2 sgRNA sequences onto the human genome (hg38) and assembled a CRISPRcleanR compatible annotation object (STAR *Methods*) for this library.

We tested the ability of CRISPRcleanR in supporting the extended set of libraries described above, in terms of correction performances and results' conservation across analyses of screens performed on the same cell line. Particularly, we tested CRISPRcleanR v2.2.1 on screens performed with all the aforementioned genome-wide CRISPR-Cas9 libraries on the HT-29 cell line. HT-29 is a human cancer cell line derived from colorectal carcinoma frequently used to assess sensitivity and specificity of CRISPR-Cas9 libraries and with multiple publicly available CRISPR screen datasets (Behan et al. 2019a; J. G. Doench et al. 2016).

First, we compared the extent of correction brought about by CRISPRcleanR on the different screens. To this aim we contrasted gene depletion FCs pre- versus post-CRISPRcleanR correction, and compared difference across libraries (**Fig. 5A**). Prior to this comparison we averaged sgRNA-level FCs on a targeted gene basis, in order to obtain gene-level FCs. As the tested screens presented different FC ranges and phenotype penetrance, due to inherent differences in the employed sgRNAs' sequences, for the sake of inter-screen comparability, we concatenated screen-wise the pre- and post-correction FC vectors and applied a min-max normalisation (STAR *Methods*) before comparing the different screens. After the normalisation, we split back each normalised vector in the two original components and computed differences between pre- and post-CRISPRcleanR -corrected FCs.

We observed an average median of these FC differences across library equal to -0.012 (min = -0.002 for KY and max = -0.02 for GeCKOv2) and an average interquartile range equal to 0.0245 (min = 0.015 for MiniLibCas9 and max = 0.037 for AVANA (**Fig. 5A**).
These results show that the correction applied by CRISPRcleanR has limited effects on the whole screen, it is focused on a small set of genes and that this minimal impact is conserved across screens of the same cell line with the different supported libraries.

Next, we asked at what extent the CRISPRcleanR correction affected each gene's FC consistently across screens. To this aim, we computed for each gene a normalised Shannon entropy, also known as efficiency, quantifying how homogeneous were the correction effects for a given gene across screens. A low entropy value indicated that a gene's FC was affected consistently across screens, whereas a high entropy value indicated the opposite, (STAR *Methods*). In Particular, we coded the gene-wise correction outcomes in a given screen as follows. A 0 indicated that the FC of the gene under consideration was not corrected by CRISPRcleanR, as the sgRNAs targeting that gene were not mapped onto a genomic segment detected as biassed. A 1 indicated a positive correction, meaning that the sgRNAs targeting the gene under consideration were mapped onto a genomic segment detected as biassed toward negative values by CRISPRcleanR and their FC increased. Following the same logic, a -1 indicated a negative correction. Applying this coding across all screens, yielded for each gene a vector of 6 entries (one per each tested library) with values in {-1, 0, 1}, from which we computed the normalised Shannon entropy.
A summary of the results is provided in **Fig. 5B**, showing the percentages of genes across efficiency values and concordance/discordance of correction effects for each value. For example, the first bar from the left (corresponding to a 0 efficiency) accounted for all the genes for which the correction outcomes were identical across screens, whereas the second bar accounts for the corrections that agree across all the screen but one, and so on.
For over 95% (**Fig. 5B**) of the genes we observed the same correction effect in at least four screens out of six (efficiency < 0.63), with the absolute majority (42%) showing different outcomes in a 4:2 proportion. Finally, only 4.31% of the genes presented all three different correction outcomes across screens, and 0.39% presented them equally partitioned (i.e. two -1s, two 0s, and two 1s). Overall these results show that the CRISPRcleanR correction affected individual genes' FCs homogeneously across screens of the same cell line performed with the different supported libraries.

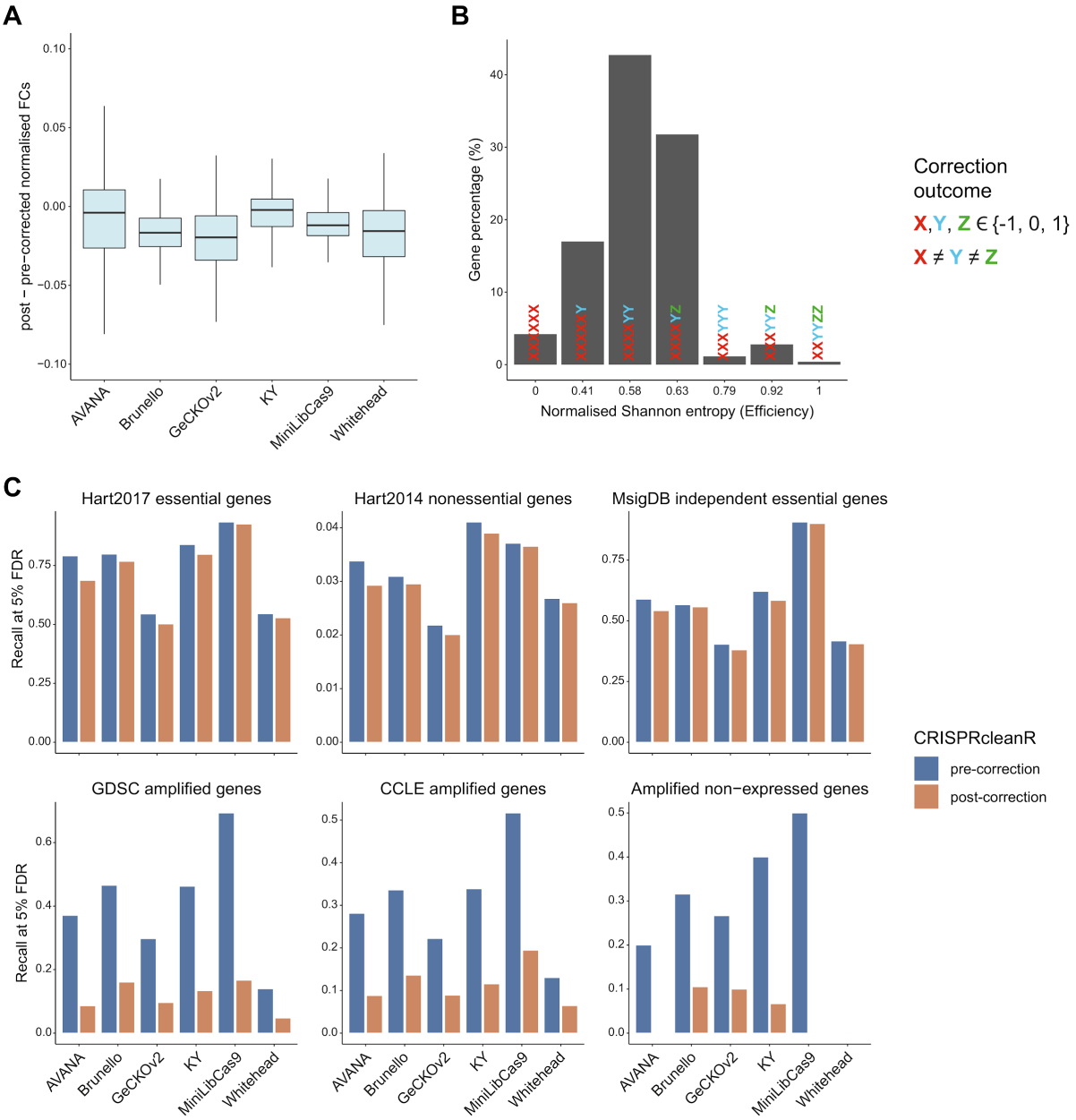


Fig. 5 - Assessment of CRISPRcleanR correction across different supported CRISPR-Cas9 libraries on the HT-29 cell line. A. Comparison of CRISPRcleanR pre- and post-correction fold-changes (FCs) across screens. **B.** Normalised Shannon entropy (efficiency) of gene-wise correction outcomes (-1, 0, 1) across screens. Only genes shared across libraries were considered. **C.** Recall at 5% FDR of six prior known gene-sets observed when considering pre- and post-correction (as indicated by the different colours) sgRNA FCs' profiles as rank based classifiers.

Finally, we measured the extent to which the CRISPRcleanR correction tendency in reducing false positive gene-essentiality calls while maintaining true positive calls was conserved across processed screens.

To this aim, we considered a six control sets of genes (**Table 2**): the Hart2017 essentials (Traver Hart et al. 2017), Hart2014 nonessentials (Traver Hart et al. 2014), respectively as positive and negative control plus another set of known essential gene-sets derived from the

MSigDB (Pacini et al. 2021; Subramanian et al. 2005) accounting for housekeeping cellular processes (DNA replication, histone genes, RNA polymerase, proteasome, ribosomal protein genes and spliceosome), and finally two sets of HT29 specific putative negative controls, i.e. copy number amplified genes (from two different public resources) (Iorio et al. 2016; Mermel et al. 2011; Barretina et al. 2012), as well as amplified and non-expressed genes (FPKM < 0.1 in HT-29, STAR *Methods*).

For each gene-set, we computed the recall at 5% FDR (Star *Methods*) for the pre- and post-corrected FCs across the six screens (**Fig. 5C**). For the negative controls (expected to be biased) we observed consistent reductions in recall across screens (median = 29.48%, 19.65% and 20.53%, respectively). In contrast, for the positive controls, we observed negligible recall reductions across screens (median = of 3.61%, 0.16% and 1.74% respectively). Thus, CRISPRcleanR can effectively correct the FCs of amplified gene-sets with high specificity without compromising the FCs of other genes, regardless of their phenotype intensity and employed supported library.

Set name	**Set Type**	**Number of genes**	**Dataset of origin**
Hart2017 essential	Reference set of core-fitness essential genes.	684	BAGEL reanalysis of 17 genome-scale knockout screens in human cell lines performed with different libraries (Traver Hart et al. 2017)
Hart2014 nonessential	Reference set of nonessential genes.	927	Large collection of shRNA gene dependency profiles analysed with a linear algebra approach (Traver Hart et al. 2014).
MSigDB independent essential	Pooled sets of independent essential genes from the MSigDB.	290	MSigDB dataset (Pacini et al. 2021; Subramanian et al. 2005)
GDSC amplified	Amplified genes in HT-29 derived from the GDSC dataset.	105	GDSC dataset (Iorio et al. 2016).
CCLE amplified	Amplified genes in HT-29 derived from CCLE dataset.	198	CCLE dataset (Mermel et al. 2011; Barretina et al. 2012).
Amplified non-expressed	Amplified non-expressed (FPKM < 0.1) genes in HT-29 derived from the GDSC dataset.	21	GDSC dataset (Iorio et al. 2016).

Table 2 - Sets of predefined genes for the assessment of CRISPRcleanR correction performances.

Discussion

We introduced CRISPRcleanR^{WebApp}, a web-based application integrating the complete suite of functionalities available in the CRISPRcleanR R/python package (Iorio et al. 2018). CRISPRcleanR is a computational tool for correcting gene-independent responses to CRISPR-Cas9 targeting that are observed in data from pooled viability screens and arise from copy number amplification. Differently from other methods, CRISPRcleanR works in an unsupervised way not requiring any input related to the copy-number variations profiles of the processed/screened model. Beside this, CRISPRcleanR carries out the correction on a single-sample basis, not requiring multiple screens to be analysed jointly nor borrowing signals across screens, thus, preserving the overall heterogeneity of the data. This makes CRISPRcleanR especially suited for the identification of context-specific dependencies and biomarkers.

With respect to the R/python implementations CRISPRcleanR^{WebApp} does not require any programming knowledge and offers a user-friendly interface giving full workflow control, and fully customisable execution of the CRISPRcleanR correction on user provided data.

The home-page of CRISPRcleanR^{WebApp} is equipped with comprehensive video-tutorials on its usage and an example dataset for testing it. A results page allows users to download input/output data, such as corrected FC file/s and static plots, as well as to explore all the results through a portfolio of interactive plots.

Furthermore CRISPRcleanR^{WebApp} supports a larger set of CRISPR libraries than the original version of CRISPRcleanR.

Here we have provided an overview of the CRISPRcleanR^{WebApp} implementation, design and interface and demonstrated that it yields consistent results across different technical settings and supported libraries.

Taken together, the features of CRISPRcleanR^{WebApp} provide an easy-to-use framework for pre-processing and correcting data derived from CRISPR-Cas9 screens, which might significantly widen the CRISPRcleanR user-community.

Limitations

While CRISPRcleanR^{WebApp} is a user-friendly web application accessible to non-computational scientists, its current version still lacks some functionality of the original package, which will be included in the next versions. In addition, the next releases of CRISPRcleanR^{WebApp} will include the capability of deriving read counts directly from sequencing files in FASTA and FASTQ formats. We are also planning to extend the number of input parameters that can be specified before submitting a job, as experienced users may

require more advanced setups. Currently, CRISPRcleanR^{WebApp} supports six of the most popular genome-wide CRISPR-Cas9 libraries, optimised to query gene essentiality in human immortalised cell lines. However, these constitute only a subset of all existing libraries and types of screens. Indeed, CRISPR-Cas9 screens can be carried out in primary cultures, organoids, or in vivo models (e.g. patient-derived xenografts), provided that they are performed using libraries with sufficient sgRNA density, and related annotations available. In future releases, it will be also possible to upload custom library annotation files provided that formatting requirements are satisfied. In addition, at the moment CRISPRcleanR^{WebApp} does not implement a statistical assessment of the correction on amplified genes and its impact on phenotype and possible distortions. Furthermore, it is not yet directly pipelined with existing tools that robustly estimate gene essentiality significance post-correction (like BAGEL (Traver Hart and Moffat 2016; Kim and Hart 2021) and MAGeCK (Li et al. 2014, 2015)). All these features are currently under development and will be included in future releases of CRISPRcleanR^{WebApp}.

Methods

Web-application architecture

Front-end

The front-end web application (also known as client) is implemented as a Single Page Application (SPA). Dynamic data is retrieved from a backend API server, which solely provides back data in JSON format, which is then used from the SPA to render HTML/CSS accordingly.

The client main page consists of a dashboard where a number of applications are listed as separated cards. For each application, a number of optional links (publications, available notebooks, etc..) can be accessed, and a dedicated button allows to access the related web app, if any. CRISPRcleanR^{WebApp} can be accessed from this page through the dedicated app card.

The client is based on Vue.js JavaScript framework (https://vuejs.org/). Interactive charts are implemented as scalable vector graphics documents managed through Vue components, also leveraging D3.js library (https://d3js.org/). Design and styling have been done entirely from scratch with Sass stylesheets, compiled as CSS during the application bundling process. Screen layout and responsiveness is done with Flexbox, CSS Grid layouts and Javascript. The bundled web application is served through a Dockerized (https://www.docker.com/) version of Nginx (https://www.nginx.com/) server on Kubernetes cluster (https://kubernetes.io/).

Back-end

The back-end consists of a docker multi-container app built upon the following containers and underlying technologies:

- Application Programming Interface (API) server: FastAPI python framework (https://fastapi.tiangolo.com/)
- Background queue: Celery Python (https://docs.celeryproject.org/en/stable/getting-started/introduction.html)
- Message Broker: Redis (https://redis.io/)
- Database: MongoDB (https://www.mongodb.com/)

Application Programming Interface. The API server is implemented with FastAPI, a modern and popular API Python framework. Pydantic Schemas (https://pydantic-docs.helpmanual.io/), already embedded within FastAPI, are used to

provide immediate data validation/sanitisation over expected fields from HTTP requests. In addition, FastAPI endpoints are almost automatically documented through Swagger (https://swagger.io/), making it easy to access the API documentation.
The API server is provided as a Docker container based on a python slim buster official docker image. Because of the containerised setup, load balancing and fault tolerance is provided from the Kubernetes cluster hosting the backend pods.

Task queue and Message Broker. To maintain the backend API server responsive while processing a job, we implemented an offline job processing mechanism. In particular, job processing is delegated to another container that receives jobs and processes them asynchronously with respect to the actual job submission, following a producer-consumer pattern. When the backend receives a new submitted job, this is forwarded to the background task queue through the message broker to inform consumers about new tasks to be executed.
Background processing is implemented through Celery, a common python task queue manager. Celery is executed on a dedicated Docker container based on a python slim buster official docker image, further customised to perform R processing. The customised image installs a Debian-compatible R distribution, along with all the needed dependencies required from CRISPRcleanR. The communication with CRISPRcleanR package is performed through Rpy2 (https://rpy2.github.io/), a python package that enables to register R functions and environments into python wrapping objects such that R interactions can be immediately executed from python code.
During job processing, the background queue container instantiates one single celery consumer, called worker, that consumes job computation requests from the message queue. In order to obtain parallel processing over multiple jobs, distinct independent workers can be spawned on container replicas by properly configuring the underlying kubernetes cluster. The message broker is then able to automatically send new job computations according to each worker's current workload, while still avoiding the same job execution to be performed on different workers. The communication between FastAPI backend and task queue is accomplished through a Redis-based message broker, on a dedicated container.

Database Container. The database container runs on a MongoDB instance, a cross-platform document-oriented NoSQL database program. The JSON-like structure of its documents allows easy interactions through Pydantic with no need for an object document mapper.
In addition, the backend container communicates with MongoDB using a Motor asyncio driver (https://motor.readthedocs.io/en/stable/), whereas the Celery-based background

queue uses a Pymongo driver (https://pymongo.readthedocs.io/en/stable/), since it is not based on asyncio patterns.

Data storage and security. CRISPRcleanR^{WebApp} implements an authorization schema based on OAuth2 protocol (https://datatracker.ietf.org/doc/html/rfc6749), following the Authorization Code Flow with public client. A further layer of authentication is provided through OpenID Connect (https://openid.net/connect/), which enables to bring user information, such as username, to the client application. This highly secure schema allows great flexibility in configuration and it is convenient to implement federated/third party access, which allows the use of external acknowledged identity providers to authenticate and authorise clients. For instance, user access through ORCiD is already supported.
The security schema is implemented through Keycloak (https://www.keycloak.org/), an integrated authorization/authentication system which provides a comprehensive solution for managing users and providing an authorization server for managing authentication and applications' tokens. Human Technopole infrastructure is equipped with a dedicated Keycloak system made accessible both from the backend (for token validation) and frontend (for token generation).
All the computed data as well as proceduced plots are stored on a S3 compliant bucket, hosted on-premises in the Human Technopole storage system, which is made accessible to the Kubernetes backend to exchange data.

Validation of the supported CRISPR-Cas9 libraries in the new version of CRISPRcleanR

Data acquisition

We validated CRISPRcleanR (v2.2.1) on six screens obtained from transducing popular CRISPR-Cas9 libraries into the HT-29 cell line. We tested the following libraries: AVANA (J. G. Doench et al. 2016), Brunello (Sanson et al. 2018), GeCKOv2 (Shalem et al. 2014; Sanjana, Shalem, and Zhang 2014), KY (K. Tzelepis et al. 2016; Behan et al. 2019a), MiniLibCas9 (Gonçalves et al. 2021) and Whitehead (T. Wang et al. 2014; Tim Wang et al. 2015; Park et al. 2017). All the raw read count files are available in the following GitHub repository: https://github.com/francescojm/CRISPRcleanR/tree/master/inst/extdata.

GeCKOv2 library mapping onto the Human genome

We mapped the protospacer sequence of each sgRNA in the GeCKOv2 library onto the human genome v38 using the short read mapping method bwa-mem. The reads that were mapped uniquely to the reference genome sequence and their positions on the reference genome are were. The remaining guides were mapped to the reference genome ambiguously and with some mismatches. We mapped these remaining unmapped guides to the reference human genome v38 using first blat and then with blastn. For the guides mapped onto multiple locations on the reference sequence we considered that in the targeted genes declared in the original library annotation. The annotations of the genes were extracted from Gencode v38. The final remaining sequences were mapped to the reference genome with mismatches and the positions with the minimum mismatch number within the targeted genes were selected. All the guides were mapped to a position on the reference genome within their targeted genes.

Comparison of CRISPRcleanR pre- and post-correction FCs across screens

To compare the differences in fold-changes (FCs) pre- and post-correction, we first normalised the sgRNAs' raw read counts by scaling for the total number of reads, after filtering out those guides having a plasmid read count less than 30. This is implemented in the *ccr.NormfoldChanges* function and resulted in uncorrected depletion FCs. Following the pipeline implemented in https://github.com/francescojm/CRISPRcleanR/blob/master/Quick_start.pdf, we applied the *ccr.GWclean* function, with default settings, to correct the FCs for each screen.
We obtained gene-level FCs by averaging the score of sgRNAs targeting the same gene. We considered only the genes in common across the six screens, leading to 8473 genes. In order to reduce technical variability due to the usage of different libraries, we applied a min-max normalisation on the concatenated vector of pre- and post-corrected gene-level FCs for each screen. We then split back the vector in pre- and post-corrected FCs, and computed gene-wise across screens.

Gene-wise correction agreement across screens

In information theory, Shannon entropy is used to quantify the uncertainty of a random variable. It is formally defined as the average auto information associated with each outcome of the random variable. Efficiency (or normalised Shannon Entropy) is defined as the Shannon entropy divided by the maximum value that the entropy can assume (which depends on the size of the spectrum of the random variable under consideration). In our case: the lower the efficiency the higher the homogeneity in the CRISPRcleanR outcomes

across screens. Thus a low efficiency score means a high agreement of correction outcomes across the screens. We computed gene-wise normalised Shannon entropy across screens when considering the three possible correction outcomes outputted by the *ccr.GWclean* function.

We considered only the genes in common across the screens, leading to 8473 genes, and removed those found at the conjunction of two segments, totalling to 7923 genes. Then, we computed the gene-wise normalised Shannon entropy H_n, following the formula (1):

$$(1) \qquad H_n = \frac{H_g}{H_{max}} = \frac{-\sum_{i=1}^{C} p(g_i) \cdot log_2(p(g_i))}{log_2(C)},$$

where H_g is the Shannon entropy of a gene g, computed as the sum over the variable's probability values, $g_i \in \{-1, 0, 1\}$ and C is the total number of possible correction outcomes (i.e. 3 in this case), divided by the maximum expected entropy (log_2 of C).

Recall of genes in six predefined gene-sets following CRISPRcleanR correction

We assessed the effect of CRISPRcleanR correction on six predefined sets of genes, namely Hart2017 essential, Hart2014 nonessential, independent sets of essential genes derived from the MSigDB, copy-number amplified genes from GDSC dataset, copy-number amplified genes from CCLE dataset, and copy-number amplified genes from GDSC that were not expressed (FPKM < 0.1 in HT-29).

For each screen, we computed the essentiality threshold at 5% false discovery rate (FDR) for the pre- and post-corrected screens, using the Hart2017 essential and Hart2014 nonessential as source of reference essential (EG) and nonessential genes (NEG), respectively. In particular, we ranked the gene-level depletion FCs of the EG and NEG in increasing order. For each rank position i, we calculated a set of predicted fitness genes (PFG) as follows:

$$(2) \qquad PFG(i) = \{g \in EG \cup NEG \,|rank(FC_g) \geq i\},$$

where $rank(FC_g)$ is the corresponding rank position of gene g in the reference gene set based on its depletion FC. The ranked list is then used to calculate positive predictive values (PPV) for the ith rank position as follows:

$$(3) \qquad PPV(i) = |PFG(i) \cap EG| \,/\, |PFG(i)|.$$

We determined the highest threshold of depletion FC (FC*) in rank position *i** such that PPV(i^*) ≥ 0.95, which is equivalent to an FDR of 5%. We considered all genes with a FC < FC* as essential for the viability of HT-29. For each gene-set *S*, we computed the recall by dividing the size of genes g_s below the 5% FDR threshold (FC*) by the total size of g_s included in the screen, according to (4):

$$(4) \qquad Recall = \frac{|g_s < FC^*|}{|g_s|}.$$

Availability

CRISPRcleanR^{WebApp} is available on the Iorio's lab webpage at: https://iorio-apps.fht.org/ccr/home. The latest version of the original CRISPRcleanR package can be found at the following GitHub repository: https://github.com/francescojm/CRISPRcleanR. The raw read count files obtained from six screens performed on the HT-29 cell line using different genome-wide CRISPR-Cas9 libraries are available as external data in the CRISPRcleanR package: https://github.com/francescojm/CRISPRcleanR/tree/master/inst/extdata.

Author Contributions

AV conceived the study, designed and performed the benchmark analyses, tested the web application, wrote and revised the manuscript. RRDL designed, developed and tested the web application, integrated all the necessary libraries and technologies with the IT infrastructure, wrote and revised the manuscript. RRDL is also responsible for the maintenance and future releases of the web application. PC customised the core package to facilitate its integration in the web application and added new functionalities, tested the web application, and revised the manuscript. UP contributed to the design of the benchmark analyses, tested the web application, and revised the manuscript. EK assembled a CRISPRcleanR compatible annotation for the GeCKOv2 library and revised the manuscript. LM, CF, KHK and DSA set up the IT infrastructure and technologies required for hosting the

web application at Human Technopole. FI conceived the study, designed the original algorithms and developed the R-based version of CRISPRcleanR, he contributed to the design of the benchmark analyses, wrote and revised the manuscript, supervised the study.

Competing interests

FI receives funding from Open Targets, a public-private initiative involving academia and industry and performs consultancy for the joint CRUK-AstraZeneca Functional Genomics Centre. All other authors declare that they have no competing interests.

Abbreviations

AUPRC = Area Under Precision-Recall Curve
AUROC = Area Under Receiver Operating Characteristic
CCLE = Cancer Cell Line Encyclopaedia
CN = copy number
CRISPR = Clustered Regularly Interspaced Short Palindromic Repeats
DSB = double strand breaks
FC = fold-change
FDR = false discovery rate
GDSC = Genomics of Drug Sensitivity in Cancer dataset
MSigDB = molecular signature database
QC = quality control
sgRNA = single-guide RNA
SPA = Single Page Application

References

Aguirre, Andrew J., Robin M. Meyers, Barbara A. Weir, Francisca Vazquez, Cheng-Zhong Zhang, Uri Ben-David, April Cook, et al. 2016. “Genomic Copy Number Dictates a Gene-Independent Cell Response to CRISPR/Cas9 Targeting.” *Cancer Discovery* 6 (8): 914–29.

Allen, F., F. Behan, A. Khodak, F. Iorio, K. Yusa, and M. Garnett. 2019. “JACKS: Joint Analysis of CRISPR/Cas9 Knockout Screens.” *Genome Research* 29. https://doi.org/10.1101/gr.238923.118.

Barrangou, Rodolphe, Amanda Birmingham, Stefan Wiemann, Roderick L. Beijersbergen, Veit Hornung, and Anja van Brabant Smith. 2015. “Advances in CRISPR-Cas9 Genome Engineering: Lessons Learned from RNA Interference.” *Nucleic Acids Research* 43 (7): 3407–19.

Barretina, Jordi, Giordano Caponigro, Nicolas Stransky, Kavitha Venkatesan, Adam A. Margolin, Sungjoon Kim, Christopher J. Wilson, et al. 2012. “The Cancer Cell Line Encyclopedia Enables Predictive Modelling of Anticancer Drug Sensitivity.” *Nature* 483 (7391): 603–7.

Behan, Fiona M., Francesco Iorio, Gabriele Picco, Emanuel Gonçalves, Charlotte M. Beaver, Giorgia Migliardi, Rita Santos, et al. 2019a. “Prioritization of Cancer Therapeutic Targets Using CRISPR-Cas9 Screens.” *Nature* 568 (7753): 511–16.

———. 2019b. “Prioritization of Cancer Therapeutic Targets Using CRISPR–Cas9 Screens.” *Nature* 568 (7753): 511–16.

Chai, Annie Wai Yeeng, Pei San Yee, Stacey Price, Shi Mun Yee, Hui Mei Lee, Vivian Kh Tiong, Emanuel Gonçalves, et al. 2020. “Genome-Wide CRISPR Screens of Oral Squamous Cell Carcinoma Reveal Fitness Genes in the Hippo Pathway.” *eLife* 9 (September). https://doi.org/10.7554/eLife.57761.

Cong, Le, F. Ann Ran, David Cox, Shuailiang Lin, Robert Barretto, Naomi Habib, Patrick D. Hsu, et al. 2013. “Multiplex Genome Engineering Using CRISPR/Cas Systems.” *Science* 339 (6121): 819–23.

Dede, Merve, Megan McLaughlin, Eiru Kim, and Traver Hart. 2020. “Multiplex enCas12a Screens Detect Functional Buffering among Paralogs Otherwise Masked in Monogenic Cas9 Knockout Screens.” *Genome Biology* 21 (1): 262.

Dempster, Joshua M., Isabella Boyle, Francisca Vazquez, David Root, Jesse S. Boehm, William C. Hahn, Aviad Tsherniak, and James M. McFarland. 2021. “Chronos: A CRISPR Cell Population Dynamics Model.” *bioRxiv*. https://doi.org/10.1101/2021.02.25.432728.

Dempster, Joshua M., Jordan Rossen, Mariya Kazachkova, Joshua Pan, Guillaume Kugener, David E. Root, and Aviad Tsherniak. 2019. “Extracting Biological Insights from the Project Achilles Genome-Scale CRISPR Screens in Cancer Cell Lines.” *bioRxiv*. https://doi.org/10.1101/720243.

Doench, J. G., N. Fusi, M. Sullender, M. Hegde, E. W. Vaimberg, and K. F. Donovan. 2016. “Optimized sgRNA Design to Maximize Activity and Minimize off-Target Effects of CRISPR-Cas9.” *Nature Biotechnology* 34. https://doi.org/10.1038/nbt.3437.

Doench, John G. 2018. “Am I Ready for CRISPR? A User’s Guide to Genetic Screens.” *Nature Reviews. Genetics* 19 (2): 67–80.

Evers, Bastiaan, Katarzyna Jastrzebski, Jeroen P. M. Heijmans, Wipawadee Grernrum, Roderick L. Beijersbergen, and Rene Bernards. 2016. “CRISPR Knockout Screening Outperforms shRNA and CRISPRi in Identifying Essential Genes.” *Nature Biotechnology* 34 (6): 631–33.

Gomez, V., and A. Hergovich. 2016. “Chapter 14 - Cell-Cycle Control and DNA-Damage Signaling in Mammals.” In *Genome Stability*, edited by Igor Kovalchuk and Olga Kovalchuk, 227–42. Boston: Academic Press.

Gonçalves, Emanuel, Fiona M. Behan, Sandra Louzada, Damien Arnol, Euan A. Stronach,

Fengtang Yang, Kosuke Yusa, Oliver Stegle, Francesco Iorio, and Mathew J. Garnett. 2019. “Structural Rearrangements Generate Cell-Specific, Gene-Independent CRISPR-Cas9 Loss of Fitness Effects.” *Genome Biology* 20 (1): 27.

Gonçalves, Emanuel, Aldo Segura-Cabrera, Clare Pacini, Gabriele Picco, Fiona M. Behan, Patricia Jaaks, Elizabeth A. Coker, et al. 2020. “Drug Mechanism-of-Action Discovery through the Integration of Pharmacological and CRISPR Screens.” *Molecular Systems Biology* 16 (7): e9405.

Gonçalves, Emanuel, Mark Thomas, Fiona M. Behan, Gabriele Picco, Clare Pacini, Felicity Allen, Alessandro Vinceti, et al. 2021. “Minimal Genome-Wide Human CRISPR-Cas9 Library.” *Genome Biology* 22 (1): 40.

Haeussler, Maximilian, Kai Schönig, Hélène Eckert, Alexis Eschstruth, Joffrey Mianné, Jean-Baptiste Renaud, Sylvie Schneider-Maunoury, et al. 2016. “Evaluation of off-Target and on-Target Scoring Algorithms and Integration into the Guide RNA Selection Tool CRISPOR.” *Genome Biology* 17 (1): 148.

Hart, T., M. Chandrashekhar, M. Aregger, Z. Steinhart, K. R. Brown, and G. MacLeod. 2015. “High-Resolution CRISPR Screens Reveal Fitness Genes and Genotype-Specific Cancer Liabilities.” *Cell* 163. https://doi.org/10.1016/j.cell.2015.11.015.

Hart, Traver, Kevin R. Brown, Fabrice Sircoulomb, Robert Rottapel, and Jason Moffat. 2014. “Measuring Error Rates in Genomic Perturbation Screens: Gold Standards for Human Functional Genomics.” *Molecular Systems Biology* 10 (July): 733.

Hart, Traver, and Jason Moffat. 2016. “BAGEL: A Computational Framework for Identifying Essential Genes from Pooled Library Screens.” *BMC Bioinformatics* 17 (April): 164.

Hart, Traver, Amy Hin Yan Tong, Katie Chan, Jolanda Van Leeuwen, Ashwin Seetharaman, Michael Aregger, Megha Chandrashekhar, et al. 2017. “Evaluation and Design of Genome-Wide CRISPR/SpCas9 Knockout Screens.” *G3* 7 (8): 2719–27.

Hsu, Patrick D., Eric S. Lander, and Feng Zhang. 2014. “Development and Applications of CRISPR-Cas9 for Genome Engineering.” *Cell* 157 (6): 1262–78.

Imkeller, Katharina, Giulia Ambrosi, Michael Boutros, and Wolfgang Huber. 2020. “Gscreend: Modelling Asymmetric Count Ratios in CRISPR Screens to Decrease Experiment Size and Improve Phenotype Detection.” *Genome Biology* 21 (1): 53.

Iorio, Francesco, Fiona M. Behan, Emanuel Gonçalves, Shriram G. Bhosle, Elisabeth Chen, Rebecca Shepherd, Charlotte Beaver, et al. 2018. “Unsupervised Correction of Gene-Independent Cell Responses to CRISPR-Cas9 Targeting.” *BMC Genomics* 19 (1): 604.

Iorio, Francesco, Theo A. Knijnenburg, Daniel J. Vis, Graham R. Bignell, Michael P. Menden, Michael Schubert, Nanne Aben, et al. 2016. “A Landscape of Pharmacogenomic Interactions in Cancer.” *Cell* 166 (3): 740–54.

Kim, Eiru, and Traver Hart. 2021. “Improved Analysis of CRISPR Fitness Screens and Reduced off-Target Effects with the BAGEL2 Gene Essentiality Classifier.” *Genome Medicine* 13 (1): 2.

Kurata, Morito, Kouhei Yamamoto, Branden S. Moriarity, Masanobu Kitagawa, and David A. Largaespada. 2018. “CRISPR/Cas9 Library Screening for Drug Target Discovery.” *Journal of Human Genetics* 63 (2): 179–86.

Li, Wei, Johannes Köster, Han Xu, Chen-Hao Chen, Tengfei Xiao, Jun S. Liu, Myles Brown, and X. Shirley Liu. 2015. “Quality Control, Modeling, and Visualization of CRISPR Screens with MAGeCK-VISPR.” *Genome Biology* 16 (December): 281.

Li, Wei, Han Xu, Tengfei Xiao, Le Cong, Michael I. Love, Feng Zhang, Rafael A. Irizarry, Jun S. Liu, Myles Brown, and X. Shirley Liu. 2014. “MAGeCK Enables Robust Identification of Essential Genes from Genome-Scale CRISPR/Cas9 Knockout Screens.” *Genome Biology* 15 (12): 554.

Lord, Christopher J., Niall Quinn, and Colm J. Ryan. 2020. “Integrative Analysis of Large-Scale Loss-of-Function Screens Identifies Robust Cancer-Associated Genetic Interactions.” *eLife* 9 (May): e58925.

Mali, Prashant, Luhan Yang, Kevin M. Esvelt, John Aach, Marc Guell, James E. DiCarlo, Julie E. Norville, and George M. Church. 2013. “RNA-Guided Human Genome

Engineering via Cas9." *Science* 339 (6121): 823–26.
Martinez-Lage, Marta, Pilar Puig-Serra, Pablo Menendez, Raul Torres-Ruiz, and Sandra Rodriguez-Perales. 2018. "CRISPR/Cas9 for Cancer Therapy: Hopes and Challenges." *Biomedicines* 6 (4). https://doi.org/10.3390/biomedicines6040105.
Mermel, Craig H., Steven E. Schumacher, Barbara Hill, Matthew L. Meyerson, Rameen Beroukhim, and Gad Getz. 2011. "GISTIC2.0 Facilitates Sensitive and Confident Localization of the Targets of Focal Somatic Copy-Number Alteration in Human Cancers." *Genome Biology* 12 (4): R41.
Meyers, Robin M., Jordan G. Bryan, James M. McFarland, Barbara A. Weir, Ann E. Sizemore, Han Xu, Neekesh V. Dharia, et al. 2017. "Computational Correction of Copy Number Effect Improves Specificity of CRISPR-Cas9 Essentiality Screens in Cancer Cells." *Nature Genetics* 49 (12): 1779–84.
Olshen, Adam B., E. S. Venkatraman, Robert Lucito, and Michael Wigler. 2004. "Circular Binary Segmentation for the Analysis of Array-based DNA Copy Number Data." *Biostatistics* 5 (4): 557–72.
Ong, Swee Hoe, Yilong Li, Hiroko Koike-Yusa, and Kosuke Yusa. 2017. "Optimised Metrics for CRISPR-KO Screens with Second-Generation gRNA Libraries." *Scientific Reports* 7 (1): 7384.
Pacini, Clare, Joshua M. Dempster, Isabella Boyle, Emanuel Gonçalves, Hanna Najgebauer, Emre Karakoc, Dieudonne van der Meer, et al. 2021. "Integrated Cross-Study Datasets of Genetic Dependencies in Cancer." *Nature Communications* 12 (1): 1661.
Panier, Stephanie, and Daniel Durocher. 2013. "Push back to Respond Better: Regulatory Inhibition of the DNA Double-Strand Break Response." *Nature Reviews. Molecular Cell Biology* 14 (10): 661–72.
Park, Ryan J., Tim Wang, Dylan Koundakjian, Judd F. Hultquist, Pedro Lamothe-Molina, Blandine Monel, Kathrin Schumann, et al. 2017. "A Genome-Wide CRISPR Screen Identifies a Restricted Set of HIV Host Dependency Factors." *Nature Genetics* 49 (2): 193–203.
Picco, Gabriele, Elisabeth D. Chen, Luz Garcia Alonso, Fiona M. Behan, Emanuel Gonçalves, Graham Bignell, Angela Matchan, et al. 2019. "Functional Linkage of Gene Fusions to Cancer Cell Fitness Assessed by Pharmacological and CRISPR-Cas9 Screening." *Nature Communications* 10 (1): 2198.
Sanjana, N. E., O. Shalem, and F. Zhang. 2014. "Improved Vectors and Genome-Wide Libraries for CRISPR Screening." *Nature Methods* 11. https://doi.org/10.1038/nmeth.3047.
Sanson, K. R., R. E. Hanna, M. Hegde, K. F. Donovan, C. Strand, and M. E. Sullender. 2018. "Optimized Libraries for CRISPR-Cas9 Genetic Screens with Multiple Modalities." *Nature Communications* 9. https://doi.org/10.1038/s41467-018-07901-8.
Shalem, O., N. E. Sanjana, E. Hartenian, X. Shi, D. A. Scott, and T. Mikkelson. 2014. "Genome-Scale CRISPR-Cas9 Knockout Screening in Human Cells." *Science* 343. https://doi.org/10.1126/science.1247005.
Smith, Ian, Peyton G. Greenside, Ted Natoli, David L. Lahr, David Wadden, Itay Tirosh, Rajiv Narayan, et al. 2017. "Evaluation of RNAi and CRISPR Technologies by Large-Scale Gene Expression Profiling in the Connectivity Map." *PLoS Biology* 15 (11): e2003213.
Subramanian, Aravind, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, et al. 2005. "Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-Wide Expression Profiles." *Proceedings of the National Academy of Sciences of the United States of America* 102 (43): 15545–50.
Symington, Lorraine S., and Jean Gautier. 2011. "Double-Strand Break End Resection and Repair Pathway Choice." *Annual Review of Genetics* 45 (September): 247–71.
Tsherniak, Aviad, Francisca Vazquez, Phil G. Montgomery, Barbara A. Weir, Gregory Kryukov, Glenn S. Cowley, Stanley Gill, et al. 2017. "Defining a Cancer Dependency Map." *Cell* 170 (3): 564–76.e16.
Tzelepis, K., H. Koike-Yusa, E. Braekeleer, Y. Li, E. Metzakopian, and O. M. Dovey. 2016. "A

CRISPR Dropout Screen Identifies Genetic Vulnerabilities and Therapeutic Targets in Acute Myeloid Leukemia." *Cell Reports* 17. https://doi.org/10.1016/j.celrep.2016.09.079.

Tzelepis, Konstantinos, Hiroko Koike-Yusa, Etienne De Braekeleer, Yilong Li, Emmanouil Metzakopian, Oliver M. Dovey, Annalisa Mupo, et al. 2016. "A CRISPR Dropout Screen Identifies Genetic Vulnerabilities and Therapeutic Targets in Acute Myeloid Leukemia." *Cell Reports* 17 (4): 1193–1205.

Venkatraman, E. S., and Adam B. Olshen. 2007. "A Faster Circular Binary Segmentation Algorithm for the Analysis of Array CGH Data." *Bioinformatics* 23 (6): 657–63.

Wang, Tim, Kıvanç Birsoy, Nicholas W. Hughes, Kevin M. Krupczak, Yorick Post, Jenny J. Wei, Eric S. Lander, and David M. Sabatini. 2015. "Identification and Characterization of Essential Genes in the Human Genome." *Science* 350 (6264): 1096–1101.

Wang, T., J. J. Wei, D. M. Sabatini, and E. S. Lander. 2014. "Genetic Screens in Human Cells Using the CRISPR-Cas9 System." *Science* 343. https://doi.org/10.1126/science.1246981.

Weck, Antoine de, Javad Golji, Michael D. Jones, Joshua M. Korn, Eric Billy, E. Robert McDonald 3rd, Tobias Schmelzle, Hans Bitter, and Audrey Kauffmann. 2018. "Correction of Copy Number Induced False Positives in CRISPR Screens." *PLoS Computational Biology* 14 (7): e1006279.