# Discovering functional sequences with RELICS, an analysis method for CRISPR regulatory screens

Patrick C Fiaux[1,2*], Hsiuyi Chen[2], Graham McVicker[2*]

[1]Bioinformatics and Systems Biology Graduate Program, University of California, San Diego, La Jolla, CA 92039, USA.

[2]Integrative Biology Laboratory, Salk Institute for Biological Studies, 10010 N Torrey Pines Rd, La Jolla, CA 92037, USA

*Correspondence should be addressed to: pfiaux@ucsd.edu or gmcvicker@salk.edu

**Abstract**

CRISPR regulatory screens are a powerful technology for discovering sequences that control gene expression, however, a lack of analysis methods limits their effectiveness. In addition, method performance is difficult to assess due to an absence of datasets where the identities of true regulatory sequences are known. To address these problems, we developed an analysis method, RELICS, and a simulation framework, CRSsim, for CRISPR regulatory screens. RELICS detects regulatory elements by modeling guide counts across multiple pools representing different conditions or expression levels and CRSsim generates realistic datasets where the ground truth is known. We used CRSsim to generate 4320 datasets representing 144 scenarios with different characteristics. We compared analysis methods on these datasets and found that RELICS has the best performance under most conditions. We applied RELICS to 8 published datasets for 15 genes and identify previously-validated elements as well as multiple putative elements that were missed by other methods.

**Introduction**

CRISPR regulatory screens identify gene-specific regulatory elements by targeting thousands of sequences for mutation, repression or activation (Fig. 1a). To target sequences for mutation, single-guide RNAs (sgRNAs) are introduced to cells alongside Cas9. Cas9 creates double strand breaks at the targeted sites, and mutations are introduced by the error-prone non-homologous end-joining process[1]. In a CRISPR interference (CRISPRi) experiment, targeted sites are silenced by a deactivated Cas9 (dCas9) enzyme fused to a repressive domain such as the Krüppel-associated box (dCas9:KRAB)[2,3]. Similarly, in a CRISPR activation (CRISPRa) experiment, dCas9 is fused to an activation domain such as VP64 or p300[4]. Finally, in a tiling deletion screen, small overlapping deletions are used to interrogate large stretches of the genome. The deletions are programmed by pairs of sgRNAs that direct Cas9 to introduce two nearby double-strand breaks resulting in the loss of the intervening sequence[5,6]. The cells containing guides are subjected to selection (e.g. treatment with a drug), allowed to proliferate, or sorted into different pools using fluorescence activated cell sorting

(FACS). Counts of guides are then obtained from pools of cells that are collected from different timepoints or FACS bins.

CRISPR regulatory screens have important advantages over other approaches for regulatory sequence discovery, which typically rely upon genomic signatures such as histone modifications (e.g. H3K4me1 and H3K27ac)[7,8], open chromatin[9,10], and enhancer RNAs (eRNAs)[11–13]. First, unlike massively parallel reporter assays (MPRAs)[14–16], CRISPR regulatory screens measure the effects of targeted regulatory sequences in their native sequence context. Second, they connect regulatory sequences to the genes that they control. Finally, CRISPR regulatory screens can discover regulatory sequences that lack canonical enhancer marks.

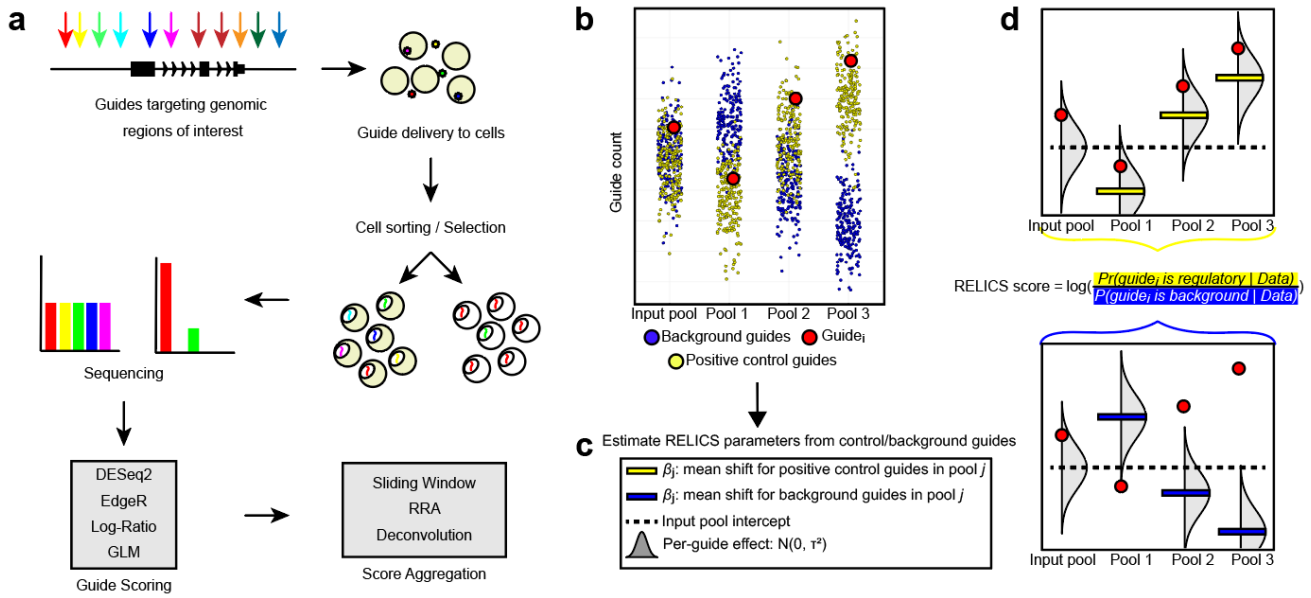



**Figure 1** | Experimental workflow of a CRISPR regulatory screen and the RELICS analysis method. **(a)** In a CRISPR regulatory screen, guides are designed to target both coding and non-coding regions. Guides are transduced into cells and the cells expressing guides are (i) selected for survival, (ii) selected for proliferation, or (iii) sorted into pools based on gene expression. The integrated guides in each pool are then sequenced and counted. The counts are used to score individual guides and the scores of multiple nearby guides are aggregated to give scores for genomic regions. **(b)** RELICS estimates model parameters from a set of positive control guides and background guides. **(c)** An intercept parameter is estimated for the first pool and mean shifts (βs) are estimated for subsequent pools. A per-guide random effect is the deviance of a guide from the pool means (this accommodates variability in guide frequencies in the vector library). **(d)** The RELICS score is the log likelihood ratio of the regulatory model to the background model, given the observed counts for a guide.

A variety of methods have been used to analyze data generated by CRISPR regulatory screens including log-ratios of normalized guide counts, and methods designed to detect differential gene expression from RNA-seq (edgeR[17] and DESeq2[18]). Unfortunately, only a small subset of the elements identified using these approaches have been experimentally validated and for most genomic regions the ground truth (i.e. whether a sequence truly regulates a gene or not) is unknown. Thus, there is currently no way to compare the performance of different analysis methods. To address this problem, we have developed CRSsim (**C**RISPR **R**egulatory **S**creen **s**imulator), an extensive simulation framework that generates datasets where the ground truth is known.

In addition, CRISPR screens often generate counts from multiple pools (e.g. input, high-expression, medium-expression, low-expression), yet existing tools are not capable of analyzing all of the pools at once and typically resort to pairwise comparisons between pools. To address this problem, we have developed RELICS (**R**egulatory **E**lement **L**ocation **I**dentification in **C**RISPR **S**creens), a new statistical method for the analysis of CRISPR regulatory screens which can jointly analyze guide counts across multiple pools or time points. We simulate datasets under a wide variety of conditions and systematically compare the performance of different analysis approaches for CRISPR regulatory screens. Under most simulation conditions RELICS is the best-performing analysis method, and when we apply RELICS to published datasets we not only identify most previously-validated elements but also several putative regulatory elements that were missed in the initial studies.

## Results

### RELICS for CRISPR regulatory screen analysis

We have developed RELICS, a flexible statistical framework that can analyze different types of screens including those based on cell dropout, cell proliferation, and cell sorting into expression pools (Fig. 1b-d). RELICS uses a Generalized Linear Mixed Model (GLMM)[19] to describe the observed counts of each guide across pools, where each pool is a timepoint or condition. For example, in a dropout screen there

will typically be two pools, one before selection and one following selection. In an expression screen, the pools are typically the input pool and expression bins that the cells are sorted into.

To identify guides targeting regulatory sequences, RELICS computes the likelihood of the observed counts for a guide under two models: a 'regulatory' model (H1: the guide targets a regulatory sequence) and a 'background' model (H0: the guide does not target a regulatory sequence). The RELICS score is the log ratio of these two likelihoods, which can be interpreted as a log Bayes Factor. The regulatory and background models are both GLMMs, with parameters that are estimated empirically from subsets of guides in the dataset. The parameters for the regulatory model are estimated from guides that are designated as positive controls; typically, these guides target the exons or promoter of the gene of interest. The parameters for the background model are estimated either (1) from all guides that are not designated as positive controls, or (2) from a set of guides that are designated as negative controls (e.g. non-targeting guides). A major advantage of RELICS is that it jointly models the data from multiple pools, and the behavior of each pool is learned empirically from the dataset. This approach differs substantially from previous analysis methods which can only perform pairwise comparisons of pools (e.g. high expression to low expression).

After computing a score for each guide, RELICS assigns scores to base positions in the genome by summing the log Bayes Factors of the guides that overlap that position (Supplementary Fig. 1). A base position is considered to be overlapped by a guide if it is within the 'area of effect' for the CRISPR system used. For a single guide Cas9 screen, the area of effect is assumed to be 20bp[20], for CRISPRi and CRISPRa the area of effect is assumed to be 1kb[3], and for a paired guide deletion screen, the area of effect is specified by each deletion.

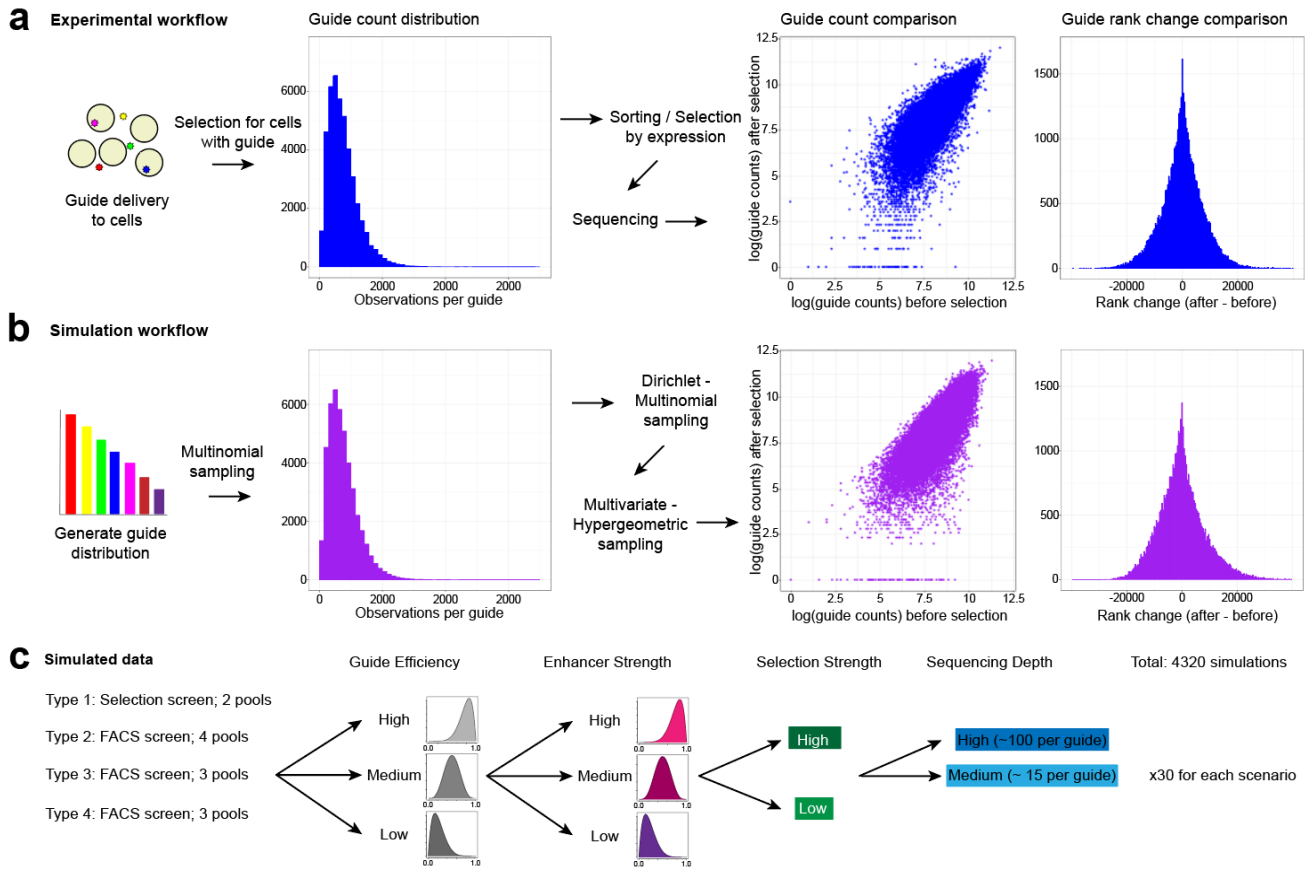



**Figure 2** | Overview of the CRISPR regulatory screen simulator (CRSsim). **(a)** In an experiment, an input pool of cells containing guides is sorted based on gene expression or undergoes selection for survival or proliferation. Guides in the input pool and output pool(s) are sequenced, counted and compared. Guide count data from Fulco et al. 2016 are shown as an example. **(b)** CRSsim mimics the experimental workflow. Cells containing guides are drawn from an input distribution and sorted into different pools with probabilities defined by a Dirichlet Multinomial distribution. Guide sequencing is simulated by sampling from input and output pools. The simulated guide count distributions are very similar to the experimentally-observed guide counts (see also Suppl. Figs 2-4). **(c)** Simulation parameters that were varied to generate data for 144 experimental and biological scenarios. 30 datasets were simulated for each scenario.

## Simulation framework

A major limitation in assessing the performance of analysis methods for CRISPR regulatory screens is that there is no gold standard dataset where all regulatory elements are known. To address this limitation, we developed a simulation framework, CRSsim, to systematically compare the performance of RELICS and other methods under different conditions. The simulations capture the effects of

experimental and biological variables by mimicking the experimental workflow of CRISPR regulatory screens (Fig. 2a). Experimental variables include the complexity of the vector library, the efficiency of the guides, the spacing of sequences targeted by guides, the sequencing depth of the different pools, and the strength and type of selection that is performed (e.g. drop-out or flow-sorting by expression). Biological variables include the number of regulatory elements, the genomic size of regulatory elements, and the effect sizes of the regulatory elements (i.e. how they affect gene expression when perturbed). CRSsim incorporates these variables by drawing cells containing guides from an input distribution; sorting cells into different pools with probabilities defined by a Dirichlet-Multinomial distribution; and "sequencing" the guides in the pools by sampling either with or without replacement (Fig. 2b).

We evaluated the similarity of data simulated by CRSsim to experimental data by comparing the distributions of guide counts, and rank changes in guide counts after selection/sorting (Fig. 2a,b, Supplementary Fig. 2-4). In most cases, we found the experimental and simulated guide datasets to be similar.

**Performance assessment of RELICS and other analysis methods with simulated data**

To evaluate the performance of RELICS and other analysis methods under a wide variety of conditions, we generated over 4000 datasets with CRSsim (Fig. 2c). We performed 3 major types of simulation, each of which was designed to resemble the experimental design of a published screen. For all simulation types, we set the input guide distribution to match the distribution observed in the original experiment. Type 1 is based on the CRISPRi proliferation screen performed by Fulco et al.[21]. Type 2 is based on the gene expression screen by Simeonov et al.[22] and sorts cells into 4 different pools (high-expression, medium expression, low expression, and no expression). Type 3 is based on the expression screen of Diao et al.[5] and sorts cells into high, medium, and low expression pools. We additionally performed a fourth type of simulation, Type 4, that is also based on Diao et al., with the modification that the medium pool is assumed to have no contribution to the overall signal (i.e. positive

and negative guides are equally likely to be sorted into the medium pool). The purpose of the latter simulation is to confirm that RELICS still performs well when one of the pools provides little extra information.

Within each of the four simulation types we varied the enhancer strength (low, medium, high), the guide efficiency (low, medium, high), the selection strength (low, high), and the sequencing depth (medium, high). The enhancer strength describes how strongly the enhancers affect the expression of the gene, while the selection strength describes the difference in sorting (or dropout) probabilities between positive and negative controls. We set the medium and high sequencing depths to an average of 15 or 100 reads per guide respectively. In each simulation we generated data for 8,700 guides targeting a 150kb region containing 25 enhancers, where each enhancer spans 50bp. The combinations of the 4 simulation types and the 4 variables described above result in a total of 144 different simulation scenarios. We performed 30 simulations for each scenario to generate a total of 4320 simulation datasets. For this analysis, we simulated CRISPRi screens, however CRSsim can also simulate CRISPRa screens, tiling deletion screens, and Cas9-based screens.

We ran RELICS and four other analysis methods (edgeR[17], DESeq2[18], fold change and CRISPRsurf[23]) on the simulated datasets. Both edgeR and DESeq2 test whether the rate of guide counts differs between two pools (e.g. high expression vs. low expression), taking into account sequencing depth and overdispersion. We used edgeR and DESeq2 to compute a p-value for each guide and combined p-values for overlapping guides using Fisher's method. For the fold change analysis method, we computed the mean fold change across overlapping guides. CRISPRsurf takes scores as input (by default the scores are fold change) and performs a LASSO-based deconvolution to estimate effect sizes and p-values for each genomic position. We tested CRISPRsurf using both fold change and RELICS guide scores as input.

To assess the performance of the analysis methods, we divided the simulated regions into positive (regulatory) and negative (non-regulatory windows). Since each window spans multiple sites with potentially different scores, we assigned each window the highest score from all of the sites within the window. We then computed precision-recall curves for the windows and assessed the overall performance of each analysis method on each dataset by computing the area under the precision-recall curve (prAUC).

RELICS had the best performance out of all of the methods on most of the simulated datasets and had the highest median prAUC for 86/144 simulation scenarios (Fig. 3, Supplementary Fig. 5). Even in the 58 scenarios where RELICS was not the best method, RELICS' performance was very close to the best with a mean difference in median prAUC of -0.03 (Supplementary Fig. 6). Fold-change also performed reasonably well across most datasets, whereas DESeq2, edgeR, and CRISPRsurf performed inconsistently. In particular, these latter three methods worked well on simulations with strong enhancers but performed poorly on data from simulations with weak enhancers or weak selection strength. We also tested different methods for combining guide scores including sliding windows and MAGeCK[24] (DESeq2 paired with α-RRA) however these modifications diminished performance (Supplementary Fig. 7). Note that our performance comparison of analysis methods should not be biased to favor RELICS as the simulation framework was designed to mimic experimental procedures and is in no way tailored to the RELICS analysis model. In summary, RELICS consistently has the best, or close to the best, performance under a wide variety of simulation scenarios.

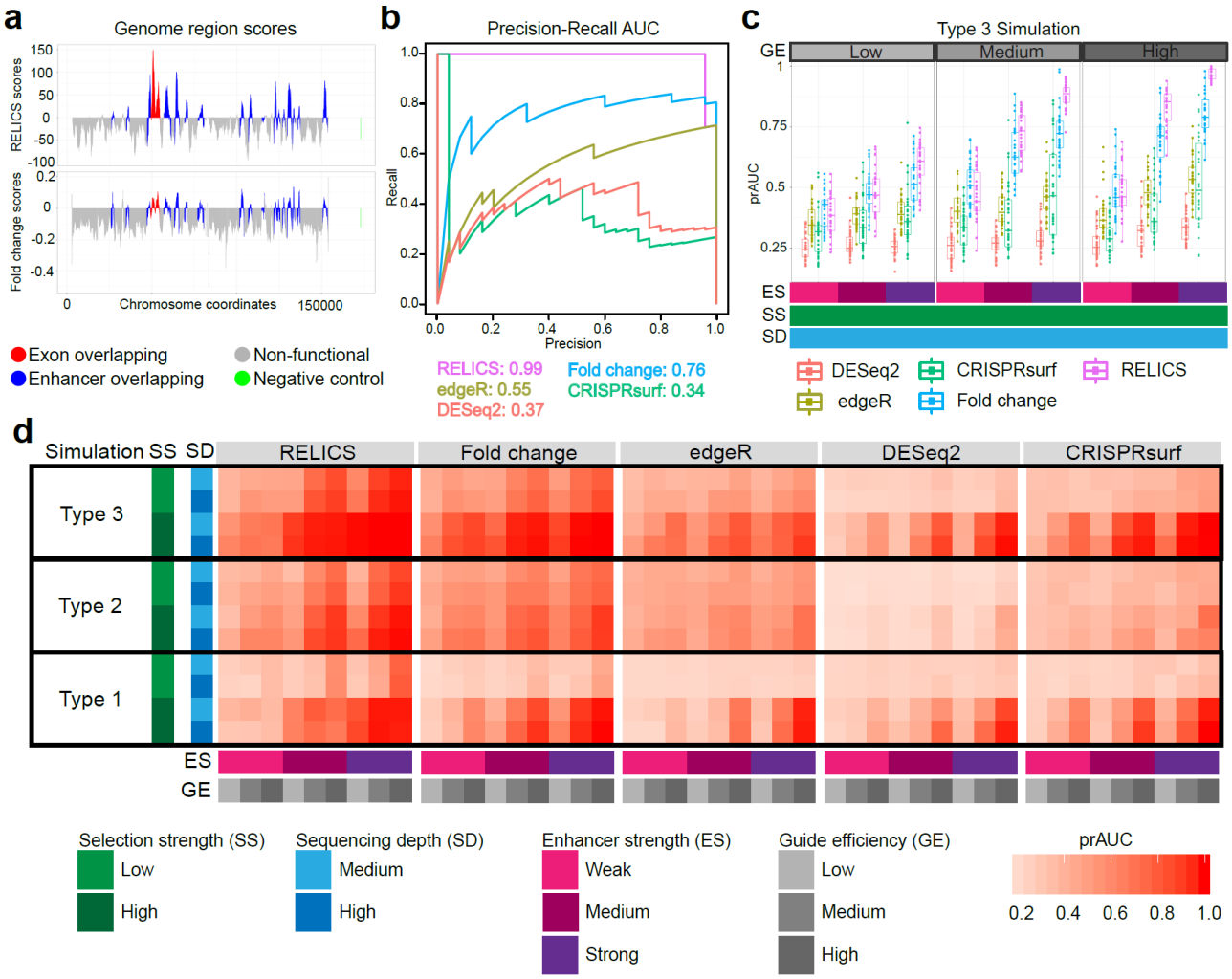



**Figure 3** | RELICS performance on a single simulation dataset. **(a)** Example of genome region scores for RELICS (top) and log fold change (bottom) on a single "Type 3" simulation with medium guide efficiency (GE), medium enhancer strength (ES), medium sequencing depth (SD), and high selection strength (SS). **(b)** Precision-recall curves for 5 different analysis methods applied to the simulation shown in panel a. **(c)** Area under the precision-recall curves (prAUC) for simulation scenarios with different parameter settings for ES (x-axis) and GE (panels). 30 simulations were performed per scenario. Upper and lower hinges represent 25$^{th}$ and 75$^{th}$ percentiles. **(d)** Heatmap of median prAUC (across 30 simulations) for 5 analysis methods run on all 144 simulation scenarios.

To determine an appropriate RELICS score cutoff for identifying regulatory elements, we computed the average false discovery rate (FDR) for all 144 simulation scenarios under different RELICS score thresholds (Supplementary Fig. 8). A score cutoff of 5 results in a FDR below 10% under most simulation scenarios with strong selection strength and we applied this threshold for the analysis of public data sets described below. Simulations with low selection strengths, low guide efficiencies, or

low sequencing depths had higher FDRs under this threshold. A more stringent cutoff of 10, generally results in FDRs beneath 10% even for simulations with low guide efficiencies as long as there is strong selection and medium to strong enhancers (Supplementary Fig. 8c).

**Application of RELICS to published datasets**

We applied RELICS to 15 genes from 8 published studies[5,6,21,22,25–29]. For most datasets, RELICS detects all of the regulatory elements that have been experimentally validated and identifies additional putative regulatory elements in several cases (Supplementary Table 1). We detail our findings from a subset of these studies here.

We applied RELICS to data from Simeonov et al. 2017, in which a CRISPRa screen[22] was performed to search for regulatory elements for *CD69* (Fig. 4a) and *IL2RA* (Fig. 4b) in Jurkat T cells. For both genes, cells were sorted into 4 pools based on expression (negative, low, medium, high), and in their published analysis, the authors performed pairwise comparisons of pools using fold change. For *CD69*, the authors identified 3 regulatory sites, one of which was a previously-reported enhancer for *CD69*, known as CNS2[30]. We analyzed all pools jointly with RELICS and found that the RELICS results are visibly less noisy than those from the fold-change analysis, with cleanly-delineated predicted regulatory regions. RELICS detected all 3 of the previously identified elements for *CD69*, as well as an additional region. This region is located in an evolutionarily-conserved region 7kb upstream of the transcription start site (TSS) with a weaker, but highly significant, RELICS score of 15 (Supplementary Fig. 9). For *IL2RA*, 6 putative regulatory elements were identified. Only 1 of the 6 exhibited enhancer activity in luciferase reporter assays, but 2 of the elements were confirmed to affect *IL2RA* expression by transduction of single guides and flow cytometry. RELICS detected both of the confirmed regulatory elements but did not detect any of the unconfirmed elements, which suggests that some of the predictions from the fold-change analysis method may have been false positives.

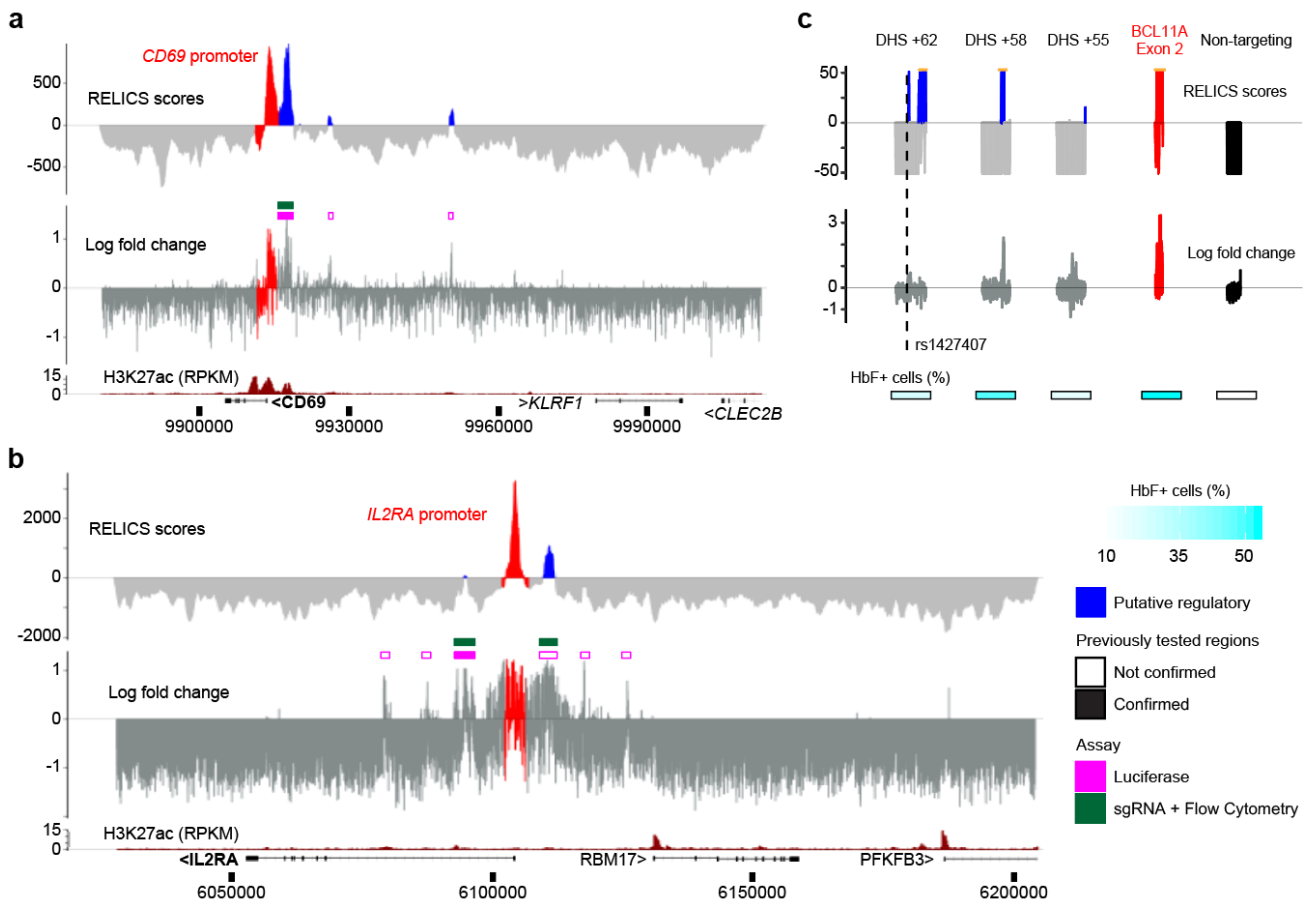



**Figure 4** | Analysis of published datasets using RELICS. **(a)** Analysis of a CRISPRa FACS screen for *CD69* regulatory elements by Simeonov et al. 2017. The top panel shows RELICS scores, and the bottom panel shows results from a windowed log fold change analysis, as performed in the original study. RELICS detects three putative regulatory elements. The validation status of two assays performed by Simeonov et al. is indicated by open or filled pink (luciferase assay) and green squares (sgRNA + flow cytometry). **(b)** Analysis of CRISPRa FACS screen for *IL2RA* regulatory elements by Simeonov et al. 2017. RELICS detects both elements that were validated with sgRNAs and flow cytometry. **(c)** Analysis of a Cas9 FACS screen for human *BCL11A* by Canver et al. 2017. Each region is a DNase hypersensitive site (DHS) upstream of *BCL11A* that was targeted by sgRNAs. RELICS detects two putative regulatory elements in DHS +62, one element in DHS +58, and one element in DHS +55. The percentage of HbF positive cells (associated with downregulation of *BCL11A*) after deletion of each DHS was measured by Canver et al. and is indicated by cyan shading.

We next applied RELICS to FACS Cas9 screens for human *BCL11A* and its mouse homolog, *Bcl11a*[26]. Canver et al. 2017 targeted sequences within and around three DNase hypersensitive sites (DHSs) at +55, +58 and +62kb from the TSS. The original analysis for human *BCL11A* showed a strong signal in h+58, modest signals in h+55 and weak signals in h+62. RELICS detected two very significant signals in h+62 (scores of 51 and 104), a single highly-significant region in DHS+58 (score of 186) and a

weaker signal in DHS+55 (score of 15). Notably, one of the two significant regions in h+62 is immediately adjacent to a SNP (rs1427407) that is associated with *BCL11A* expression and β-hemoglobin disorders (Fig. 4c)[31]. Using the data for mouse *Bcl11a*, RELICS discovered the regulatory element in the previously-reported m+62 region. In addition, we also detected two putative regulatory elements in the m+58 region, which were not identified in the original study. (Supplementary Fig. 10).

Finally, we applied RELICS to data from a CRISPRi proliferation screen by Fulco et al. 2016, which targeted non-coding regions around *GATA1* and *MYC* in K562 cells[21] (Supplementary Fig. 11). The authors found 3 regulatory regions around *GATA1* and validated two of them. RELICS detected both validated regions but did not detect the non-validated region near *GLOD5* (Supplementary Fig. 11a). This suggests that the non-validated region near *GLOD5* may be a false positive detected by fold-change. Fulco et al. also found 7 genomic regions around *MYC* (e1 – e7) that affect cellular proliferation and that were interpreted as likely enhancers for *MYC*. RELICS detected these same 7 regions with a stringent score cutoff of 10. At a less-stringent cutoff of 5, RELICS detected an additional 18 regions which may serve as weaker regulatory regions. Notably, RELICS scores are substantially less noisy than those obtained using log fold change (Supplementary Fig. 11b).

**Discussion**

Using data generated by CRSsim, we performed the first systematic comparison of analysis methods for CRISPR regulatory screens. CRSsim is open-source and we envision that it will be a useful tool for future performance comparisons, power analyses, and making informed decisions about experimental designs for CRISPR regulatory screens (e.g. spacing of sites targeted by sgRNAs, the sequencing depth, and the complexity of the vector library).

We also introduced RELICS, which identifies regulatory elements by comparing the likelihood of the observed guide counts under a regulatory model to the likelihood under a background (null) model. RELICS jointly learns the parameters of each model, using a set of guides that are designated as

positive and negative controls. This is both a strength and limitation of RELICS. A strength of this approach is that the behavior of guides across different pools is learned from the data itself. A limitation of this approach is that it is difficult to apply RELICS to datasets that do not have many positive controls and the positive controls may not adequately represent all types of regulatory elements, especially repressive or weak regulatory elements. These limitations could potentially be addressed with an unsupervised learning approach where training labels are not provided and instead categories of similarly-behaving guides are identified by clustering the data. Ideally the categories identified by this approach would represent different types of elements (e.g. strong regulatory elements, weak regulatory elements, silencers, non-regulatory elements).

RELICS does not currently model variation in guide efficiency, variation in the strength of regulatory elements, or off-target effects. These limitations are somewhat mitigated by combining information across multiple guides when computing scores for genomic positions, and our simulations demonstrate that RELICS performs reasonably well, even in the presence of weak enhancers or weak guide efficiency. Nonetheless, future versions of RELICS that explicitly model these sources of variation will likely achieve even better results.

In summary, we developed a flexible simulation framework, CRSsim, that allowed us to systematically evaluate the performance of analysis methods for CRISPR regulatory screens under different scenarios. We also developed RELICS, a powerful and flexible tool for the analysis of CRISPR regulatory screens that incorporates information from an arbitrary number of guide pools. RELICS performs better than previous analysis methods under most simulation conditions and, when applied to real data, identifies both previously-validated regulatory elements and new putative regulatory elements. Thus, RELICS is an extremely useful tool for the discovery of regulatory elements from CRISPR screens.

## Online Methods

### Simulation framework

CRSsim is open source and available on GitHub (https://github.com/patfiaux/CRSsim). To run it, the user first specifies the screen type, which can either be a selection screen (e.g. drop out or proliferation) or an expression screen, with an arbitrary number of expression pools. (We describe the simulation framework for an expression screen, but the same principles hold for selection screens.) Next, the user specifies (1) the screening method (Cas9, CRISPRi, CRISPRa, dual-guide CRISPR), (2) the number and size of true regulatory elements, (3) the spacing/density of guide target sites, and (4) the distribution of guide counts in the initial set of cells (i.e. the frequency of each guide).

The probability of a cell being sorted into each of the pools depends on whether the guide in that cell targets a regulatory sequence. We refer to the difference in sorting probabilities between guides that do/do not target regulatory sequences as the "strength of selection". The sorting of guides into the different pools is simulated by sampling from a Dirichlet-Multinomial distribution. We also model both the efficiency of guides and the strength of regulatory elements as fractions between 0 and 1. Guides targeting a strong enhancer, with a strength near 1, behave like positive control guides. The simulation sets the efficiency of each guide and the strength of each enhancer by sampling from Beta distributions that have configurable shapes. Finally, the user can specify how deeply the guide pools are sequenced.

The sorting probabilities of a guide that does not target a regulatory element are given by a vector of Dirichlet parameters $B$, where the probability of being sorted into pool j is $\frac{B_j}{\sum_k B_k}$. The sorting probabilities of guides that overlap a regulatory element depend on the strength of selection, *T*, the guide efficiency, *g*, and the enhancer strength, *h*. Both *g* and *h* are proportions between 0 and 1, which are obtained by observing Beta random variables: $g \sim Beta(shape_1, shape_2)$, $h \sim Beta(shape_3, shape_4)$. The strength of selection, *T*, is a vector that gives the difference in sorting probabilities from $B$.

The guide efficiency for guide *i*, $g_i$, specifies the fraction of cells containing the guide where the guide is 'effective' and perturbs the sorting probabilities. For example, if $c_i$ cells contain guide *i*, then $w_i = c_i g_i$ cells contain guides that are effective and they are sorted with probabilities specified by the Dirichlet vector $R_i = B + h_k T$, where $T$ is the strength of selection, $h_k$is the strength of the enhancer overlapped by guide i, and $B$ is the baseline Dirichlet sorting vector described above. For guides that overlap positive controls (e.g. exon or promoter) the sorting vector for the $w_i$ effective cells is given by $R_i = BT$. The sorting vector for the remaining $c_i - w_i$ cells with 'ineffective' guides is simply $B$.

Each simulation starts with a guide library with a distribution of guide counts. We assume the guide counts follow a zero-inflated negative binomial distribution (ZINB), $Y \sim ZINB(\mu, d, \varepsilon)$, where $\mu$ is the mean, $d$ the dispersion and $\varepsilon$ the proportion of the distribution that comes from the zero point mass. The parameters of these distributions can be specified or estimated by maximum likelihood from a provided input pool.

The pool used for sorting contains millions of cells, which we sample from to form the input pool using a multinomial sampling step. Each guide is selected for the input pool with a probability $p = \frac{c_i}{\sum_j c}$, where $c_i$ is the number of cells containing guide *i*. We simulate the sorting of cells into different pools by sampling counts for each guide from a Dirichlet-Multinomial (DM) distribution which allows the sorting probabilities and dispersion (variability in sorting) to be specified. If a guide targets a regulatory element, the Dirichlet parameters are shifted by the vector $T$ such that a greater proportion of cells containing the guide are sorted into the low expression and the medium expression pools (for details see above). Lastly, we simulate the sequencing step by drawing from a multivariate hypergeometric distribution (sampling without replacement) or a multinomial distribution (sampling with replacement) that reflects the counts of guides in the sorted pools. Sampling without replacement is used to simulate the use of unique molecular identifiers that allow duplicate reads to be filtered.

## Performance assessment in simulations

We compared the performance of several analysis methods for CRISPR regulatory screens. Additionally, we compared different ways to combine per-guide scores to compute scores for genome regions. Both sliding windows[21,22] as well as a modified robust ranking algorithm (α-RRA)[5,24] have previously been used to aggregate guide scores within a specific region. Briefly, a sliding window approach combines the scores of all guides that fall within a window. The window is then shifted by one guide and the scores are computed for the next window. This approach is simple, however, the size of the window to use is typically not clear especially given the variable spacing of target sites. The α-RRA approach ranks all guide scores, places guides into bins and tests bins for an enrichment of low ranks. This approach is powerful and p-values can be computed using permutations. However, it is not clear how large the bins should be and we have found this approach to be sensitive to outliers.

We implemented both methods in addition to the genome score calculation described above. Briefly, in the genome score method, we assign base pair positions scores by combining scores from guides with overlapping areas of effect (Supplementary Fig. 1). The advantage of combining scores across regions of overlapping effect is that there is no need to set arbitrary bin sizes—instead the range is defined by the CRISPR system used. An additional improvement could be to describe the area of effect with a gaussian distribution instead of assuming a uniform effect across a region, as has been implemented in CRISPR-surf[23]. We compared all three methods for combining guide scores and found that the best-performing method was RELICS when the per-guide scores are combined across their region of effect (Supplementary Fig. 7). Note that the combination of α-RRA and DESeq2 is equivalent to the MAGECK method which is commonly used to analyze screens of protein coding genes[24].

We assessed the performance of different analysis methods in simulations by calculating the area under the precision-recall curve (prAUC).

## RELICS analysis model

RELICS is open source and available on GitHub (https://github.com/patfiaux/RELICS). RELICS represents the observed guide counts as a column vector $\boldsymbol{y}$ of length $n = km$, where $k$ is the number of

sgRNAs (or sgRNA pairs) and $m$ is the number of pools. The observations are assumed to be from a Negative Binomial random variable that can be described by a generalized linear mixed model (GLMM)[19] with the following form:

$$\mathrm{E}(\mathbf{Y}) = \boldsymbol{\lambda} = \exp(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathrm{u})$$

$$\mathbf{Y} \sim \mathrm{NB}(\boldsymbol{\lambda}, d)$$

Here $\boldsymbol{\lambda}$ is the expected number of guide counts and $d$ is a dispersion parameter. We use a Negative Binomial distribution rather than a Poisson distribution to describe the guide counts, because, like most next-generation sequencing data, we have found that the guide counts are over-dispersed[17,18,32].

We number the pools so that the first pool (pool 1) is the 'reference' pool. The reference pool will typically be the input pool, which contains cells that have not been subjected to sorting or selection. If an experiment does not provide data from an input pool, one of the pools is arbitrarily chosen to be the reference.

The fixed-effect vector, $\boldsymbol{\beta} = [\beta_1\ \beta_{b2} \ldots\ \beta_{bm}\ \beta_{r2} \ldots \beta_{rm}]^{\mathrm{T}}$ is a column vector of length $2m - 1$. $\beta_1$ is an intercept that describes the expected count in the reference pool, and the $\beta_b$ coefficients are the expected differences in guide counts from $\beta_1$ for the other pools under the 'background' model. The background model assumes guides have no effect on selection or gene expression, but their counts are still expected to differ across pools due to other factors such as sequencing depth. The $\beta_r$ coefficients are the expected differences in the guide counts under the 'regulatory' model where the presence of the guide in a cell shifts the probability that the cell will be present in each of the pools.

The design matrix, $\mathbf{X}$, specifies which fixed-effects should be combined for a particular guide count observation, and can be written as a block matrix $\mathbf{X} = [\mathbf{X_1 B R}]$. Here $\mathbf{X_1}$ is an $n \times 1$ column matrix of 1s corresponding to the reference pool intercept, $\mathbf{B}$ is an $n \times (m-1)$ matrix corresponding to the

background coefficients and $\mathbf{R}$ is a $n \times (m-1)$ matrix corresponding to the regulatory coefficients. The elements of $\mathbf{B}$ are:

$$b_{ij} = \begin{cases} 1 & \text{if observation } i \text{ from pool } j+1 \\ 0 & \text{otherwise} \end{cases}$$

Similarly, the elements of $\mathbf{R}$ are:

$$r_{ij} = \begin{cases} \delta & \text{if observation } i \text{ from pool } j+1 \\ 0 & \text{otherwise} \end{cases}$$

where $\delta = 1$ for guides that target regulatory elements, and $\delta = 0$ otherwise.

The random effect term describes the per-guide deviation from the mean rate defined by the fixed effects. This deviation is necessary because the frequency of each guide in the input library is highly variable. We use a random effect rather than fixed effects to describe the per-guide deviation to keep the number of parameters in the model small and avoid overfitting. The random effects matrix, $\mathbf{Z}$, has dimension $n \times m$ with elements that indicate the guide the observations are from:

$$z_{ij} = \begin{cases} 1 & \text{if observation } i \text{ from guide } j \\ 0 & \text{otherwise} \end{cases}$$

The random effect for guide *j* is assumed to be normally distributed with variance $\tau^2$:

$$u_j \sim \mathrm{N}(0, \tau^2)$$

We jointly estimate the parameters of the RELICS model numerically by maximum likelihood, using as data observed counts for guides which have been labeled as regulatory (i.e. positive control guides) or non-regulatory. The set of non-regulatory guides can be the complete set of non-positive control guides or a set of negative control guides. In total there are $2m+1$ free parameters in the model $\theta = (\boldsymbol{\beta}, d, \tau)$. We perform bootstrap parameter estimation, estimating the parameters from a random sample of 1/3 of the guides. We perform 10 bootstrap iterations and use the median parameter estimates from these iterations as our final parameter estimates. We fit RELICS parameters using the glmmTMB R package[33] which is fast due to its use of a Laplace approximation for evaluating integrals and template metaprogramming. Fitting of parameters for a screen with 2 replicates containing 4 pools each and ~8700 guides takes roughly 1000 seconds on a laptop with an intel i7 core.

Once RELICS' parameters have been estimated, we can use them to calculate the likelihood of observed counts for a single guide, $i$:

$$\Pr(\boldsymbol{y}_i|\theta,\delta) = \int p_{NB}(\mathrm{Y} = y_{\mathrm{i1}}|\lambda = \exp[\beta_1 + u], d) \prod_{j=2}^{m} p_{NB}\big(\mathrm{Y} = y_{\mathrm{ij}}\big|\lambda = \exp\big[\beta_1 + \beta_{Bj} + \delta\beta_{Rj} + u\big], d\big)\, p_N(u|\mu = 0, \tau^2)du$$

where we have written the counts for guide $i$ in pool $j$ *as* $y_{ij}$ and $p_{NB}$ and $p_N$ are the probability density functions for the Negative Binomial and Normal distributions. To compute a RELICS score we take the log ratio of the probabilities under the regulatory model and background models, where $\delta = 1$ and $\delta = 0$ respectively:

$$S = \log\left[\frac{\Pr(\boldsymbol{y}_i|\theta,\delta = 1)}{\Pr(\boldsymbol{y}_i|\theta,\delta = 0)}\right]$$

This score can be considered a log Bayes Factor, where $u$ has a normal prior distribution with an empirically-estimated hyperparameter, $\tau^2$. Note that we do not integrate over the $\boldsymbol{\beta}$ or $d$ parameters, which is equivalent to assuming that they have point mass prior probabilities with no uncertainty. This assumption is reasonable since these parameters are typically estimated from a large number of guides, and it greatly reduces the required computation time. As defined above, the RELICS scores assume that the prior probabilities of a guide targeting a regulatory or non-regulatory sequence are equal ($\Pr(\delta = 1) = \Pr(\delta = 0) = 0.5$). For most screens, this is unlikely to be true, but the scores can be adjusted according to one's prior belief by adding a log prior ratio. For example, if 5% of guides are believed to target a regulatory sequence the scores can be adjusted as follows: $S' = S + \log\left(\frac{0.05}{0.95}\right)$. In practice, we assume that scores > 5 provide reasonably strong evidence that a guide targets a regulatory sequence and that scores > 10 provide very strong evidence. This is further corroborated by our FDR calculations described above (Supplementary Fig. 7).

**CRISPR regulatory screen data sources**

RELICS takes two text-based .csv files as input (for examples see https://github.com/patfiaux/RELICS). One file contains the guide information, and the other file contains the observed guide counts in different pools. We have made the public datasets we analyzed available in this format at (https://figshare.com/projects/RELICS_data/65303).

**Simeonov et al. 2017.** CRISPRa screen for *IL2RA* and *CD69* in Jurkat T-cells, using FACS to sort cells based on protein expression. We downloaded the guide counts from the paper supplement (https://www.nature.com/articles/nature23875#supplementary-information) and converted them to RELICS input format. Promoter region was defined as the transcription start site (TSS) +/- 1kb. All guides were used for the analysis.

**Canver et al. 2015.** Cas9 screen targeting DHS sites upstream of *BCL11A* in the HUDEP-2 cell line and mouse *Bcl11a* in the MEL cell line. Canver et al. sorted cells based on HbF expression into a high expression and a low expression pool. Guide counts were obtained from the authors and converted to RELICS input format. We followed the filtering procedure used in the original study and removed guides with a log2 dropout score below -3.

**Fulco et al. 2016.** CRISPRi proliferation screen for *GATA1* and *MYC* in K562 cells. We downloaded the guide counts from the paper supplement (https://science.sciencemag.org/content/suppl/2016/09/28/science.aag2445.DC1) and converted them to RELICS input format. For detecting *MYC* regulatory elements, all guides labelled as 'MYC Tiling' guides as well as 'Non-targeting Negative Controls' guides were used. For detecting *GATA1* regulatory elements, all guides labelled as 'GATA1 Tiling' guides as well as 'Non-targeting Negative Controls' guides were used.

**Diao et al. 2017.** Tiling deletion Cas9 screen (CRESTseq) for *OCT4* expression in the H1 human embryonic stem cell line (hESC). The programmed deletions were tiled across a 2MB region surrounding *OCT4* and Diao et al. sorted cells based on the expression of the *OCT4*-GFP reporter into high and low expression pools. We downloaded the guide counts from the paper supplement (https://www.nature.com/articles/nmeth.4264#supplementary-information). The data can also be obtained from GEO under accession GSE81026. The data was converted to RELICS format and guides targeting GFP were removed from the analysis. We used exon-overlapping guide pairs as positive controls.

**Diao et al. 2016.** Cas9 screen for *OCT4* in the H1 hESC line. Their guides targeted 174 DHS sites around *OCT4*. The authors sorted cells based on the *OCT4*-GFP reporter into a low expression pool. We downloaded guide counts from the paper supplement (https://genome.cshlp.org/content/26/3/397/suppl/DC1) and converted them to RELICS format. The data can also be obtained from GEO under accession GSE75450.

**Klann et al. 2017.** Klann et al. performed a CRISPRi screen for *HBE1* by targeting DHS sites. They also performed multiple screens for *HER2* with either CRISPRi, CRISPRa or Cas9. All 7 data sets were converted to RELICS format. For both CRISPRi and CRISPRa analysis guides targeting the +- 1kb around the transcription start site were used as positive controls; the remaining guides were used as 'background'.

**Gasperini et al. 2017.** Both a dual-Cas9 screen (scanDel) and Cas9 screen for *HPRT1* regulatory elements in the HAP1 cell line by tiling across a 206kb region. Guide counts were obtained from the authors and converted to RELICS format. We removed 'self-self' guides from the paired guide library. Guides overlapping exons were used as positive controls.

**Sanjana et al. 2016.** Cas9 screen for *CUL3*, *NF1* and *NF2* targeting a total of 700kb surrounding the three genes in the A375 cell line. The guide counts were obtained from the authors and converted to RELICS format.

**Rajagopal et al. 2016.** Cas9 screen for *Zfp42*, *Tdgf1*, *Nanog* and *Rpp25* targeting the surrounding regions and a total of 40kb in mESCs. Guide counts were downloaded from the paper supplement (https://www.nature.com/articles/nbt.3468#supplementary-information) and converted to RELICS format.

**H3K27ac data**

Jurkat H3K27ac data from Mansour et al.[34] was downloaded from GEO (GSM1296384). The reads were aligned with bwa-mem[35] using default parameters and filtered for duplicates and low mapping quality (MAPQ < 30) using samtools[36]. The reads were then converted to reads per kilobase per million in 200bp bins using deepTools2[37].

ENCODE H3K27ac ChIP-seq data for the K562 cell line was downloaded in bedgraph format from the UCSC genome browser[38].

**PhastCons data**

Vertebrate PhastCons conservation scores[39] for chromosome 12 were downloaded from the UCSC genome browser (http://hgdownload.cse.ucsc.edu/goldenPath/hg19/phastCons46way/vertebrate/).

**Data availability**

All data were obtained from published papers or simulated as described above. The formatted data that we used for analyses can be downloaded from: https://figshare.com/projects/RELICS_data/65303

**Acknowledgements**

We thank B. Ren, Y. Diao, and R. Fang for helpful discussions about dual-guide regulatory screens and

analysis methods for CRISPR regulatory screens. We thank Y. Zheng and Z. Liu for helpful discussions about CRISPR screens and the RELICS analysis method. We thank A. Sen for guidance in obtaining and processing public datasets. We thank J. Zhou and A. Massarat for testing RELICS and CRSsim. We thank members of the McVicker laboratory for helpful discussions. This research was supported by NIH grant 2R01AI107027-06; by the National Cancer Institute funded Salk Institute Cancer Center (NIH-NCI CCSG: P30 014,195); and by a gift from the Jacobs Foundation. P.C.F. was partially supported by a fellowship from the H.A and Mary K. Chapman Charitable Trust. G.M. is partially supported by the Frederick B. Rentschler Developmental Chair.

**Author contributions**

G.M. and P.C.F. conceived of the idea for RELICS and CRSsim. G.M. supervised the research. G.M. and P.C.F. wrote the manuscript, with input and edits from H.C. P.C.F. implemented RELICS and CRSsim, obtained the public data sets, and analyzed the data. H.C. participated in many helpful discussions about RELICS, CRSsim, and CRISPR regulatory screens.

**Competing interests**

The authors declare no competing interests.

## References


1. Jinek, M. *et al.* A programmable dual-RNA-guided DNA endonuclease in adaptive bacterial immunity. *Science* **337**, 816–821 (2012).
2. Gilbert, L. A. *et al.* Genome-Scale CRISPR-Mediated Control of Gene Repression and Activation. *Cell* **159**, 647–661 (2014).
3. Thakore, P. I. *et al.* Highly specific epigenome editing by CRISPR-Cas9 repressors for silencing of distal regulatory elements. *Nat Methods* **12**, 1143–1149 (2015).
4. Konermann, S. *et al.* Genome-scale transcriptional activation by an engineered CRISPR-Cas9 complex. *Nature* **517**, 583–588 (2015).
5. Diao, Y. *et al.* A tiling-deletion-based genetic screen for cis-regulatory element identification in mammalian cells. *Nat Methods* (2017).
6. Gasperini, M. *et al.* CRISPR/Cas9-Mediated Scanning for Regulatory Elements Required for HPRT1 Expression via Thousands of Large, Programmed Genomic Deletions. *Am. J. Hum. Genet.* **101**, 192–205 (2017).
7. Ernst, J. & Kellis, M. ChromHMM: automating chromatin-state discovery and characterization. *Nat Methods* **9**, 215–216 (2012).
8. Hoffman, M. M. *et al.* Unsupervised pattern discovery in human chromatin structure through genomic segmentation. *Nat Methods* **9**, 473–476 (2012).
9. Buenrostro, J. D., Wu, B., Chang, H. Y. & Greenleaf, W. J. ATAC-seq: A Method for Assaying Chromatin Accessibility Genome-Wide. *Curr Protoc Mol Biol* **109**, 21.29.1–9 (2015).
10. Thurman, R. E. *et al.* The accessible chromatin landscape of the human genome. *Nature* **489**, 75–82 (2012).
11. Wu, H. *et al.* Tissue-specific RNA expression marks distant-acting developmental enhancers. *PLoS Genet* **10**, e1004610 (2014).
12. Chae, M., Danko, C. G. & Kraus, W. L. groHMM: a computational tool for identifying unannotated and cell type-specific transcription units from global run-on sequencing data. *BMC Bioinformatics* **16**, 222 (2015).

13. Kim, T.-K. *et al.* Widespread transcription at neuronal activity-regulated enhancers. *Nature* **465**, 182–187 (2010).

14. Melnikov, A. *et al.* Systematic dissection and optimization of inducible enhancers in human cells using a massively parallel reporter assay. *Nat Biotechnol* **30**, 271–277 (2012).

15. Patwardhan, R. P. *et al.* Massively parallel functional dissection of mammalian enhancers in vivo. *Nat Biotechnol* **30**, 265–270 (2012).

16. Arnold, C. D. *et al.* Genome-wide quantitative enhancer activity maps identified by STARR-seq. *Science* **339**, 1074–1077 (2013).

17. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**, 139–140 (2010).

18. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol* **15**, 550 (2014).

19. Gelman, A. & Hill, J. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. (Cambridge University Press, 2007).

20. van Overbeek, M. *et al.* DNA Repair Profiling Reveals Nonrandom Outcomes at Cas9-Mediated Breaks. *Mol. Cell* **63**, 633–646 (2016).

21. Fulco, C. P. *et al.* Systematic mapping of functional enhancer-promoter connections with CRISPR interference. *Science* **354**, 769–773 (2016).

22. Simeonov, D. R. *et al.* Discovery of stimulation-responsive immune enhancers with CRISPR activation. *Nature* **549**, 111–115 (2017).

23. Hsu, J. Y. *et al.* CRISPR-SURF: discovering regulatory elements by deconvolution of CRISPR tiling screen data. *Nat Methods* **15**, 992–993 (2018).

24. Li, W. *et al.* MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. *Genome Biol* **15**, 554 (2014).

25. Gasperini, M. *et al.* CRISPR/Cas9-Mediated Scanning for Regulatory Elements Required for HPRT1 Expression via Thousands of Large, Programmed Genomic Deletions. *Am J Hum Genet*

26. Canver, M. C. *et al.* BCL11A enhancer dissection by Cas9-mediated in situ saturating mutagenesis. *Nature* **527**, 192–197 (2015).

27. Diao, Y. *et al.* A new class of temporarily phenotypic enhancers identified by CRISPR/Cas9-mediated genetic screening. *Genome Res* **26**, 397–405 (2016).

28. Klann, T. S. *et al.* CRISPR-Cas9 epigenome editing enables high-throughput screening for functional regulatory elements in the human genome. *Nat Biotechnol* (2017).

29. Rajagopal, N. *et al.* High-throughput mapping of regulatory DNA. *Nat Biotechnol* **34**, 167–174 (2016).

30. Laguna, T. *et al.* New insights on the transcriptional regulation of CD69 gene through a potent enhancer located in the conserved non-coding sequence 2. *Mol. Immunol.* **66**, 171–179 (2015).

31. Bauer, D. E. *et al.* An erythroid enhancer of BCL11A subject to genetic variation determines fetal hemoglobin level. *Science* **342**, 253–257 (2013).

32. van de Geijn, B., McVicker, G., Gilad, Y. & Pritchard, J. K. WASP: allele-specific software for robust molecular quantitative trait locus discovery. *Nat Methods* **12**, 1061–1063 (2015).

33. Brooks, M. E. *et al.* glmmTMB balances speed and flexibility among packages for zero-inflated generalized linear mixed modeling. *R J.* **9**, 378–400 (2017).

34. Mansour, M. R. *et al.* Oncogene regulation. An oncogenic super-enhancer formed through somatic mutation of a noncoding intergenic element. *Science* **346**, 1373–1377 (2014).

35. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *ArXiv13033997 Q-Bio* (2013).

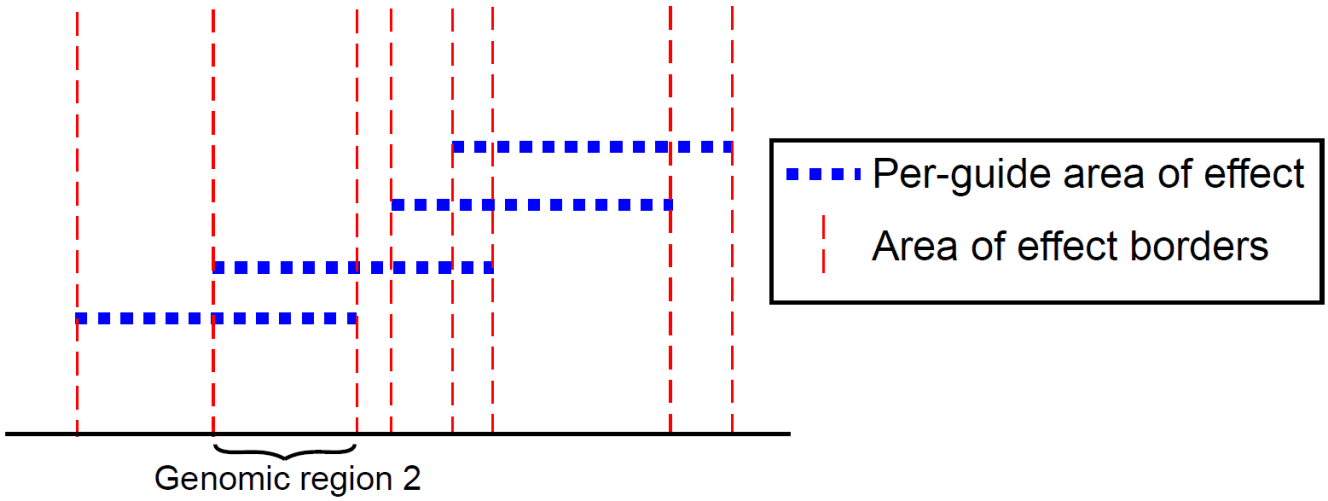
36. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009).

37. Ramírez, F. *et al.* deepTools2: a next generation web server for deep-sequencing data analysis. *Nucleic Acids Res.* **44**, W160–W165 (2016).

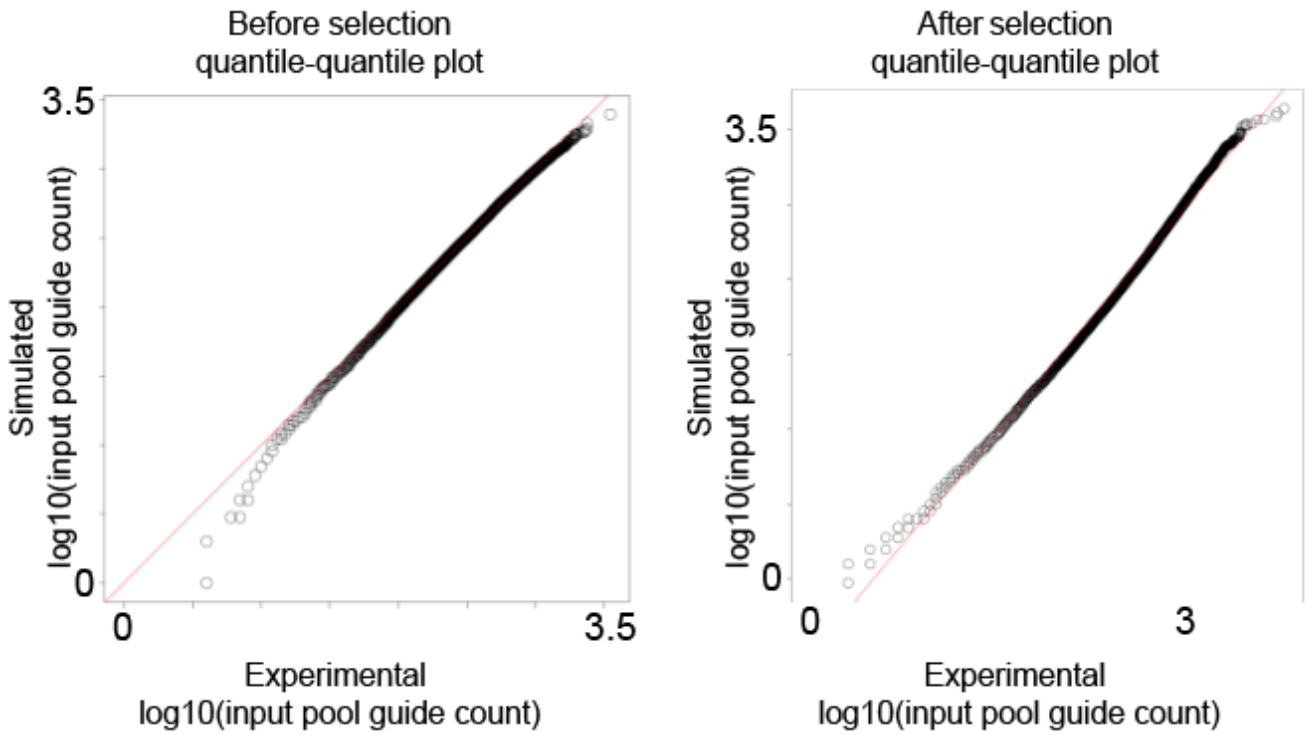
38. Kent, W. J. *et al.* The Human Genome Browser at UCSC. *Genome Res.* **12**, 996–1006 (2002).

39. Siepel, A. *et al.* Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Res.* **15**, 1034–1050 (2005).

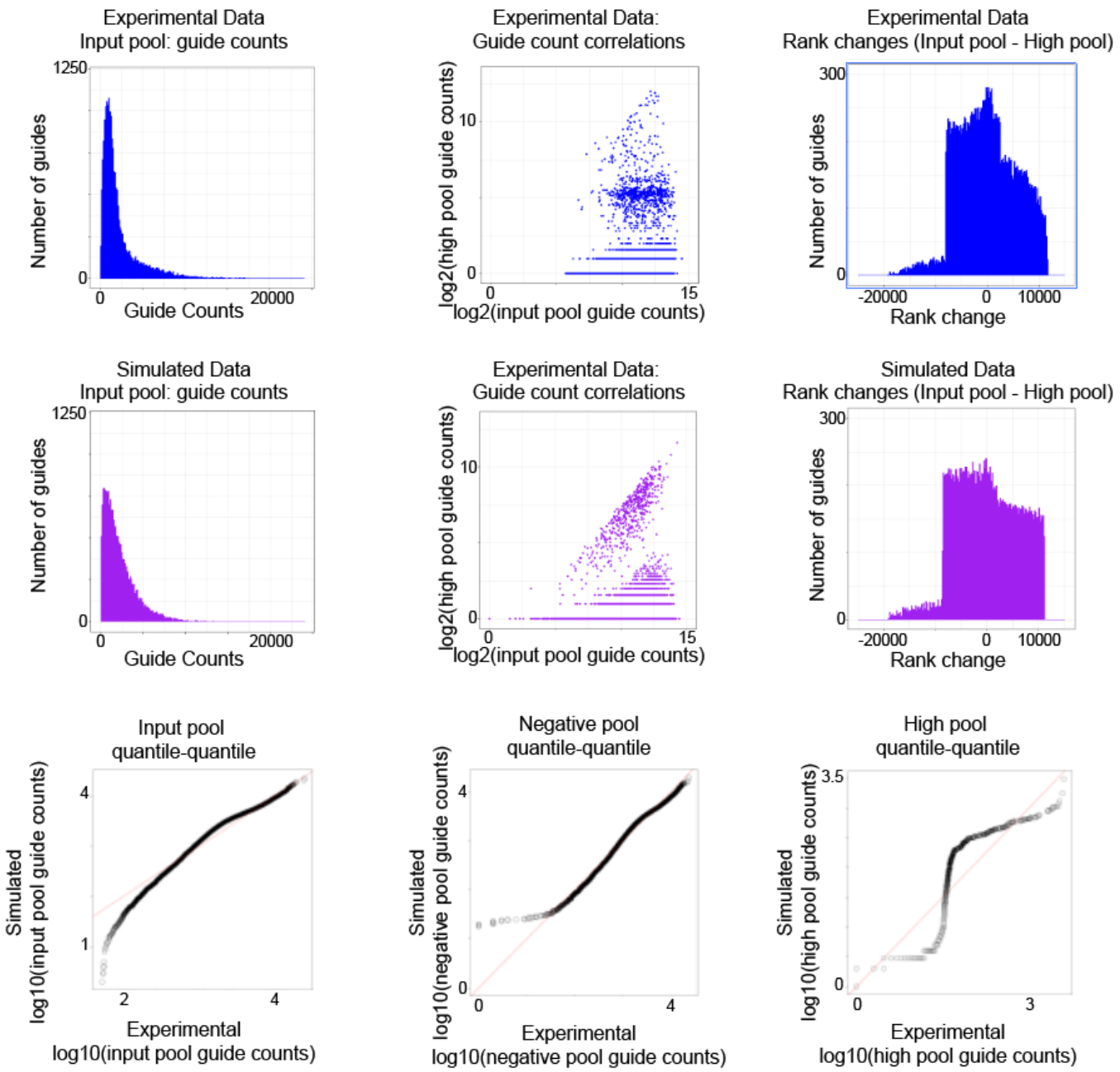
**Supplementary Figures**



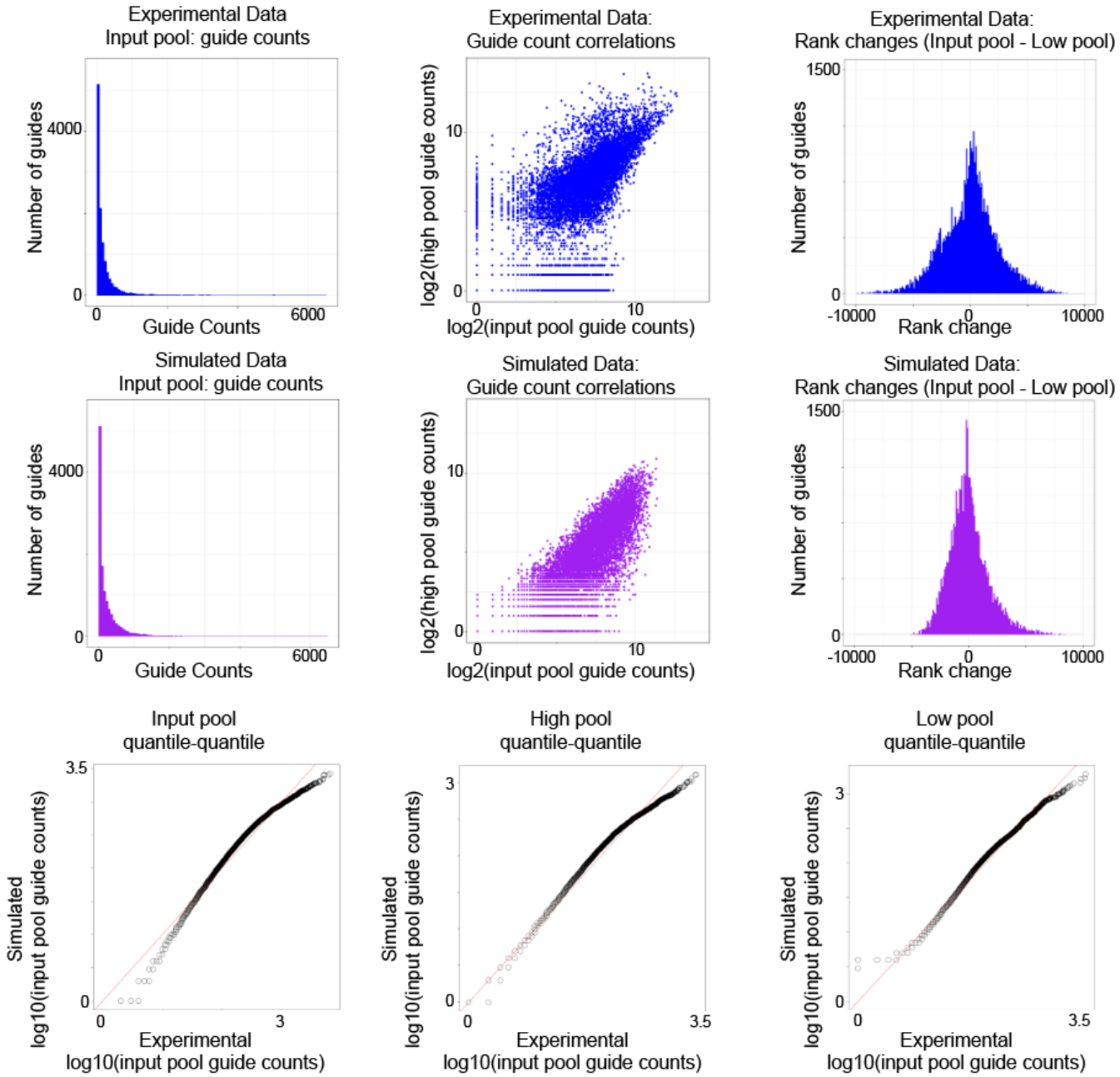

**Supplementary Figure 1** | RELICS genome region scores. RELICS computes scores for individual guides, and combines scores from overlapping guides to compute a genome region score. Guides are considered to overlap when the 'areas of effect' (blue dashed lines) surrounding their target sites overlap. In this example, the score for genomic region 2 is the sum of the scores from the first and second guides, which both overlap this region.



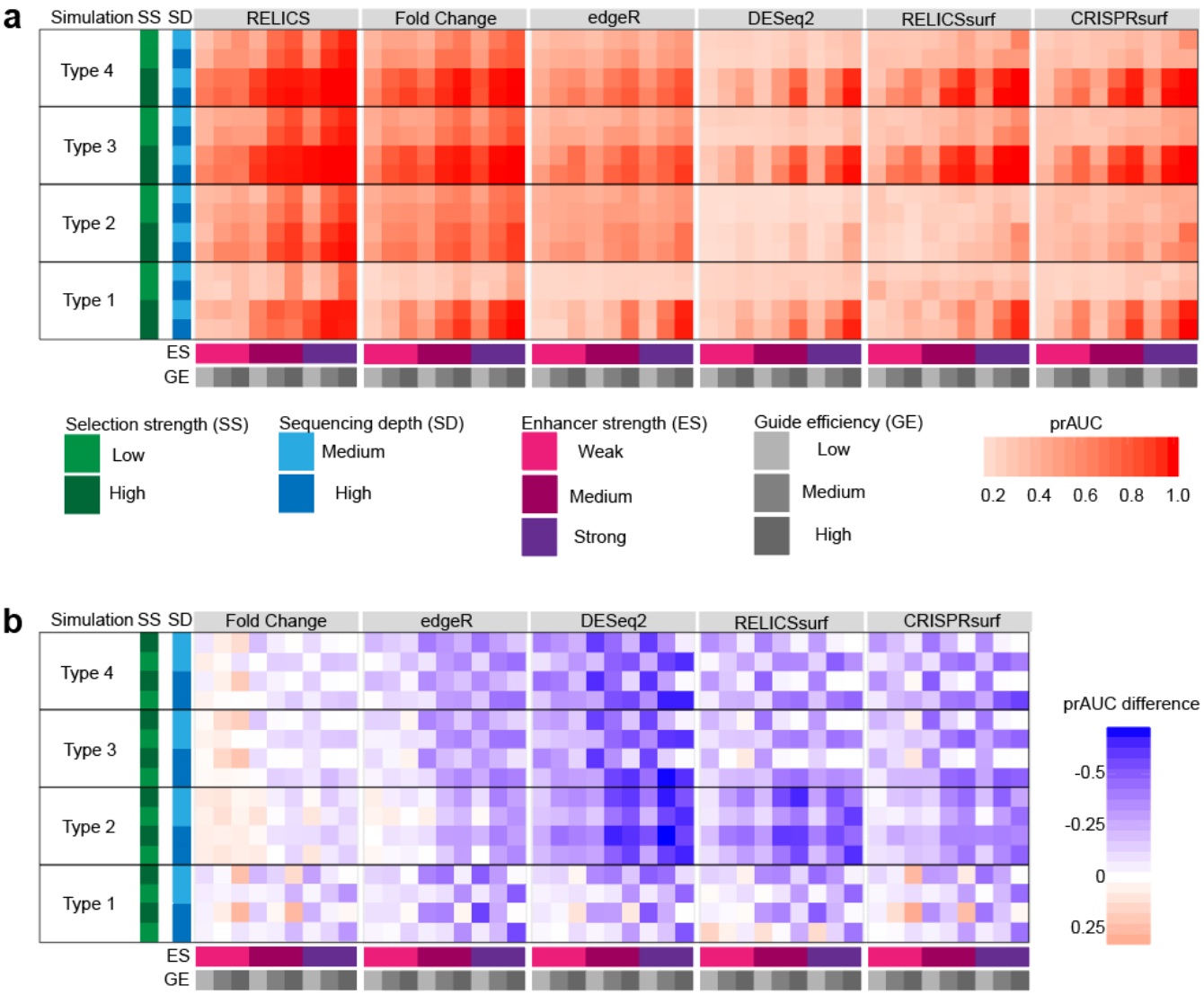

**Supplementary Figure 2 |** Quantile-quantile plots of log guide counts, comparing “Type 1” simulated data to experimental data from Fulco et al. 2016.



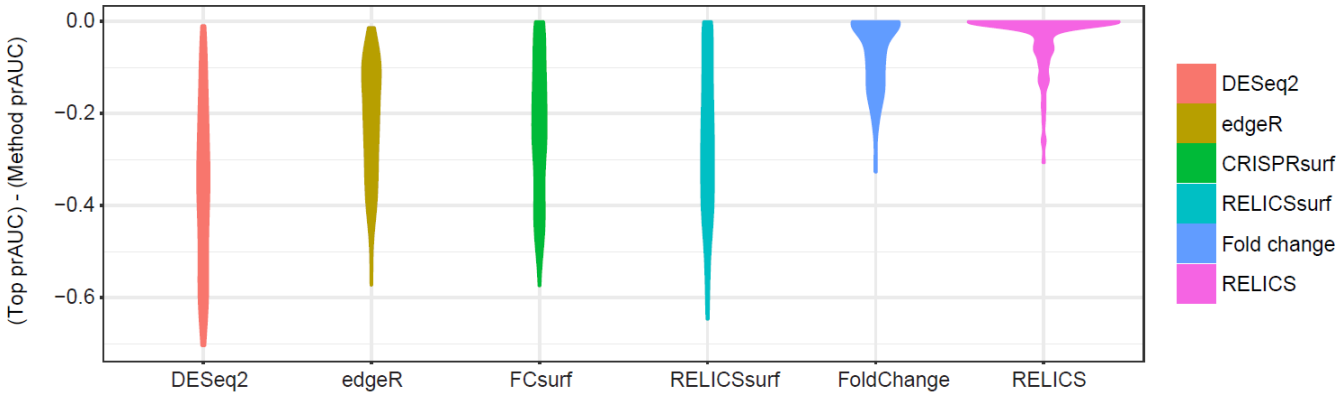

**Supplementary Figure 3** | Guide count distributions; scatter plots of guide counts in input pools vs low expression pools; distributions of rank changes in guide counts; and quantile-quantile plots comparing simulated and experimental guide counts. Experimental data from Simeonov et al. 2017 are shown in blue, while simulated data from a “Type 2” simulations are shown in purple.



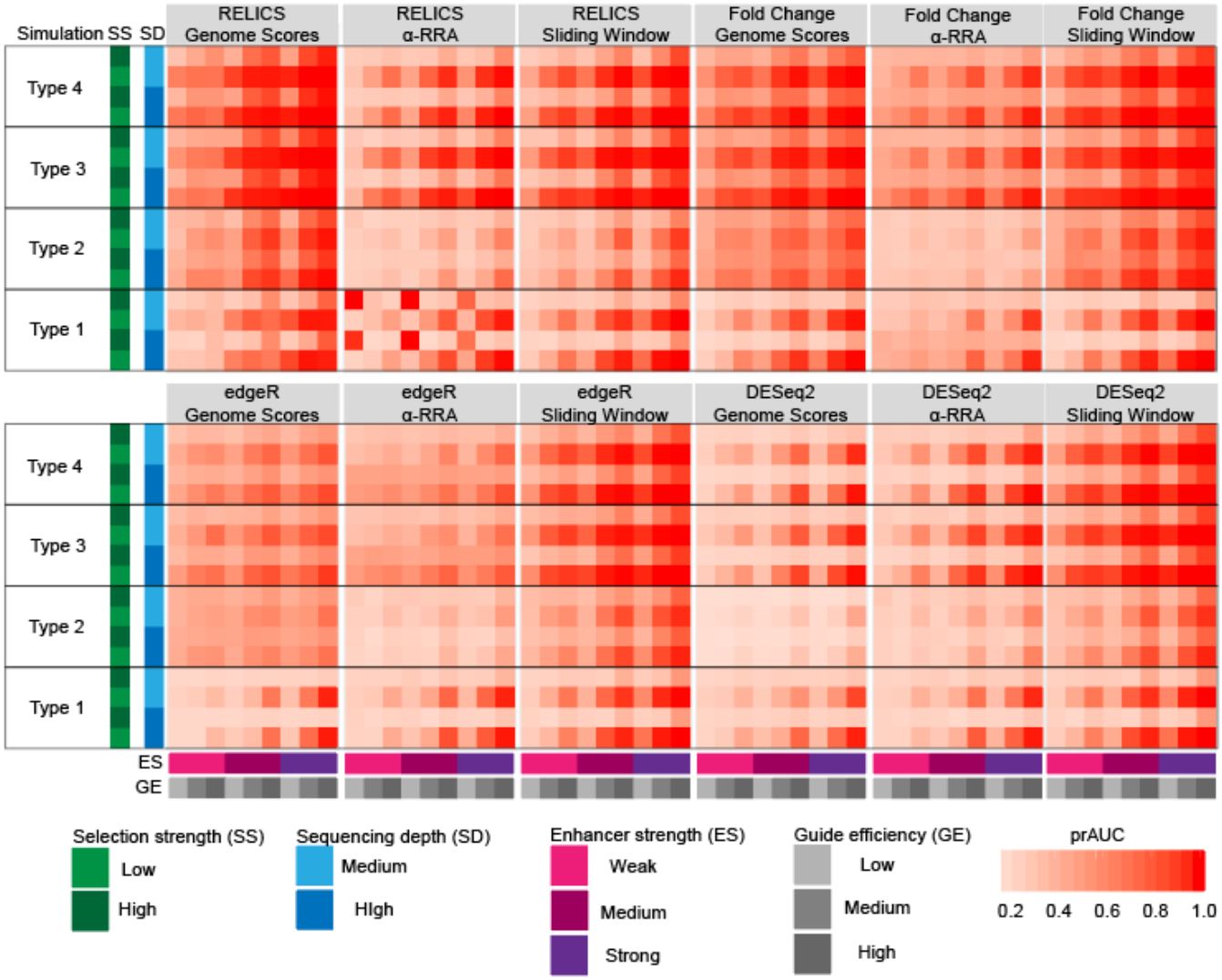

**Supplementary Figure 4** | Guide count distributions; scatter plots of guide counts in input pools vs low expression pools; distributions of rank changes in guide counts; and quantile-quantile plots comparing simulated and experimental guide counts. Experimental data from Diao et al. 2017 are shown in blue, while simulated data from a “Type 3” simulations are shown in purple. For the rank change plots, guides with counts of 0 in both pools were removed.



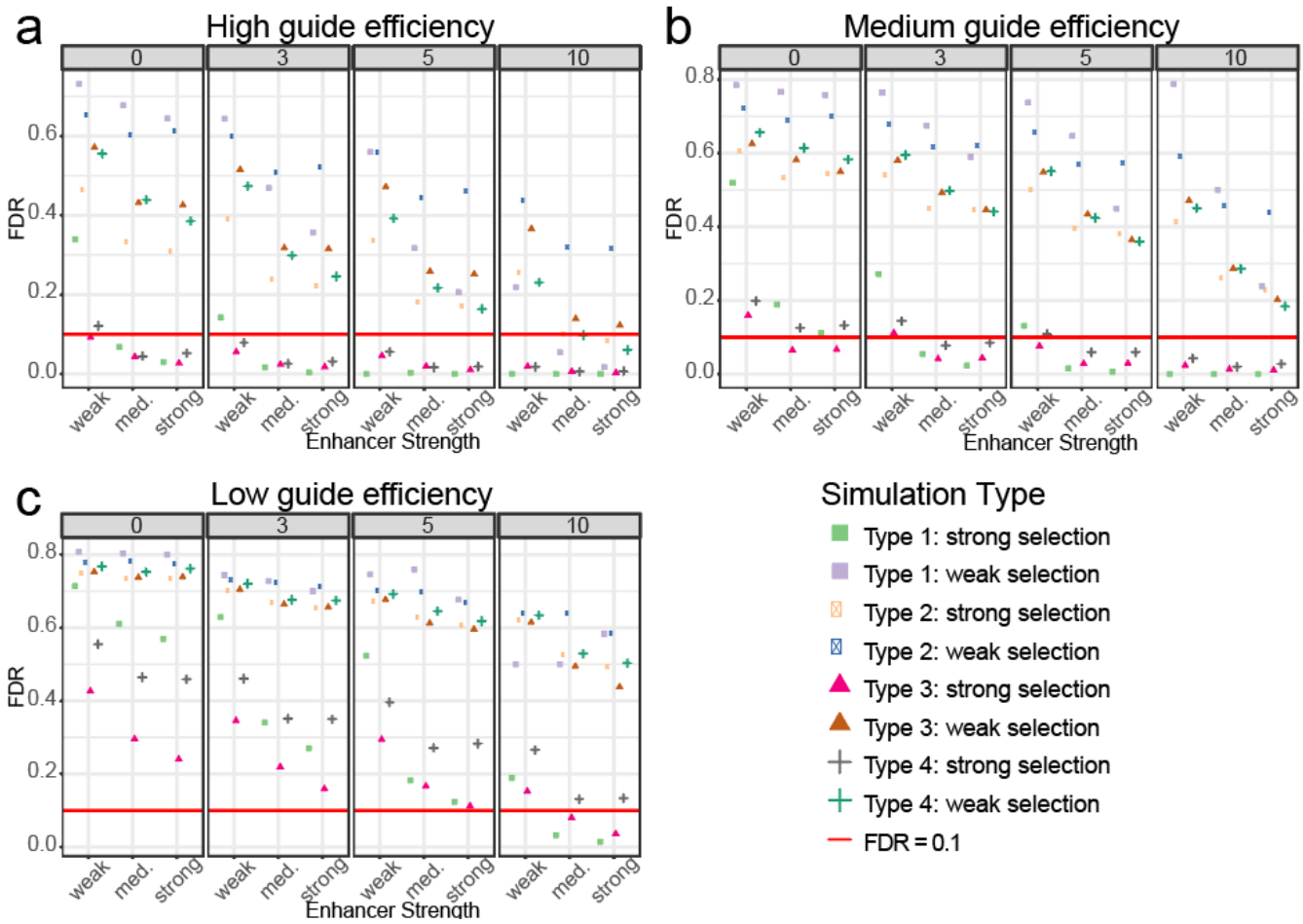

**Supplementary Figure 5 |** Summary of analysis method performance on simulated datasets. **(a)** Heatmap of area under precision recall curves (prAUCs) for six analysis methods, across four simulation types. Type 1 is based on a proliferation screen from Fulco et al. 2016. Type 2 is based on a FACS screen into 4 different pools from Simeonov et al. 2017. Type 3 and 4 are based on a FACS screen into three pools from Diao et al. 2017. In type 3, guides which target a regulatory region get sorted into both the high and the medium pool with an increased probability. In type 4, the sorting probability only changes for the low expression pool. The color intensity of each element corresponds to the median prAUC from 30 simulations for that simulation scenario. **(b)** Difference in prAUC between RELICS and other methods. A negative prAUC (blue) indicates that RELICS has better performance.



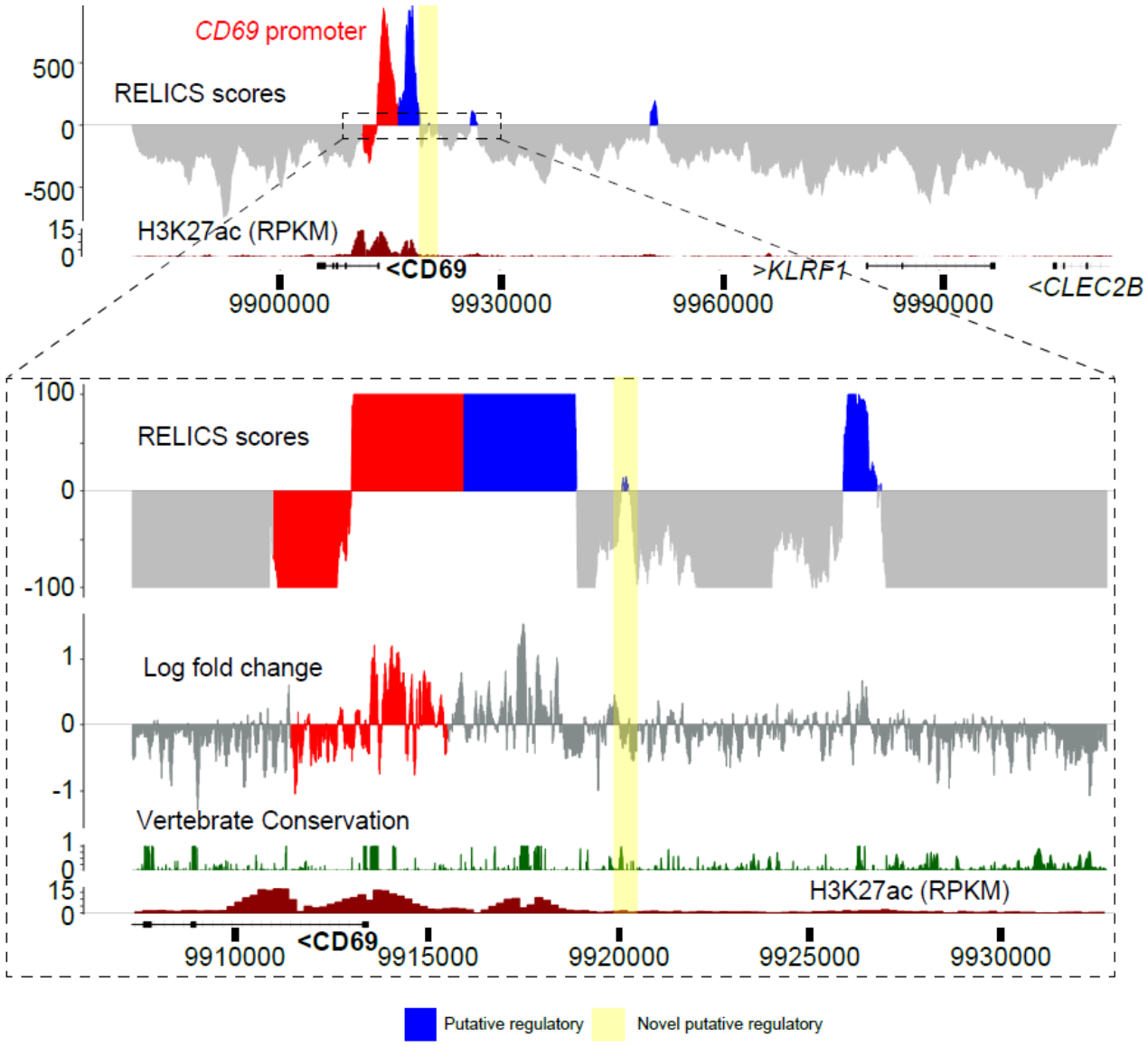

**Supplementary Figure 6 |** Violin plot showing difference in median prAUC from the best-performing method for 144 different simulation scenarios. A difference of 0 means that a method had the best prAUC for a particular scenario.



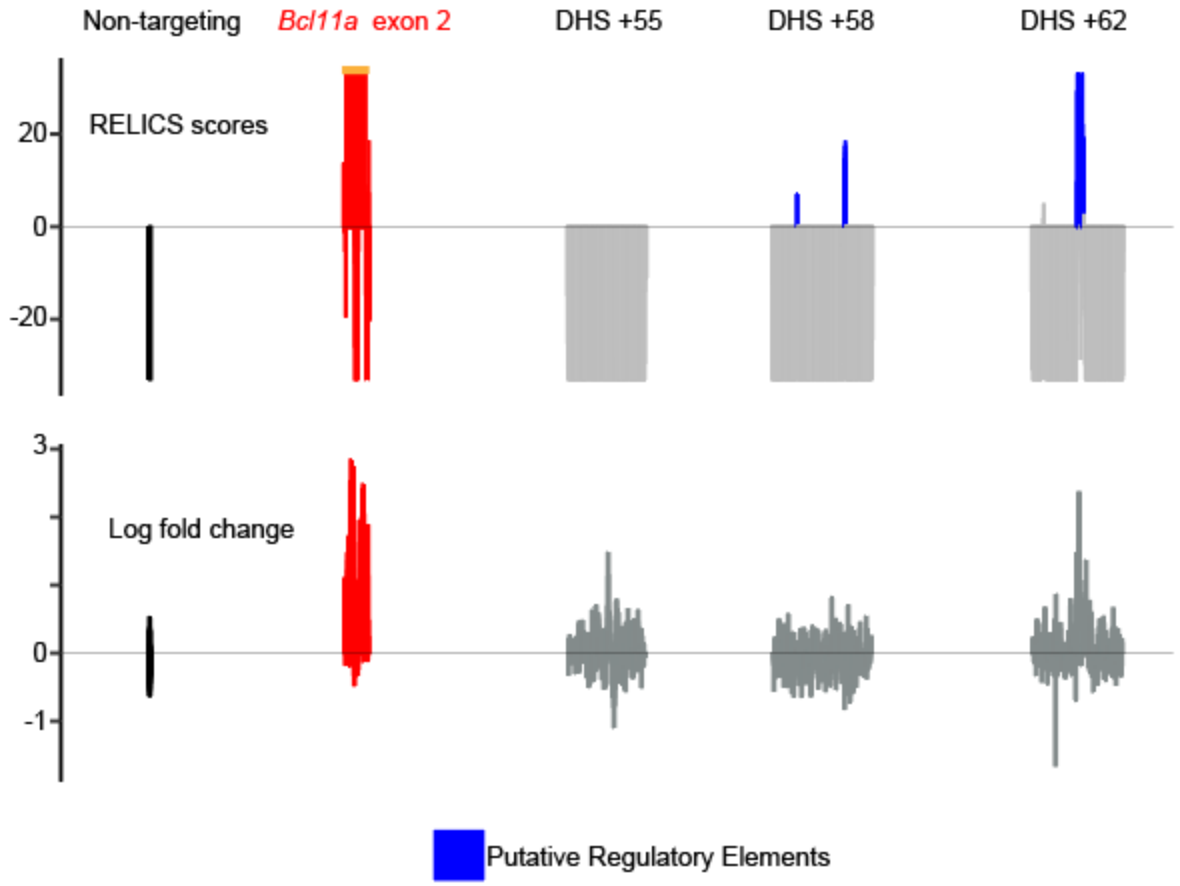

**Supplementary Figure 7 |** Performance of analysis methods using different methods to combine per-guide scores into genome region scores. The 'Genome Scores' method refers to the combining of scores from guides with overlapping areas of effect as described in the Methods and Supplementary Fig. 1. The combination of DESeq2 and $\alpha$-RRA is equivalent to the MAGeCK method of Li et al. 2014, as applied by Diao et al. 2017 to CRISPR regulatory screens. The heatmaps indicate the median area under the precision recall curves from 30 simulations as described in Fig. 3.



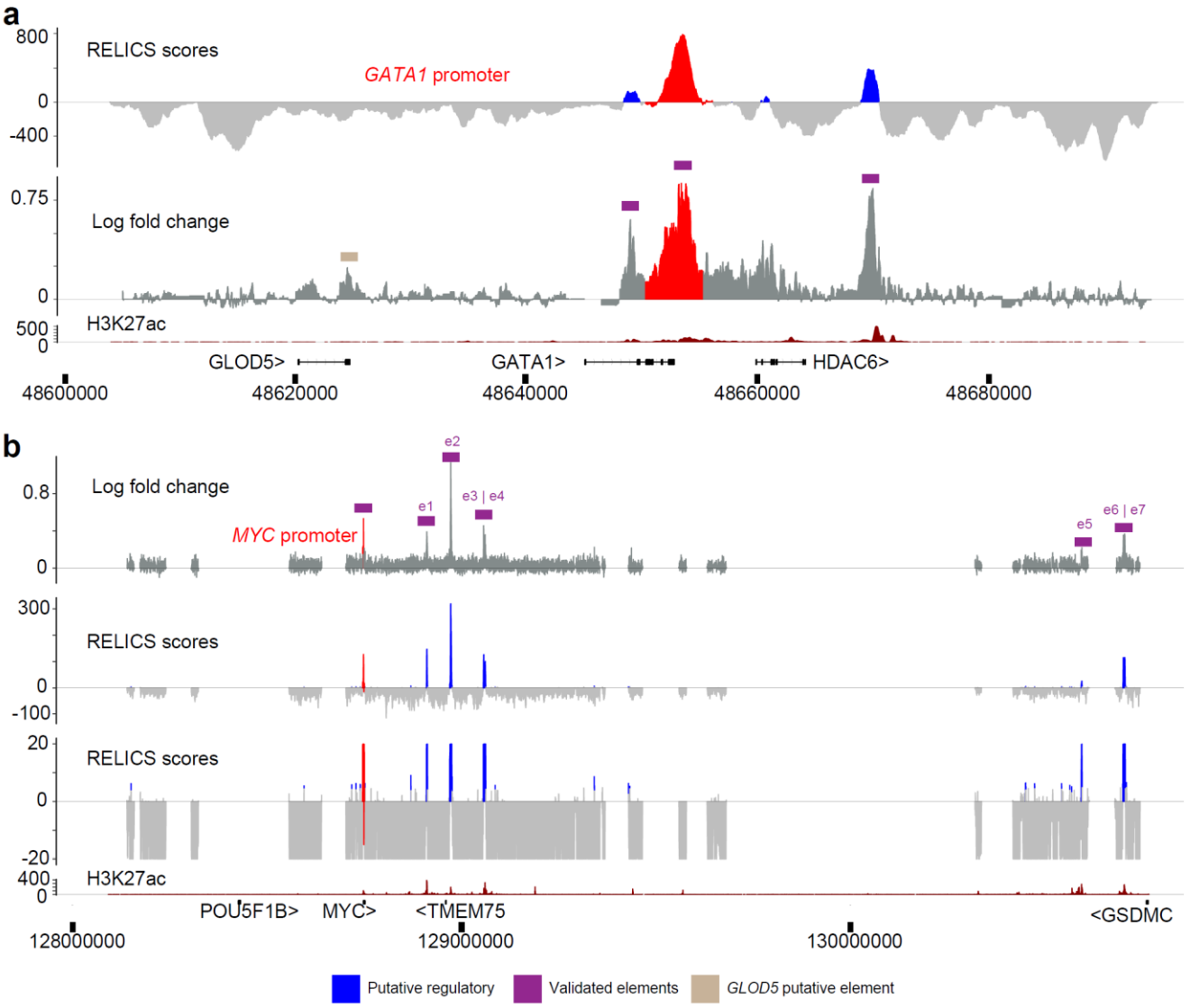

**Supplementary Figure 8** | Estimated False Discovery Rates (FDRs) for RELICS under different significance thresholds**.** Each point gives the median FDR from 30 simulations. Plots are split first by the guide efficiency of the simulations **(a, b, c)** and then by the RELICS score used as a significance threshold (>0, >3, >5, >10).




**Supplementary Figure 9 |** A novel putative regulatory element for *CD69*. In addition to the three putative regulatory elements detected by Simeonov et al. 2017, RELICS identifies a fourth putative regulatory element. Although the signal is not as strong as the other 3 putative regulatory regions, it is highly significant with a RELICS score of 15, and it overlaps an evolutionarily conserved region identified by PhastCons from 46 vertebrate species.




**Supplementary Figure 10** | RELICS analysis of the Canver et al. 2017 Cas9 CRISPR screen for mouse *Bcl11a*. The interrogated regions are DNase hypersensitive sites located at +55, +58 and +62 kilobases from the transcription start site. RELICS identifies two additional signals at site +58 in addition to the reported signal in +62.




**Supplementary Figure 11** | RELICS analysis of the CRISPRi proliferation screens for *GATA1* and *MYC* performed by Fulco et al. 2016. **(a)** RELICS scores and log fold change (as performed in the original paper) for the CRISPRi screen for *GATA1*. Of the 3 elements originally identified, 2 were validated by quantitative PCR in cell lines stably expressing individual sgRNAs and a plasmid-based reporter assay. RELICS identifies both of the validated elements and an additional element near the 3' end of *GATA1*. **(b)** The CRISPRi screen for *MYC* regulatory elements from Fulco et al. 2016 identified 7 putative cis-acting regulatory elements, all of which were validated by quantitative PCR in cell lines stably expressing individual sgRNAs and dCas9-Krab. RELICS identifies the same elements using a stringent threshold of 10. At a more lenient threshold of 5, an additional 18 putative elements are detected.