nature biotechnology

Article

https://doi.org/10.1038/s41587-023-01830-8

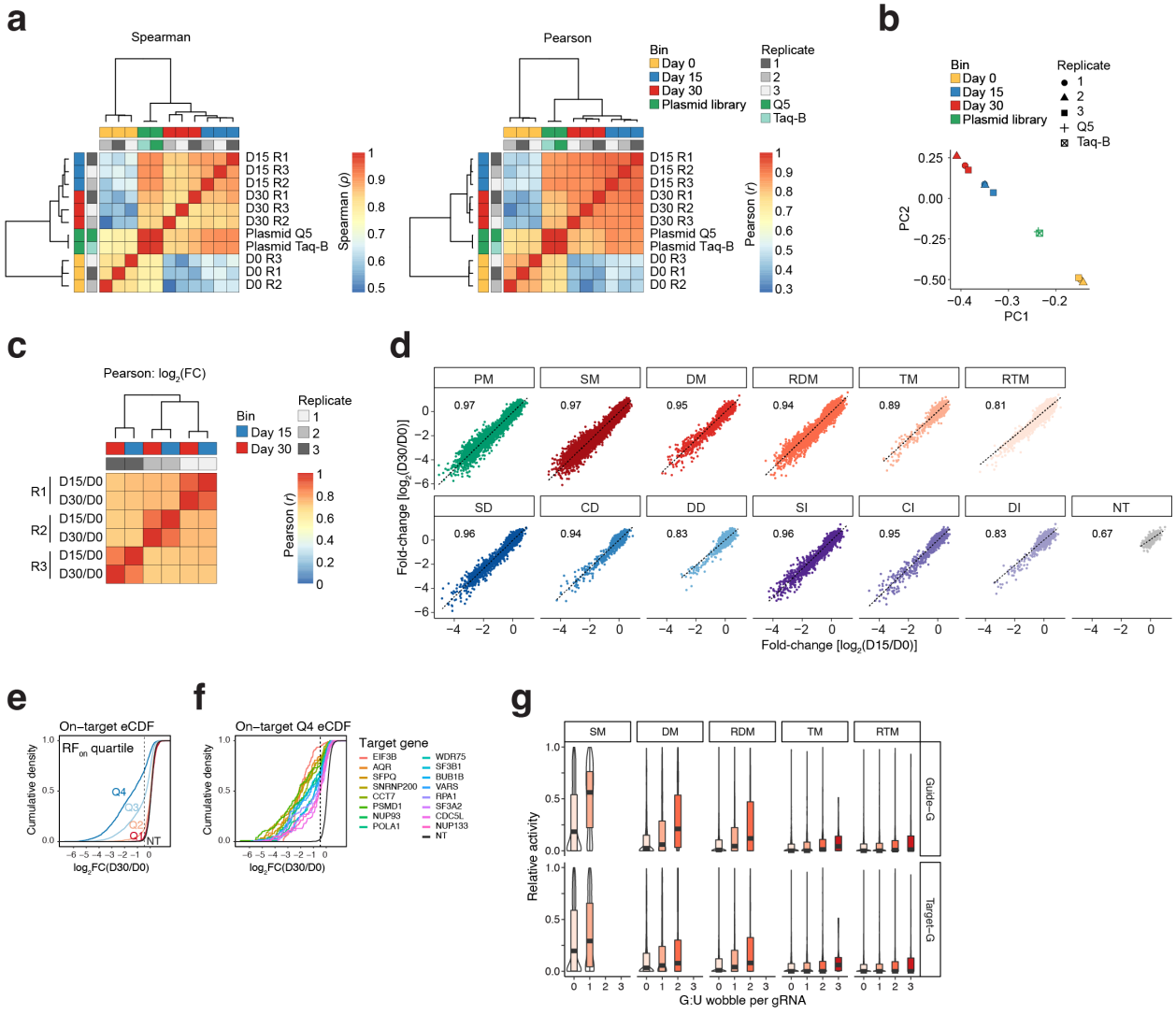
Prediction of on-target and off-target activity of CRISPR–Cas13d guide RNAs using deep learning

In the format provided by the authors and unedited

Supplementary Information

Supplementary Figures

Supplementary Figures



Supplementary Figure 1. Pooled CRISPR-Cas13d screen quality controls.

(a) Spearman (*left*) and Pearson (*right*) correlations coefficients of gRNA counts across 11 libraries after crRNA count processing (normalization, batch-correction, and outlier-masking; n = 119,846 gRNAs).

(b) Principal components (PC) analysis for replicates.

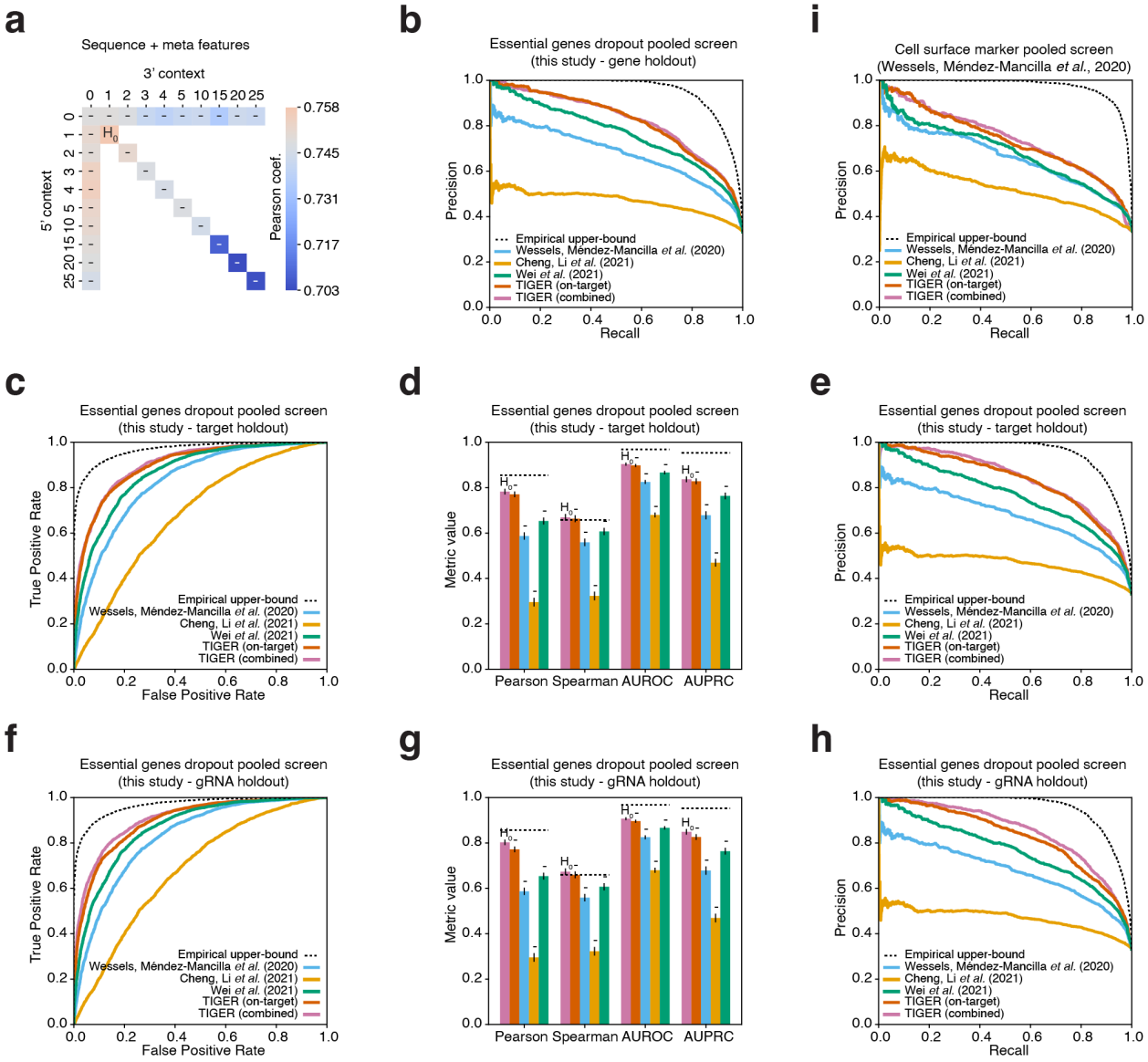
(c) Pearson correlation of log_2FC (Day 15 or Day 30 vs. Day 0) from processed gRNA counts for each biological replicate (n = 119,846 gRNAs).

(d) Pearson correlation of gRNA log_2FC (mean of three replicates) between day 15 and day 30 separated by gRNA type.

(e) Cumulative distribution of PM gRNAs stratified by efficacy quartile as predicted by the random forest (RF_{on}) model [2].

(f) Cumulative distribution of the most active quartile (Q4) PM gRNAs separated by target gene. Data shown in *e* and *f* corresponds to *Figure 1e, f*. The vertical line indicates $\log_2$(fold-change) = -0.5.

(g) Relative targeting efficacy (percent of parental PM gRNA $\log_2$FC) for gRNAs contain 1-3 G-U wobble base pairs compared to unpaired mismatches for all mismatched gRNA types (n = 388 reference PM gRNA with $\log_2$FC < -0.5). Boxes indicate the median and IQRs, with whiskers indicating 1.5× the IQR or the most extreme data point outside the 1.5-fold IQR.



Supplementary Figure 2. A deep learning model to predict optimal Cas13d guide RNAs.

(a) The effect of including sequence context left (5'), right (3') or combined (5' and 3') to the 23nt gRNA target site on the correlation of predictions aggregated from the held-out target sites (n = 10 folds). Analysis was done using a model that includes non-sequence features. H_0 denotes the best performing condition and all differences between other conditions and H_0 are significant (p < 0.05, Steiger's test [1]).

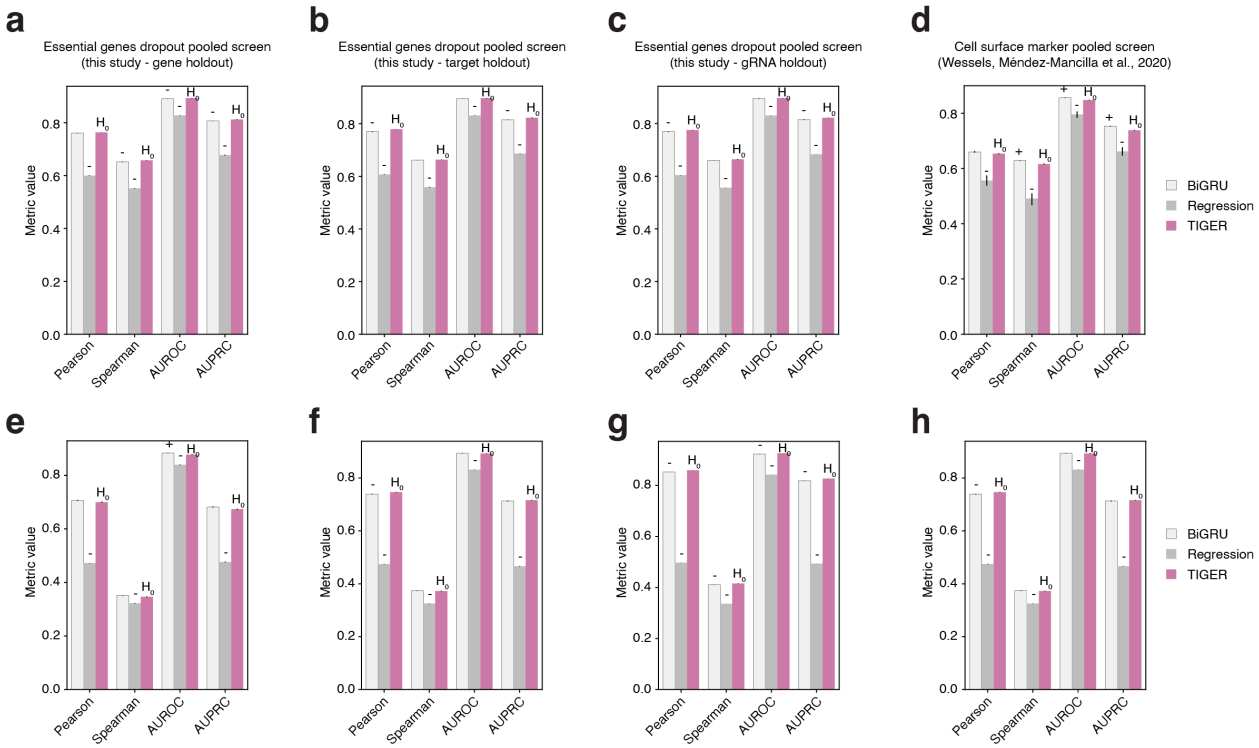
(b) Precision recall curves for the predictions aggregated from the target gene holdouts (n = 16 folds) on our survival screen.

(c - e) Receiver-operator characteristic (ROC) curve (c), summary statistics (d) and precision recall curves (e) for the predictions aggregated from held-out target sites (n = 10 folds) from the survival screen of essential genes.

(f - h) Receiver-operator characteristic (ROC) curve (*f*), summary statistics (*g*) and precision recall curves (*h*) for the predictions aggregated from held-out gRNAs (n = 10 folds) from the survival screen of essential genes.

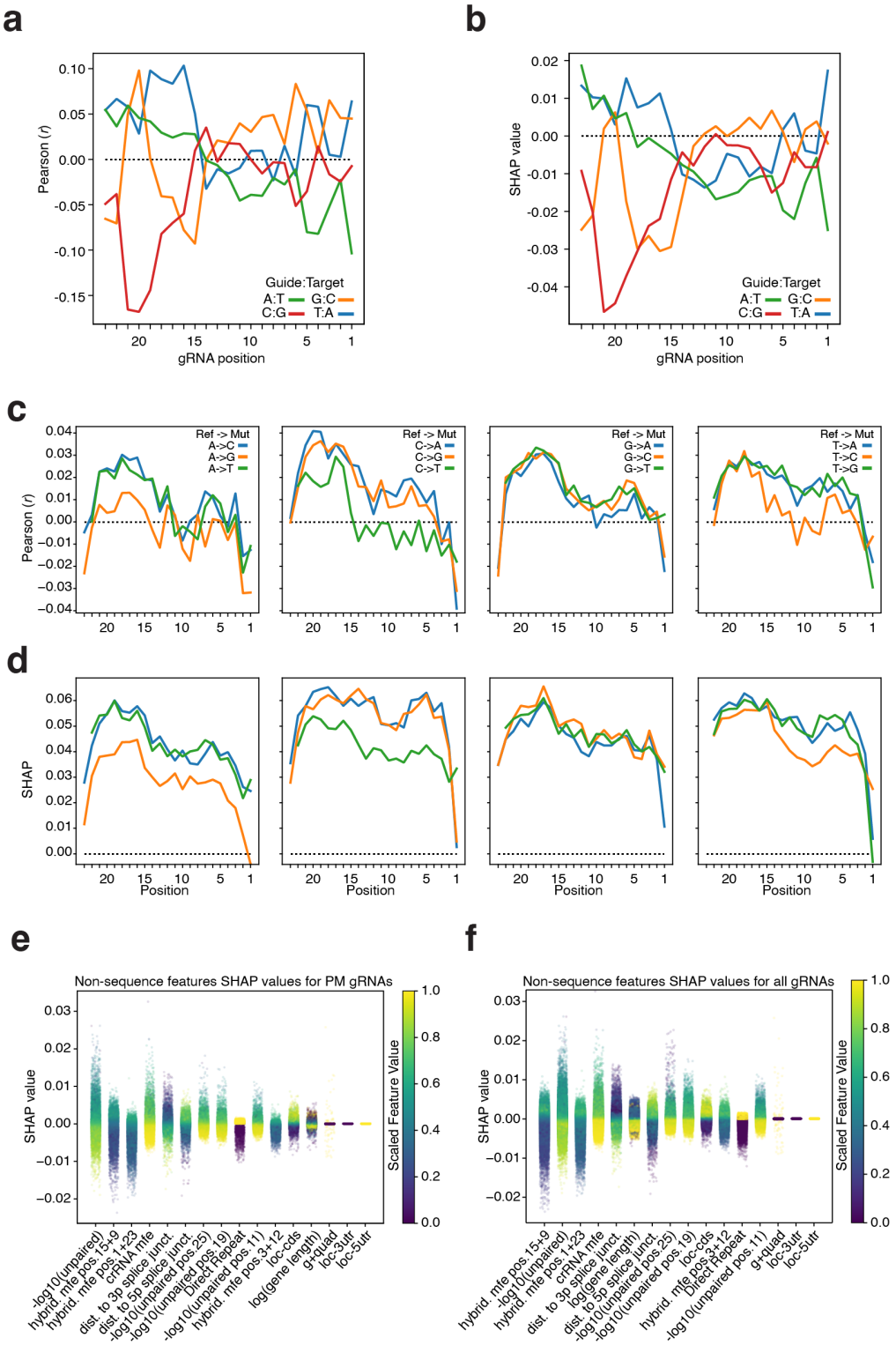
(i) Precision recall curves for all gRNAs from a previously published screen using flow cytometry of cell surface proteins (ref. [2]).

Panels *d* and *g* employ a Steiger's test [1] for Pearson and Spearman comparisons, DeLong's test [3,4] for AUROC comparisons and a bootstrapped Kolmogorov-Smirnov test [5] for AUPRC comparisons. Performance values denote aggregate performance over cross-validation folds. Error bars denote ±2 standard errors.



Supplementary Figure 3. TIGER performance evaluation to alternative models.

On-target performance for TIGER and alternative model architectures when trained and tested on just PM guides **(a - d)** and on PM gRNAs in combination with mismatch gRNA variants **(e - f)**. We mark statistically significant differences with a '+' if an alternative model performs better than TIGER or a '-' if a model performs worse TIGER. Unmarked bars denote nonsignificant differences from TIGER. Pearson and Spearman comparisons employ a Steiger's test [1]. AUROC comparisons employ a DeLong's test [3,4]. AUPRC comparisons employ a bootstrapped Kolmogorov-Smirnov test [5]. Performance values denote aggregate performance over cross-validation folds. Error bars denote ±2 standard errors.



Supplementary Figure 4. TIGER SHapley Additive exPlanations (SHAP) analysis.

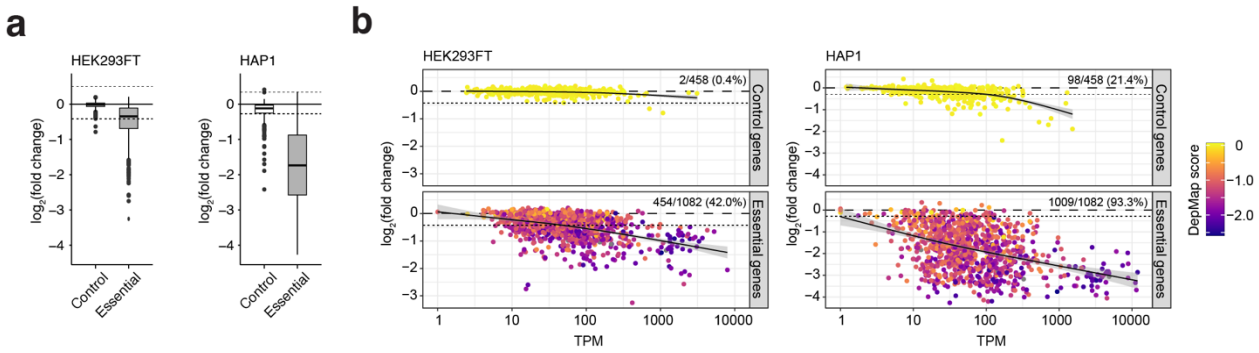
(a) Per-nucleotide Pearson correlation between $\log_2$FC and the binary outcome of a guide-target base pair being equal (or not) to the four possible perfect matches in the legend.

(b) SHAP values for perfect match guide-target base pairs.

(c) Per nucleotide Pearson correlation between $\log_2$FC and the binary outcome of a guide mutation being equal (or not) to the three possible departures from a perfect match.

(d) SHAP values for each possible guide mutation.

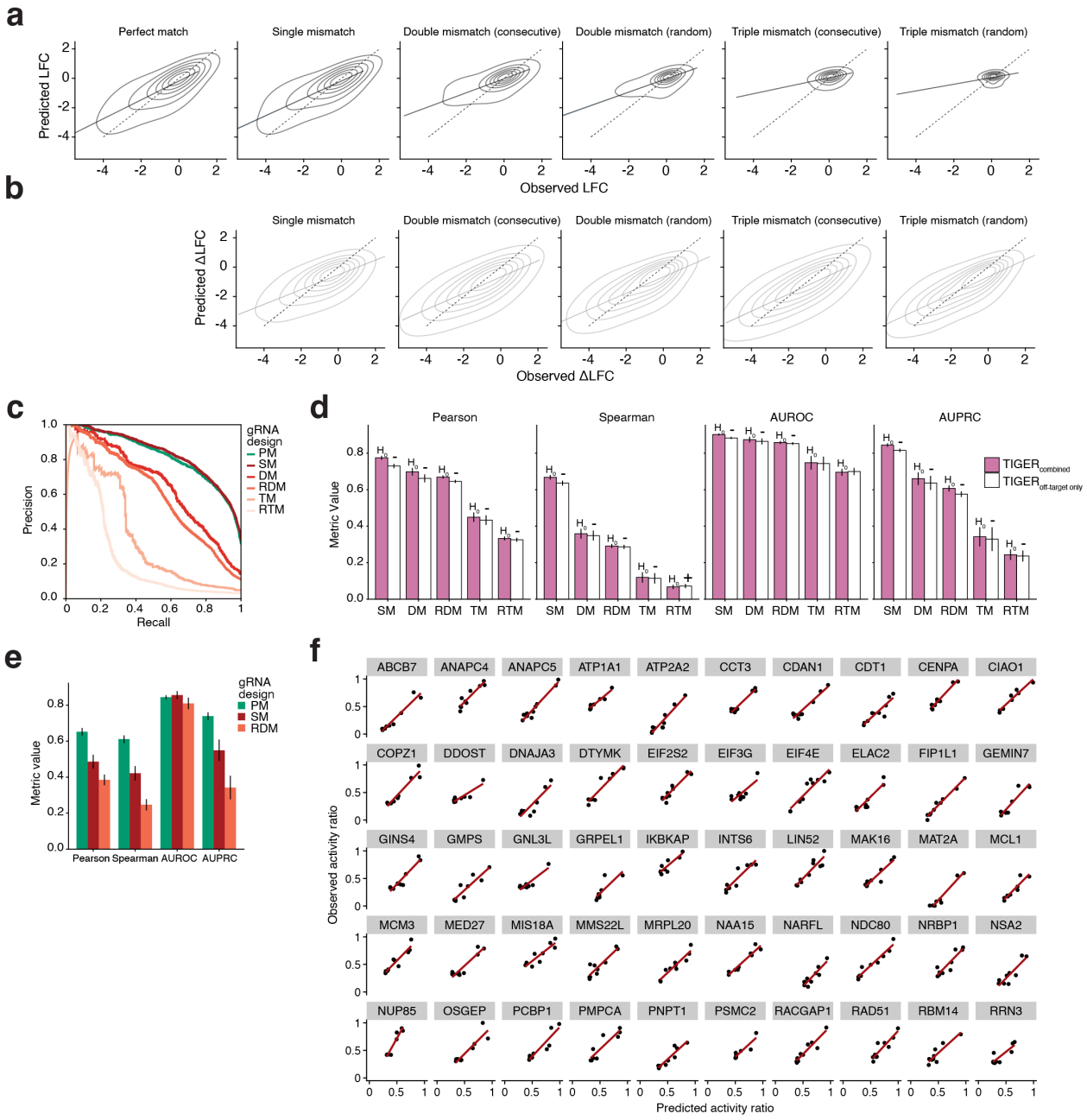
(e - f) SHAP analysis evaluating the non-sequence features provided in addition to the sequence information in the context the $\text{TIGER}_{\text{on-target}}$ model trained on PM gRNAs (*e*) and mismatched gRNAs (*f*). Non-sequence features were scaled to [0,1] to equalize feature magnitudes with respect to the one-hot encoded sequence.



Supplementary Figure 5. TIGER on-target performance identifying essential genes.

(a) Relative gene depletion for two independent Cas13d pooled screens (HEK293FT, HAP1 cells). The box plot captures the mean depletion as an average across gRNAs (n = 8 per gene) and replicates, comparing gRNA abundance at Day 14 to Day 0 (n = 1,082 essential genes and 458 control genes). Dashed lines indicate the 1st and 99th percentiles of the non-targeting gRNA distribution. Boxes indicate the median and IQRs, with whiskers indicating 1.5× the IQR or the most extreme data point outside the 1.5-fold IQR.

(b) Gene depletion as a function of target gene expression. Same data as in (*a*) separated into essential gene and control gene groups. Each data point is a gene, colored by the mean DepMap score (CRISPR loss-of-function screens) across all available cell lines. A gene is classified as depleted, if more depleted than the 99th percentile (*FDR* < 0.01) of the non-targeting gRNA distribution.



Supplementary Figure 6. TIGER performance evaluation on mismatch gRNA variant data.

(a) Kernel density estimate for observed and TIGER-predicted gRNA abundance by gRNA design type.

(b) Kernel density estimate for change in observed and TIGER-predicted gRNA abundance by gRNA design type. The change in gRNA abundance is defined as the difference in fold-change for a particular gRNA with mismatches and its cognate perfect match (PM) gRNA.

(c) Precision recall curves by guide mismatch type for the aggregated predictions of held-out target sites (n = 10 cross-validation folds).

(d) Aggregate correlation (Pearson and Spearman) and AUROC and AUPRC for each gRNA design type from 10-fold target-site cross-validation comparing TIGER models trained on PM and

SM gRNAs (combined) or only SM gRNAs (off-target only). We mark statistically significant differences with a '+' if an alternative model performs better or a '-' if a model performs worse. Unmarked bars denote nonsignificant differences. Pearson and Spearman comparisons employ a Steiger's test [1]. AUROC comparisons employ a DeLong's test [3,4]. AUPRC comparisons employ a bootstrapped Kolmogorov-Smirnov test [5]. Performance values denote aggregate performance over cross-validation folds. Error bars denote ±2 standard errors.

(e) Aggregate correlation (Pearson and Spearman) and AUROC and AUPRC for PM and mismatched gRNA variants from a previously published screen using flow cytometry of cell surface proteins[2] that was independent of the experiments used to train the model. Bars show aggregate value across three screens in ref. [2].

(f) Predicted and observed gRNA activity ratios of all 10 SM gRNA variants at 50 example target sites (50 genes).

Supplementary References

1. Steiger, J. H. Tests for comparing elements of a correlation matrix. *Psychol. Bull.* **87**, 245–251 (1980).
2. Wessels, H. H. *et al.* Massively parallel Cas13 screens reveal principles for guide RNA design. *Nat. Biotechnol.* **38**, 722–727 (2020).
3. DeLong, E. R., DeLong, D. M. & Clarke-Pearson, D. L. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics* **44**, 837–845 (1988).
4. Sun, X. & Xu, W. Fast implementation of DeLong's algorithm for comparing the areas under correlated receiver operating characteristic curves. *IEEE Signal Process. Lett.* **21**, 1389–1393 (2014).
5. Massey, F. J. J. The Kolmogorov-Smirnov Test for Goodness of Fit. *J. Am. Stat. Assoc.* **46**, 68–78 (1951).

Supplementary Note 1

Validation Folding Strategies

We consider three validation folding strategies, which fig. 1 summarizes graphically. The first (algorithm 1) creates folds along gene boundaries. In particular, it holds out all guides targeting a single gene for testing while training on guides for the remaining fifteen genes in our screen. This strategy is the most difficult to perform well on as it requires transcriptome-wide generalization. The second (algorithm 2) randomly assigns target sites into K (10 in our case) approximately equal sized folds. Folding by target site places a target sequence and all its guides (matched and mismatched) into the same fold, thus ensuring a target sequence cannot simultaneously appear both in training and testing sets; enforcing this is important since active PM and SM guides that share a target site can convey significant information about one another. For example, training on an active PM guide and its associated target sequence and testing on a highly active SM guide targeting the same sequence is akin to testing on the training set (i.e. cheating). For this reason, the final validation strategy (algorithm 3) is the easiest to perform well on because it does not make such an enforcement. Here, guides are randomly split into K folds regardless of the target sequence.

Figure 1: Graphical summary of validation folding strategies

Algorithm 1 Gene Folds

Require: Data set: $\mathcal{D} = \{(\text{gene, target sequence, guide sequence, LFC})\}_{i=1}^{N}$

$\mathcal{G} \leftarrow \text{Unique}(\mathcal{D}.\text{gene})$ ▷ Get set of unique genes

$\mathcal{F} \leftarrow \{\emptyset\}$ ▷ Initialize folds

for $g \in \mathcal{G}$ **do** ▷ For each unique gene

$\mathcal{F} \leftarrow \mathcal{F} \cup \{\{(\text{gene, target, guide, LFC}) \in \mathcal{D} : \text{gene} = g\}\}$ ▷ Append data for this gene as a new fold

end for

return $\mathcal{F}$

Algorithm 2 Target Folds

Require: Number of folds: K
Require: Data set: $\mathcal{D} = \{(\text{gene, target sequence, guide sequence, LFC})\}_{i=1}^{N}$
  $\mathcal{F} \leftarrow \{\emptyset\}$ ▷ Initialize folds
  while $|\mathcal{D}| > 0$ **do** ▷ Loop until no data remains
    $\mathcal{S} \leftarrow \emptyset$ ▷ Initialize a new fold
    while $|\mathcal{S}| < \frac{N}{K}$ **do** ▷ Loop until new fold has $\geq K$ tuples
      $(g, t, \_, \_) \leftarrow \text{Sample}((\text{gene, target, guide, LFC}) \in \mathcal{D})$ ▷ Sample a target site
      $\mathcal{T} \leftarrow \{(\text{gene, target, guide, LFC}) \in \mathcal{D} : \text{gene} = g \wedge \text{target} = t\}$ ▷ Get all guides for sampled target site
      $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{T}$ ▷ Append samples as a new fold
      $\mathcal{D} \leftarrow \mathcal{D} \setminus T$ ▷ Remove samples from data set to prevent reusage
    end while
    $\mathcal{F} \leftarrow \mathcal{F} \cup \{\mathcal{S}\}$ ▷ Append the newly completed fold
  end while
  return $\mathcal{F}$

Algorithm 3 Guide folds

Require: Number of folds: K
Require: Data set: $\mathcal{D} = \{(\text{gene, target sequence, guide sequence, LFC})\}_{i=1}^{N}$
  $\mathcal{F} \leftarrow \{\emptyset\}$ ▷ Initialize folds
  while $|\mathcal{D}| > \frac{N}{K}$ **do** ▷ Loop $K-1$ times
    $\mathcal{S} \leftarrow \text{SampleWithoutReplacement}(\mathcal{D}, \text{numSamples} = \lceil \frac{N}{K} \rceil)$ ▷ Randomly sample tuples from data set
    $\mathcal{F} \leftarrow \mathcal{F} \cup \{\mathcal{S}\}$ ▷ Append samples as a new fold
    $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathcal{S}$ ▷ Remove samples from data set to prevent reusage
  end while
  return $\mathcal{F}$

Measuring Validation Performance

After assigning folds (algorithms 1 to 3), we aggregate predictions from each held-out fold during cross-validation such that we have a held-out prediction for every data point (Höfling and Tibshirani, 2008). It is upon these N aggregated predictions we compute performance metrics. Please see algorithm 4 for details.

Algorithm 4 Performance Metrics

Require: Data set: $\mathcal{D} = \{(\text{gene, target sequence, guide sequence, LFC})\}_{i=1}^{N}$
Require: Folds: $\mathcal{F} = \{\mathcal{F}_k\}_{k=1}^{K}$
Ensure: $\bigcap_{k=1}^{K} \mathcal{F}_k = \emptyset$
Ensure: $\bigcup_{k=1}^{K} \mathcal{F}_k = \mathcal{D}$
  $\mathcal{P} \leftarrow \emptyset$ ▷ Initialize set of predictions
  for $\mathcal{F}_k \in \mathcal{F}$ **do** ▷ For each fold
    $\mathcal{T} \leftarrow \bigcap(\mathcal{F} \setminus \{\mathcal{F}_k\})$ ▷ Training set is all data except the current fold
    model $\leftarrow$ trainModel($\mathcal{T}$) ▷ Train model on training set
    $\mathcal{P}_k \leftarrow$ model.predict($\mathcal{F}_k$) ▷ Predict on held out fold
    $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{P}_k$ ▷ Append this fold's predictions
  end for
  Pearson, Spearman, AUROC, AUPRC $\leftarrow$ alignAndComputeMetrics($\mathcal{D}, \mathcal{P}$)
  return Pearson, Spearman, AUROC, AUPRC

When we consider a technical holdout (e.g. Wessels et al. (2020)), then we train on our screen and test on the other screen's data. Please see algorithm 5 for details.

Algorithm 5 Technical Holdout Metrics

Require: Training data set: $\mathcal{D} = \{(\text{gene, target sequence, guide sequence, LFC})\}_{i=1}^{N}$
Require: Technical holdout: $\mathcal{H} = \{(\text{gene, target sequence, guide sequence, LFC})\}_{i=1}^{M}$
Ensure: $\mathcal{D} \cap \mathcal{H} = \emptyset$
  model $\leftarrow$ trainModel($\mathcal{D}$) ▷ Train model on training set
  $\mathcal{P} \leftarrow$ model.predict($\mathcal{H}$) ▷ Predict on technical holdout
  Pearson, Spearman, AUROC, AUPRC $\leftarrow$ alignAndComputeMetrics($\mathcal{D}, \mathcal{P}$)
  return Pearson, Spearman, AUROC, AUPRC

In our report, we set model performance expectations by computing a pseudo-upperbound by using the data to predict itself. Per algorithm 6, we predict the mean of two replicates from the third.

Algorithm 6 Replicate Metrics

Require: Data set: $\mathcal{D} = \{(\text{gene, target sequence, guide sequence, } \text{LFC}_1, \text{LFC}_2, \text{LFC}_3)\}_{i=1}^{N}$
  $\mathcal{T} \leftarrow \emptyset$ ▷ Initialize set of targets
  $\mathcal{P} \leftarrow \emptyset$ ▷ Initialize set of predictions
  for $(\_, \_, \_, \text{LFC}_1, \text{LFC}_2, \text{LFC}_3) \in \mathcal{D}$ **do** ▷ For each element in the data
    for $i \in \{1, 2, 3\}$ **do** ▷ For each replicate
      $\mathcal{T} \leftarrow \mathcal{T} \cup$ mean($\{\text{LFC}_1, \text{LFC}_2, \text{LFC}_3\} \setminus \text{LFC}_i$) ▷ Target is mean of non-held out replicates
      $\mathcal{P} \leftarrow \mathcal{P} \cup \text{LFC}_i$ ▷ Predictor is held out replicate
    end for
  end for
  Pearson, Spearman, AUROC, AUPRC $\leftarrow$ computeMetrics($\mathcal{T}, \mathcal{P}$)
  return Pearson, Spearman, AUROC, AUPRC

References

Höfling, H. and Tibshirani, R. (2008). A study of pre-validation. *The Annals of Applied Statistics*, 2(2):643–664.

Wessels, H.-H., Méndez-Mancilla, A., Guo, X., Legut, M., Daniloski, Z., and Sanjana, N. E. (2020). Massively parallel Cas13 screens reveal principles for guide RNA design. *Nature Biotechnology*, 38(6):722–727.