CRISPys: Optimal sgRNA Design for Editing Multiple Members of a Gene Family Using the CRISPR System

Gal Hyams [1,†], Shiran Abadi [1,†], Shlomtzion Lahav [2], Adi Avni [1], Eran Halperin [3,4], Eilon Shani [1] and Itay Mayrose [1]

1 - School of Plant Sciences and Food security, Tel Aviv University, Ramat Aviv, 69978, Israel
2 - Cell Research & Immunology, Tel Aviv University, Ramat Aviv, 69978, Israel
3 - Department of Computer Science, University of California, Los Angeles, Los Angeles, CA, 90095, United States
4 - Department of Anesthesiology, University of California, Los Angeles, Los Angeles, CA, 90095, United States

***Correspondence to Itay Mayrose:** School of Plant Sciences and Food security, Tel Aviv University, Tel Aviv 69978, Israel. itaymay@post.tau.ac.il*
https://doi.org/10.1016/j.jmb.2018.03.019
Edited by Nir Yosef

Abstract

The development of the CRISPR–Cas9 system in recent years has made eukaryotic genome editing, and specifically gene knockout for reverse genetics, a simple and effective task. The system is directed to a genomic target site by a programmed single-guide RNA (sgRNA) that base-pairs with it, subsequently leading to site-specific modifications. However, many gene families in eukaryotic genomes exhibit partially overlapping functions, and thus, the knockout of one gene might be concealed by the function of the other. In such cases, the reduced specificity of the CRISPR–Cas9 system, which may lead to the modification of genomic sites that are not identical to the sgRNA, can be harnessed for the simultaneous knockout of multiple homologous genes. We introduce CRISPys, an algorithm for the optimal design of sgRNAs that would potentially target multiple members of a given gene family. CRISPys first clusters all the potential targets in the input sequences into a hierarchical tree structure that specifies the similarity among them. Then, sgRNAs are proposed in the internal nodes of the tree by embedding mismatches where needed, such that the efficiency to edit the induced targets is maximized. We suggest several approaches for designing the optimal individual sgRNA and an approach to compute the optimal set of sgRNAs for cases when the experimental platform allows for more than one. The latter may optionally account for the homologous relationships among gene-family members. We further show that CRISPys outperforms simpler alignment-based techniques by *in silico* examination over all gene families in the *Solanum lycopersicum* genome.

Introduction

Due to extensive history of local and large-scale genomic duplications, many eukaryotic genomes harbor homologous gene families of partially overlapping functions [1–4]. For example, 72% of the protein coding genes in the Plaza 3.0 Monocots database [5], which presently covers 16 fully sequenced plant genomes, belong to paralogous gene families. This redundancy often leads to mutational robustness such that the inactivation of one gene often results in no or minimal phenotypic consequence [2,4,6–8]. As there are no observable phenotypes for many single-gene loss-of-function mutants, it is often necessary to mutate multiple members of a gene family to uncover phenotypic consequences and to enable in-depth molecular characterization of their function. Here, we present a computational methodology that will facilitate such endeavors via genome editing techniques.

The Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR)–associated protein 9 nuclease (Cas9) system has been recently adopted as a genome editing technique of eukaryotic genomes. The target genomic DNA sequence consists of 20 nucleotides followed by the Protospacer Adjacent Motif (PAM), which is usually in the form of NGG (where N stands for any nucleotide and G for guanine). The system is directed to the genomic site using a

0022-2836/© 2018 Elsevier Ltd. All rights reserved.

programmed single-guide RNA (sgRNA) that base-pairs with the DNA target, subsequently leading to a site-specific double-strand break three base pairs upstream of the PAM. This break can be repaired through the Non-Homologous End Joining repair pathway, frequently resulting in a frameshift in the encoded protein, thus leading to its inactivation. Using the CRISPR–Cas9 system, DNA sequences within the endogenous genome are now regularly edited in diverse organisms as human [9], mouse [10], zebrafish [11], yeast [12], and plants [13,14], and the system is rapidly becoming the technology of choice for generating single gene knockouts for reverse genetics studies [13,15–18]. Importantly, the binding affinity of the CRISPR–Cas9 system does not require perfect matching between the sgRNA and the DNA target. Thus, in addition to cleaving the desired on-target, cleavage may occur at multiple unintended genomic sites (termed off-targets) that are similar, up to a certain degree, to the sgRNA. Several studies have demonstrated that multiple mismatches—occasionally more than four—can be tolerated between the sgRNA and the target site, depending on the location of the mismatches and their spatial distribution [19–21]. Indeed, it is well acknowledged that mismatches at PAM-distal positions are better tolerated than those occurring at PAM-proximal sites [20,22,23]. Thus, when designing an sgRNA for editing a single gene, two topics should be considered: the sgRNA sensitivity (maximizing the targeting probability of the on-target) and its specificity (minimizing the targeting probabilities of off-target sites). Several computational tools have been constructed to deal with these challenges [24–30].

To date, much effort has been devoted to refine the specificity of the CRISPR system as a means to decrease the off-target effect [31–33]. However, the low specificity of the system could be harnessed to enable a rational design of an sgRNA that would simultaneously target multiple genes. An example of such a possibility was recently shown in rice, where a single sgRNA led to the modification of three homologous genes from the cyclin-dependent kinase protein family [34]. The sgRNA used in that study was designed to perfectly match one of the family members, while the other two homologs possessed one and two mismatches and were silenced as a byproduct. A fourth homolog, with three mismatches, was not affected by this transfection. However, it is possible that considering all family members in the design process would enable the knockout of a larger fraction of the homologous gene family and would enhance the balance between their cleavage frequencies.

One approach for accomplishing this task is to align the sequences of the given genes and then locate highly similar CRISPR–Cas9 target sites in the consensus sequence, while allowing for a few mismatches between the consensus and each of the aligned sequences. This strategy is implemented in the MultiTargeter tool [35], which also accounts for CRISPR-specific considerations such as allowing for mismatches only within PAM-distal nucleotides. In case that multiple potential sgRNAs are found, these are ranked according to their efficacy, as predicted by the CFD score [25]. While the MultiTargeter algorithm is computationally efficient, it may miss valid candidates, for example, similar subsequences that appear in homologous genes but do not overlap in the resulting alignment, or those that are located in opposite strands. Furthermore, targets that have similar copies along a gene can increase the targeting efficiency; however, such considerations are not accounted for in this approach.

Here, we present a novel method, termed CRISPys, aimed for the design of an optimal set of sgRNAs for silencing multiple members of a gene family using the CRISPR–Cas9 system. CRISPys detects highly similar sequences among the set of all potential CRISPR–Cas9 targets (i.e., sequences that are followed by a PAM site), located within the genes of interest, and designs sgRNAs that would target the gene set with highest efficacy. CRISPys can further incorporate any scoring function specifying the targeting efficiency of a given sgRNA to a given genomic site, thus allowing for flexible use of the method with the accumulation of knowledge regarding this emerging genome engineering technique. We present the utility of CRISPys by applying it in a genome-wide manner to numerous gene families in the *Solanum lycopersicum* genome and compare its performance to the existing alignment-based approach.

Results

Algorithm description

Given a set of genes, $G = \{g_i\}$, potentially belonging to a homologous gene family, and a scoring function φ, we would like to identify suitable sgRNA candidates that are likely to target the largest number of genes in G. The input scoring function φ(sgRNA, target) → $[0,1]$ specifies the estimated targeting efficiency of a given sgRNA to a given genomic site, such as the CFD score [25], CROP-IT [36], Optimized CRISPR Design [27], or CRISTA [24]. We present four alternative strategies for sgRNA design, depending on the needs and limitations of the researcher:

I. A single sgRNA that could best target the entire gene set G

II. A single sgRNA that is optimized to target each of the input genes. However, this design could ignore some of the genes if it does not succeed to target all with high efficiency.

III. Multiple sgRNAs, each directed toward subgroup of homologous genes

IV. The minimal set of sgRNAs that could target the entire gene set with high efficiency

Ideally, for each of these alternatives, all possible sgRNA candidates would be examined to identify the one (or set of sgRNAs) that would target the given gene set with highest propensity. This, however, entails the examination of an exceedingly large number of sgRNA possibilities (4^l, where l is the length of the sgRNA, typically $l = 20$) leading to computationally intractable running time. Thus, the examined set is narrowed to the sgRNAs that are most relevant to the input gene set. To this end, we first cluster all potential targets within the genes in G into a hierarchical tree structure that specifies the similarity among the targets. Then, the design is optimized in the course of a bottom-up traversal of this tree.

Targets-tree construction

Given the set of genes G, all potential targets are first extracted for each gene $g_i \in G$. By default, these are defined as 20-nt-long sequences upstream to an NGG motif; additional PAM motifs (e.g., NAG) or other sequence lengths (17–23) can be specified. Second, the entire set of potential targets, T, are clustered using the Unweighted Pair Group Method with Arithmetic Mean (UPGMA) hierarchical clustering algorithm [37], which yields a tree structure, such that more similar targets (represented by the tips of the tree) are placed closer to each other on the tree. The input pairwise distance matrix used for clustering is computed using the scoring function $\varphi(\text{sgRNA}, \text{target})$ that is transformed into a metric that represents the distance between two targets (see Methods). An internal node, a, in this tree induces a set, T^a, of potential targets that are the descendants of this node. As a gets closer to the root, the size of T^a increases such that the contained targets become less similar and are less likely to be targeted by a single sgRNA. We also denote by G^a the subgroup of the input genes to which the targets in T^a belong. Figure 1 presents an example of a targets-tree for a family of five genes, each represented by a single potential target.

Targets-tree traversal and identification of sgRNA candidates

Once the targets-tree is constructed, the algorithm proceeds by traversing the tree in a post-order manner,

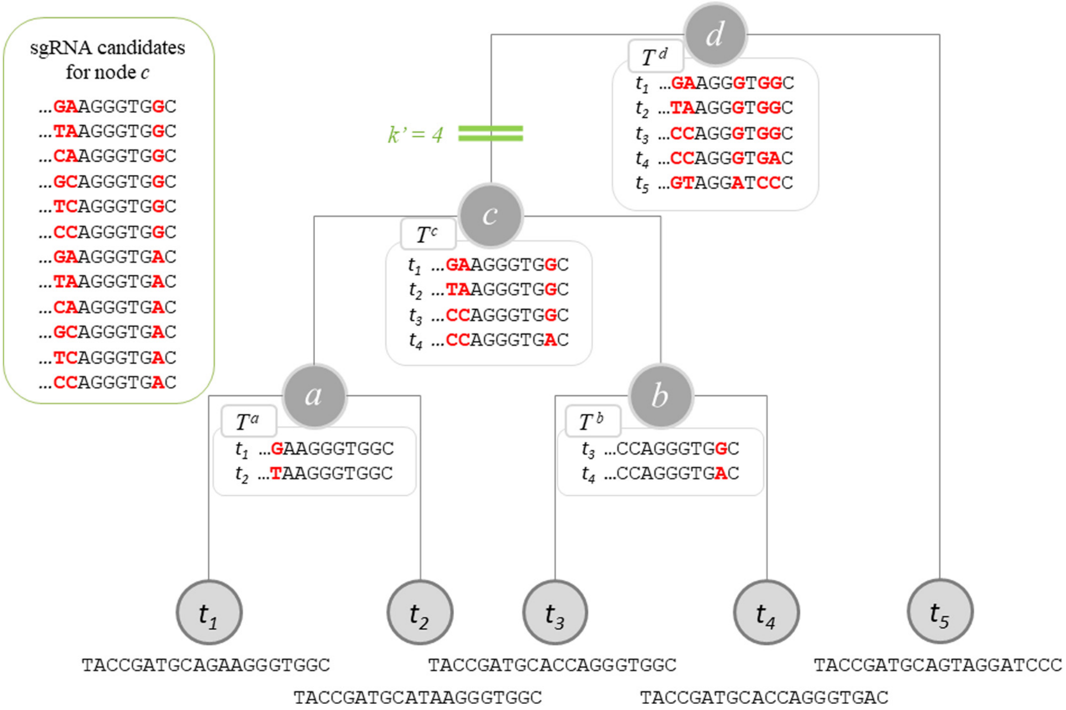

Fig. 1. An illustrative example of a targets-tree and post-order traversal. Five genomic targets (t_1–t_5) are hierarchically clustered according to pairwise distances. Each 20-nt target represents a different gene. The target set induced by each internal node is indicated in the box below it and the nucleotides at polymorphic positions are indicated in red. The numbers of polymorphic sites within T^a, T^b, T^c, and T^d are 1, 1, 3, and 5, respectively. For illustration, the cutoff of polymorphic sites, k', is set to 4. Since the number of polymorphic sites in T^d is above k', the search for candidates stops at node c. All possible combinations of the polymorphic sites in node c induce the sgRNA candidates that are listed at the panel to the left of the tree.

identifying sets of targets for which sgRNA candidates will be designed. Specifically, upon reaching an internal tree node a, the polymorphic sites in the induced target set, T^a, are counted. A site is termed polymorphic if more than one type of nucleotide is found at this position in different targets (Fig. 1). If the number of polymorphic sites is above a cutoff k, the search does not proceed up the tree and potential sgRNAs are designed based on each of the descendent subtrees. Otherwise, tree traversal is continued. We note that the number of mismatches between a potential sgRNA and each of the targets can be small even if the number of polymorphic sites is rather large (see Fig. S1 for an illustrative example). In preliminary trials, setting k to 13 or more led to the assessment of additional sgRNAs with much lower efficacy. Exploring such a broad set of sgRNAs against all targets resulted in exceedingly long running time without producing any improvement in the assessed efficacies of the designed sgRNAs. Therefore, k was set to 12 in all our analysis as well as in CRISPys webserver.

For each target set identified, sgRNA candidates are designed by enumerating over all possible combinations of the polymorphic sites found within it (Fig. 1). The efficacy of each candidate sgRNA s to target the genes in G is then assessed. Specifically, $\varphi_s(g_i)$, the targeting propensity of gene $g_i \in G$ by sgRNA s is computed by considering all possible targets that belong to g_i as follows:

$$\varphi_s(g_i) = 1 - \prod\nolimits_j \left[1 - \varphi(s, t_{ij})\right] \qquad (1)$$

where $\varphi(s, t_{ij})$ is the targeting propensity of the jth target site of gene g_i by sgRNA s (as calculated by the input scoring function). This way, as the number of similar targets in g_i increases, so does the propensity that the gene is targeted by the specified sgRNA; in case a gene has only one target, $\varphi_s(g_i)$ reduces to $\varphi(s, t_{i1})$. Notably, the calculation of $\varphi_s(g_i)$ [as well as Eqs. (2, 5)], relies on the assumption that for every s and t_{ij}, $\varphi(s, t_{ij})$ are considered as independent probabilities. While some of the scoring functions do not estimate the actual targeting probability, in all of them, a score closer to 1.0 indicates a higher targeting propensity, and that this propensity decreases toward 0.0. These scores are thus assumed to represent the probability of success in a Bernoulli experiment and treated as probability functions.

Next, we use $\varphi_s(g_i)$ for each $g_i \in G$ to design the optimal sgRNA by one of four strategies detailed below.

Design strategy I: A single sgRNA that could best target the entire gene set. When reaching a node a at which the subtree is pruned the targeting expectation across **all** genes in G^a is computed for each potential sgRNA s:

$$E_s(G^a) = \sum\nolimits_{g_i \in G^a} \varphi_s(g_i) \qquad (2)$$

The optimal sgRNA candidate under this strategy is defined as

$$s^{\exp} = \underset{s}{\operatorname{argmax}} \left\{ E_s(G^a) \middle| \forall a \right\} \qquad (3)$$

This strategy opts to target the entire group G, but effectively, genes that contain too many mismatches and are not found within G^a could be skipped as their targeting score is negligible, and thus $E_s(G^a) \cong E_s(G)$.

Design strategy II: A single sgRNA optimized to target most of the genes in G, each gene with a score above a given threshold Ω. Occasionally, especially when G is large or the genes are distinct, an sgRNA that could target the entire gene set with high efficacy does not exist. In such cases, instead of optimizing the targeting potential of all genes, it is preferable to concentrate only on those with high targeting propensity. We thus implemented a second design strategy that ignores genes whose targeting propensity by a considered sgRNA is below a given threshold Ω. Specifically, when reaching node a, G_s^Ω indicates the group of genes that are expected to be targeted by sgRNA s above a given threshold Ω:

$$G_s^\Omega = \left\{ g_i \in G^a \middle| \varphi_s(g_i) \geq \Omega \right\} \qquad (4)$$

Let c_s^Ω be the size of this group: $c_s^\Omega = |G_s^\Omega|$ and let $c_{\max}^\Omega$ be the highest c_s^Ω value computed for G^a. Because there may be multiple sgRNAs with this $c_{\max}^\Omega$ value, the optimal sgRNA is chosen as the one with the highest propensity to target all $c_{\max}^\Omega$ genes:

$$s^\Omega = \underset{s}{\operatorname{argmax}} \prod\nolimits_{g_i \in G_s^\Omega} \varphi_s(g_i) \qquad (5)$$

$$\text{s.t.}\, c_s^\Omega = c_{\max}^\Omega$$

Notably, the use of this criterion necessitates the use of a pre-specified threshold Ω. Setting Ω to 0.0 results in all input genes affecting the score, while setting it to 1.0 practically considers only the most certain matches (e.g., with no mismatches). See Methods for a description of suitable choice of Ω.

Design strategy III: Multiple sgRNAs, each directed toward a subgroup of homologous genes. Although mutating a gene family can overcome functional redundancy, mutating too many genes simultaneously may lead to a lethal or sterile phenotype, limiting our ability to elucidate their function. In such cases, dividing G into smaller groups according to sequence similarity would increase the flexibility of the screen and allow a more focused experimental design. To this end, this strategy recursively splits G into homologous subgroups and generates potential sgRNAs for each of them. Specifically, a hierarchical clustering of G is constructed using the UPGMA algorithm (we note that this tree represents the similarities among the input genes, while the hierarchical clustering detailed in the Targets-tree construction section represents the similarities among the targets). The gene tree is constructed using the UPGMA algorithm as implemented in Biopython [38]. The input distance matrix for the UPGMA is computed using Protdist [39] given a multiple sequence alignment generated by MAFFT [40], with its default options, on the translated genes. CRISPys design strategies I or II are then applied to each node of the constructed UPGMA tree, producing an optimal sgRNA design for each homologous subgroup.

Design strategy IV: The minimal set of sgRNAs that could target the entire gene set with high efficiency. Strategy I focused on the design of an optimal sgRNA that could best target the entire gene set G. In case that a single sgRNA could not be found but mutating the whole family is desired in the cost of applying multiple sgRNAs, a possible approach would be to use strategy III to design sgRNAs for subgroups of G. Nevertheless, this could lead to suboptimal design since gene partitioning according to homology (or functionality) does not guarantee the availability of similar targets within each partition that would provide the minimal and most efficient set of sgRNAs. Thus, an alternative strategy would aim to design the minimal set of sgRNAs that target the entire gene set with highest efficiency. This can be formulated as the set cover [41] problem. Accordingly, given a collection of elements, each belonging to one or more sets, the task is to identify the smallest number of sets whose union covers the entire collection. In our case, each element represents a gene. Several genes belong to the same set if and only if they can be targeted by the same sgRNA. Therefore, solving the set cover problem would provide the optimal and minimal set of sgRNAs that can target all of the genes. Specifically, all potential sgRNAs are collected while traversing the targets-tree as detailed above. For every sgRNA s, the subgroup of all genes for which $\varphi_s(g_i) \geq \Omega$ (Ω defined as in design strategy II), is considered a set. Since the set cover problem cannot be solved in polynomial time (i.e., it is NP-hard), for computation efficiency, we apply an approximation algorithm ("greedy set cover" [42]). Namely, the sets that cover the largest number of elements (genes) that have not been covered to that point are iteratively selected until all elements are covered. In case of a tie, the selected sgRNA is the one that maximizes the targeting efficiency of the added genes (Eq. (5)).

In silico application over the tomato genome

To demonstrate the utility of CRISPys and to compare its single sgRNA design strategies I and II, we applied it to all 3697 gene families of size 2–10 within the tomato (*S. lycopersicum*) genome using CFD as the scoring function. The classification of

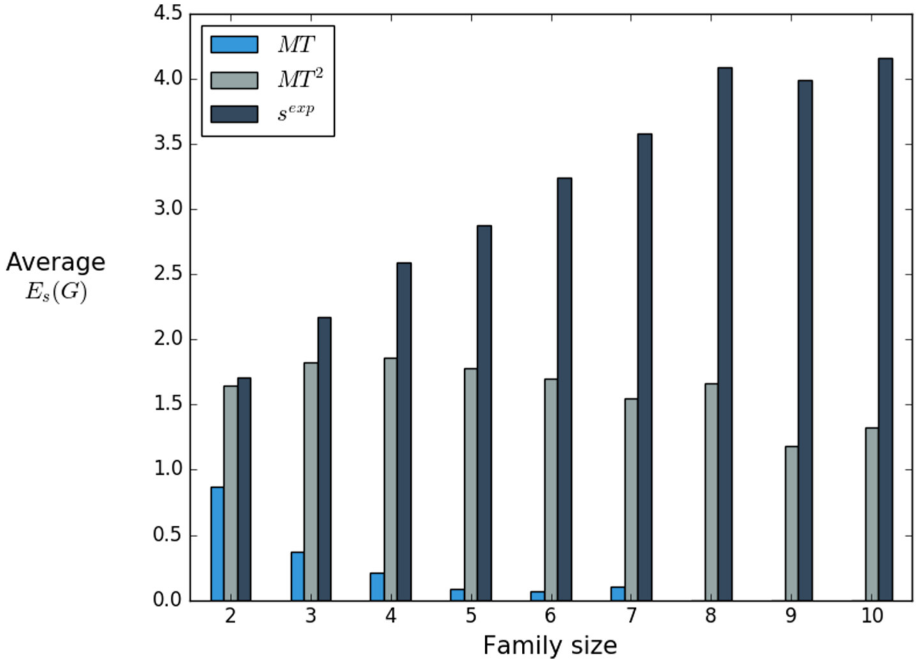

Fig. 2. Comparison of targeting efficiencies across the tomato genome. The average over the targeting expectation for the entire gene family G [computed using $E_s(G)$ and the CFD score; Eq. (2)] by the optimal sgRNA s according to design strategy I of CRISPys (s^{exp}), MultiTargeter (MT), and MT^2. The families are binned by family size across the tomato genome.

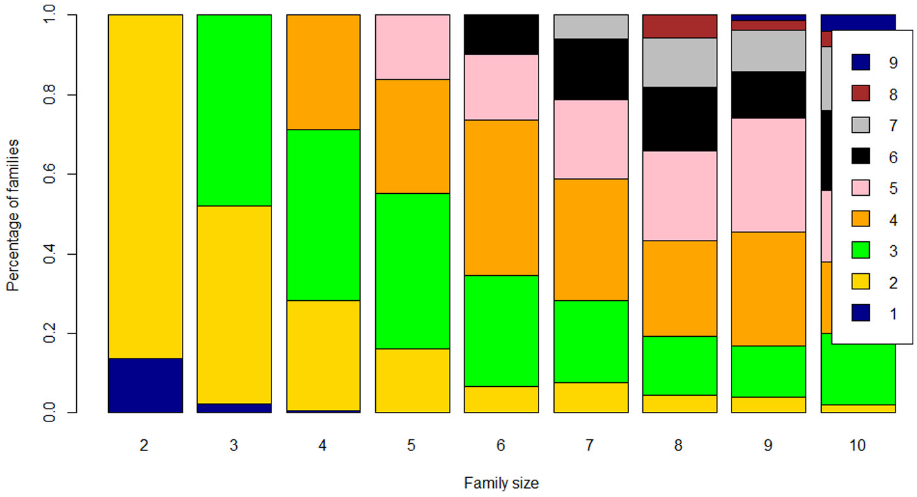

Fig. 3. CRISPys results across the tomato genome using design strategy II. The bars are divided according to family size between 2 and 10. For each family size, the families are categorized to the number of genes that achieved a score above $\Omega = 0.45$ by the designed s^{Ω} (specified by the color bar at the right of the panel). For example, s^{Ω} was designed for targeting only one gene in 13% of the families of size 2, and the complete set of 2 genes in 87% of these families.

genes to families was taken from the Plaza plant comparative genomics database [5]. Using design strategy I—that aims to target the entire gene set—the average targeting expectation increases with the number of genes that are included in the family (Fig. 2). This incline approaches a plateau for families of size eight or higher. A similar trend is obtained using design strategy II (Fig. 3), which optimizes for genes whose targeting propensity is high. As expected, as the family size increases, the optimal sgRNA could target more genes, although the tendency to target *all* genes in the family decreases. For example, for 87% of the families of size two, the designed sgRNA could target all family members. This percentage decreases to 10% for families of size 6, while for all 50 gene families of size 10, no such sgRNA could be obtained. Finally, an asymmetric tradeoff between the two design strategies is demonstrated in Table 1. When all genes in the family are considered, the targeting expectation of design strategy I is higher than that of design strategy II (second and fourth columns). However, when only genes with high targeting propensity are counted (above a CFD score of 0.45), the design strategy I would result in targeting fewer genes than design strategy II (third and fifth columns).

Comparison between CRISPys and a consensus-based approach

To compare the results obtained using CRISPys to an alignment-based approach, we applied the MultiTargeter tool [35] to the same set of families within the *S. lycopersicum* genome. Notably, MultiTargeter designs an sgRNA to target all family members, while no results are obtained if one or more genes cannot be matched. In contrast, design strategy I of CRISPys always provides a result when all family members are considered, while design strategy II provides a result that may be directed to a portion of the genes. Thus, in this comparison, we defined the prediction of CRISPys as successful if the targeting score for each of the genes is above Ω. This definition was applied to both design strategies I and II of CRISPys. As demonstrated in Fig. 4, the percentage of successful designs by all alternatives decreases as the family size increases. However, this decline is much shallower for the two design criteria of CRISPys, particularly design strategy II, as

Table 1. Comparison between the decision criteria of CRISPys

Family size	Design strategy I: s^{exp}		Design strategy II: s^{Ω}	
	Average $E_{s^{\text{exp}}}(G)$	Average $c_{s^{\text{exp}}}{}^{\Omega}$	Average $E_{s^{\Omega}}(G)$	Average $c_{s^{\Omega}}{}^{\Omega}$
2	1.70 (0.3)	1.77 (0.4)	1.69 (0.3)	1.87 (0.3)
3	2.16 (0.5)	2.28 (0.6)	2.12 (0.5)	2.45 (0.5)
4	2.59 (0.7)	2.81 (0.8)	2.15 (0.7)	3.07 (0.7)
5	2.87 (0.7)	3.04 (0.93)	2.72 (0.7)	3.44 (0.9)
6	3.23 (0.9)	3.51 (1.05)	3.06 (0.9)	4.04 (1.0)
7–8	3.76 (1.2)	4.11 (1.5)	3.51 (1.3)	4.61 (1.4)
9–10	4.03 (1.2)	4.38 (1.5)	3.76 (1.2)	5.18 (1.5)

The table presents the average number of targeted genes across all families of size 2–10 in the tomato genome. $E_s(G)$ is the targeting expectation for the entire gene family G (Eq. (2)) by the optimal sgRNA s, according to the two design strategies. $c_s{}^{\Omega}$ is the number of genes with targeting efficiency above a threshold of $\Omega = 0.45$ (Eq. (4)) by each design strategy. The standard deviation of each statistic is given in parentheses.

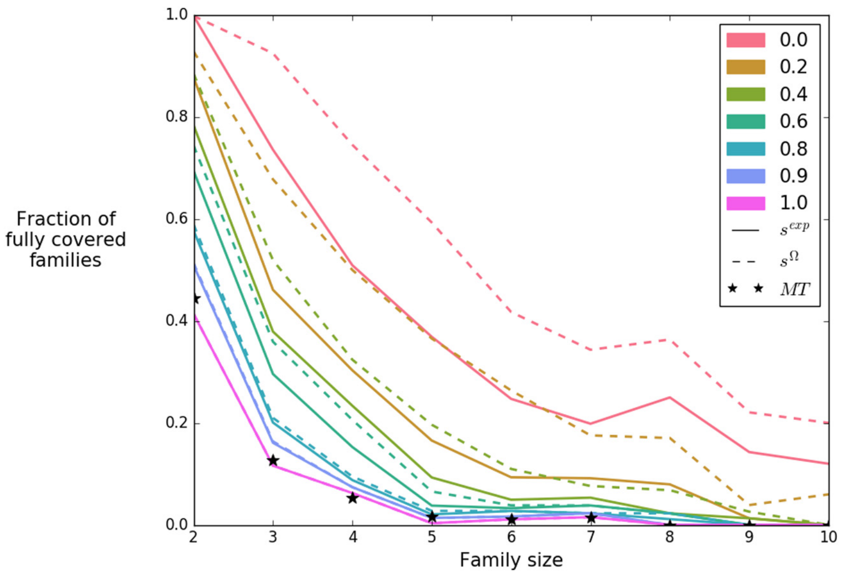

Fig. 4. Whole family coverage by design strategies I and II, and MultiTargeter across changing Ω thresholds. The families for which the entire family members obtained a score above the indicated Ω thresholds (according to the legend) are presented as a fraction of the family size, for each family size. The solid and dashed lines represent design strategies I (s^{exp}) and II (s^{Ω}), respectively. The fraction of families for which MultiTargeter (MT) produced results are marked with asterisks.

compared with that obtained using MultiTargeter. Second, CRISPys provided a successful design for a larger fraction of families compared to MultiTargeter, for all but the most stringent threshold of $\Omega = 1$. This latter threshold is generally stricter than the criterion employed by MultiTargeter, which allows for any mismatch in the eight positions most distal to the PAM (for some of these mismatches, the CFD score is 1.0, while for others, it may be as low as 0.35).

There are two main differences between CRISPys and MultiTargeter. First, CRISPys directly clusters the targets and thus permits highly similar targets to be dispersed along the gene sequences rather than requiring that they would be aligned together. Second, MultiTargeter allows for a single mismatch between the consensus and each of the genes, and forces this mismatch to be at a single consensus position. CRISPys, on the other hand, employs a scoring function, which grants it with increased flexibility. To distinguish between these two factors, we modified MultiTargeter such that it meets the scoring method of CRISPys and term this version MT^2. Specifically, MT^2 allows for up to 12 polymorphic positions in the consensus. For each 20-nt match, an sgRNA candidate is designed by enumerating over all possible combinations of nucleotides in the polymorphic positions and the one with the highest summation of CFD scores over all genes is selected for this match. The final sgRNA is the one with highest score over all candidates. This computation manner is similar to CRISPys design strategy I (Eq. (2)), and thus can be used to evaluate the benefit of the hierarchical clustering procedure. Our analysis demonstrate that the results obtained using MT^2 substantially improved over those of MultiTargeter across all family sizes, indicating that setting strict rules on the number of mismatches (and their location) reduced flexibility and, in many cases, resulted in inferior design (Fig. 2). Furthermore, the results obtained by CRISPys for the smallest gene families (size 2) did not substantially improve over those of MT^2, although the improvement was statistically highly significant ($p < 10^{-15}$; paired *t*-test). For such small families, a simple alignment strategy is, in many cases, sufficient to locate highly similar targets. However, the difference between the methods became more noticeable with the increase in the size of the gene family, when genes are harder to align and alignment errors are more common. Together, these results demonstrate that the integration of both factors (the clustering of targets and a flexible scoring function) in the design yielded the most promising sgRNA candidate.

Discussion

In this work, we presented CRISPys, a novel computational method that utilizes the nonspecificity of the CRISPR–Cas9 system for the design of an optimal sgRNA that would most efficiently mutate multiple members of a gene family. Acquiring a set of sgRNA candidates is done through aggregating the 20-nt-long targets according to their similarity and embedding mismatches within them. In this manner, the best sgRNA might not be identical to any of the targets such that it can optimally target the set of input genes by embedding the least destructive mismatches to each. Notably, previous studies deduced some observatory rules regarding the effect of the mismatches on the sgRNA efficacy. Although these are generally based on experimental evidence, they do not always conform with one another. For example, while Hsu *et al*. [20] concluded that two mismatches at the PAM-proximal region abolish sgRNA activity, Tsai *et al*. [19] observed that such cases may still be cleaved at relatively high frequencies. In order to refrain from setting rudimentary rules, the design of CRISPys is dictated by one of several scoring functions regarding the targeting propensity of genomic sites by a given sgRNA. This allows the user to prioritize the considerations of choice. For example, the function provided by the "Optimized CRISPR Design" [27] computes the effect of position-specific mismatches on the targeting efficiencies based on experimental data. In practice, it is quite dichotomous, as it assigns a score of 1.0 to all on-targets and to some targets with a single mismatch, while the score of nearly all other targets approaches zero. Using this function will thus consider for each candidate sgRNA only the most definite targets. The CFD function [25], on the other hand, was trained on knockdown efficiency experimental data and is more delicate, since it is sensitive to the type of mismatch at each position. Because its possible values are spread more evenly across the [0,1] range, its use will consider a broader collection of potential targets. Aside from functions that consider the pairwise similarity between the sgRNA and the DNA target, CRISPys can also integrate functions, such as CRISTA [24], that account for additional genomic features (e.g., the GC content or the DNA rigidity surrounding the target site). Notably, using alternative scoring functions yield different results from, and as the research of the CRISPR system evolves, any new function can be easily incorporated within CRISPys. This flexibility in the underlying scoring function further enables CRISPys to readily design sequences for guiding any of the emerging CRISPR nucleases variants, such as Cas9 endonucleases with alternative PAM sequences [32,43], deactivated Cas9 mutants purposed for RNA interference (CRISPRi) [44], or the recently studied class-2 Cas proteins [45–47].

The sgRNA design by CRISPys depends not only on the scoring function but also on the criterion by which one chooses to select the optimal sgRNA. Two alternative strategies were presented for a single sgRNA design. A user that is interested to maximize the number of family members to be targeted should, in principle, select the sgRNA with the highest targeting expectation over the entire set of input genes, as computed using the $s^{\exp}$ strategy (Eq. (2)). However, at least at present, the large gaps of knowledge surrounding the CRISPR–Cas9 mode of action and the noisy experimental procedures that are used for its evaluation translate to scoring functions with large degrees of uncertainty. Given the costly (and timely) experimental resources that are needed to validate a successful assay, researchers are most often interested in focusing their efforts in validating those targets whose targeting probability is high. This is implemented using the s^{Ω} design, but necessitates the use of a pre-specified threshold above which targets are considered.

An alternative approach for targeting multiple genomic sites using a single sgRNA has been previously implemented in the MultiTargeter webserver [35]. This approach detects potential targets in the consensus sequence of a multiple sequence alignment of the input sequences. This procedure entails several difficulties. First, it relies on the ability to correctly align the input gene sequences, a procedure that is known to be error-prone [48–51]. Second, this approach entails that the entire set of genes is aligned at a particular site, and thus, a single incompatible sequence suffices to prevent the design of a proper sgRNA, even if the rest are highly conserved. Third, the consensus would assign the most abundant character at each position, while a more balanced design would disperse mismatches over the input genes, accounting for the specific penalty of each assignment. In contrast, since CRISPys first clusters all potential targets according to sequence similarity, it is not dependent on their orientation or location within the alignment. Figures S2–S5 present representative examples of such cases in the tomato genome that could not be detected by an alignment-based procedure but could be successfully recovered using CRISPys clustering approach. Moreover, by incorporating any specified scoring function, CRISPys allows for a more sensitive consideration of each site. Indeed, as shown by our analysis of the tomato genome, CRISPys succeeded in providing promising sgRNA candidates for a larger number of gene families compared to the MultiTargeter consensus-based approach.

Evidently, CRISPys aims at optimizing the editing efficacy of the input gene family. However, a designed sgRNA may also target additional "off-target" sites, leading to the knockout of undesired genes. Therefore, sgRNA design should also minimize any off-target effects. Ideally, off-target considerations should be integrated within the computations performed by CRISPys. One option would be to balance between efficiency (maximizing targeting propensity of the input genes) and specificity (minimal off-target effect) using a tunable parameter. This, however, requires a genome-wide search for potential off-targets for each and every sgRNA considered throughout the course of the algorithm, rendering it computationally infeasible. In addition, a genome-wide search for off-targets obligates the availability of the reference genome, which would have set a prerequisite for using CRISPys. Alternatively, CRISPys generates a list of sgRNAs ranked according to their computed efficacies. For each of these, off-target detection could be performed through a number of existing applications [26,27,30,52], thereby allowing researchers to choose the sgRNA that is most suitable for their need.

Recently, multiplex genome editing has been introduced, thereby enabling the application of multiple sgRNAs within a single construct [9,22,23,53–56]. These systems have been shown to be useful for the simultaneous knockout of multiple protein coding genes and for the deletion of noncoding RNA regions and other genetic elements [34,57]. Multiplex genome editing could be combined with the CRISPys algorithm to design a set of sgRNAs that would collectively mutate a large fraction of the input gene set. The s^{Ω} design option of CRISPys is particularly appealing in this regard since the Ω threshold could be tuned in such a way to allow a more strict (or lenient) design of sgRNAs such that each sgRNA would target a narrower (or broader) fraction of the genes. A refined tuning of the Ω threshold can be derived with specific considerations of the gene family at hand and the experimental conditions (e.g., the number of homologous genes, the sequence homology among the family members, and the number of sgRNAs that are collectively applied). Notwithstanding, the application of multiple sgRNAs simultaneously within a multiplex system also comes at the cost of potential efficacy reduction [58] as well as a higher number of off-targets. Moreover, increasing the frequency of potential target sites enhances the chances to chromosomal translocations and may not be desired.

Ultimately, the sgRNAs designed by CRISPys should be validated experimentally. This task could present some difficulties since the programmed sgRNA could practically contain mismatches to all of the assessed targets, and thus, the results would be highly dependent on the sensitivity of the experimental system. One possibility is to validate the cleavage sites of the designed sgRNAs, for example, by using *in vitro* digestion assays [58]. Another possibility is to test the knockout activity *in vivo* by targeting genes with known phenotypes, or other marker genes, in systems such as yeast, mammalian cells, or bacteria. Evaluation of the sgRNAs designed by CRISPys in such systems should enable its utilization for designing sgRNAs libraries to screen for phenotypes whose expression is dictated by genes that backup each other in human, plants, and other organisms.

Methods

Determination of the Ω thresholds

To set Ω to realistic values, the threshold used for performance evaluation was determined according to an experimental data set that was profiled by the genome-wide detection technique, GUIDE-Seq [19]. Specifically, the data in that study are composed of a collection of sgRNAs that overall cleaved 413 targets throughout the human genome. These validated targets were sorted according to their reported cleavage efficiencies. The $\Omega = 0.45$ threshold used for analysis is the averaged CFD score [25] for the targets in the 90–95 percentiles, representing highly cleaved off-target sites.

Converting the scoring function to a metric

In order to construct a hierarchical clustering of the target set T, the distance between every two targets must be computed. In the context of the CRISPR–Cas9 system, however, available scoring functions assess the targeting efficiency of a nuclear target by a given sgRNA rather than the distance between two nuclear targets. A naïve approach would be to set one of the targets as the sgRNA and compute its targeting efficiency to the other target. But such an approach will not result in a valid distance metric (e.g., the scoring function is not necessarily symmetric nor does it satisfy the triangle inequality). We thus implemented two alternative procedures for converting a scoring function φ to an Euclidean distance function. The first alternative corresponds to a scoring function, like the CFD score [25], that treats each position independently, such that each mismatch is penalized according to the type and position of the mismatch, and the resulting score is a multiplication over all individual positions. Specifically, given a 20-nt-long target t_i, a vector of length 80 is constructed in which every four entries correspond to a position in t_i; the first entry specifies the penalty according to φ if “A” was placed in this position in the opposing sgRNA, the second stands for “C”, and so on. Given this vector representation, the distance between two targets is calculated as the Euclidean distance between the two corresponding vectors. Thus, targets that are similar to one another receive similar scores for the different substitution possibilities in every position (dictated by φ), which in turn leads to a low Euclidean distance.

The second conversion was implemented to deal with any scoring function, without relying on its specific characteristics. The scoring function provided by CRISTA [24] is one example where the cleavage propensities are computed based on a non-linear combination of features. This conversation is based on transforming every target to a multidimensional space, where it is represented by a vector of propensities by a large set of sgRNAs, S. Specifically, for every target $t_i \in T$, its propensity $\varphi(s, t_i)$ by each sgRNA $s \in S$ is computed. Here, we defined S to be the set of sgRNAs that perfectly match the targets in T but other possibilities, such as a randomly generated set of sgRNAs can be used. This produces a representative vector of length $|T|$ for every target in the new space. The distance between targets t_i and t_j is calculated as the Euclidean distance between their corresponding vectors. Thus, a couple of targets with similar efficiencies by a set of sgRNAs will be converted to vectors with similar values, leading to a low Euclidian distance compared to targets with diverse targeting scores according to the same set of sgRNAs. We note that while this conversion procedure is more computationally demanding than the former technique detailed above, the two approaches yield similar results (Supplementary Text S1).

Program availability

An online version of the CRISPys algorithm described here is freely available at http://multicrispr.tau.ac.il/. The server accepts as input a set of (potentially homologous) sequences for which sgRNA candidates should be designed. In order to avoid targeting the designed sgRNA at intron–exon junctions, users may provide each gene as a set of exon sequences. The webserver enables the design according to the four strategies presented in this paper. By default, design strategy I is computed, but strategy II may be employed by setting an Ω threshold. Design strategy III is computed in case that the option to consider the homologous relationships among the input genes is selected. Strategy IV is computed automatically if a threshold is set (i.e., design strategy II is chosen). The webserver allows users to choose among several available functions that determine the targeting efficiency of a given sgRNA on a DNA target by (with the CFD score [25] being the default) and alternative PAM types. Once the sgRNAs design process is completed, the website provides the possibility to search off-targets through CRISPOR [52] or CRISTA [24] webservers.

Acknowledgment

This research was supported by a grant from the Chief Scientist of the Israeli Ministry of Agriculture and Rural Development (1832/14; I.M., A.A., and E.S.), a European Research Council Starting Grant (757683-RobustHormoneTrans to E.S.), MSc fellowships provided by the Manna Program in Food Safety and Security and the Edmond J. Safra Center for Bioinformatics at Tel-Aviv University (to G.H.), and PhD fellowships provided by the Rothschild Caesarea Foundation and the Edmond J. Safra Center for Bioinformatics at Tel-Aviv University (to S.A.).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jmb.2018.03.019.

Received 30 October 2017;
Received in revised form 19 February 2018;
Accepted 22 March 2018
Available online 3 April 2018

Keywords:
gene knockout;
gene family knockout;
CRISPR–Cas9;
functional redundancy

†G.H. and S.A. have contributed equally to this work.

Abbreviations used:
CRISPR, Clustered Regularly Interspaced Short Palindromic Repeats; Cas9, CRISPR-associated protein 9; sgRNA, single-guide RNA; PAM, Protospacer Adjacent Motif; *S. lycopersicum*, *Solanum lycopersicum*; UPGMA, Unweighted Pair Group Method with Arithmetic Mean.

References

[1] A. Wagner, Redundant gene functions and natural selection, J. Evol. Biol. 12 (1999) 1–16.

[2] T.J. Gibson, J. Spring, Genetic redundancy in vertebrates: polyploidy and persistence of genes encoding multidomain, Proteins (1995) 46–49.

[3] M.A. Nowak, M.C. Boerlijst, J. Cooke, J.M. Smith, Evolution of genetic redundancy, Nature 388 (1997) 167–171.

[4] R. Kafri, M. Springer, Y. Pilpel, Genetic redundancy: new tricks for old genes, Cell 136 (2009) 389–392.

[5] S. Proost, M. Van Bel, D. Vaneechoutte, Y. Van de Peer, D. Inzé, B. Mueller-Roeber, K. Vandepoele, PLAZA 3.0: an access point for plant comparative genomics, Nucleic Acids Res. 43 (2015) D974–D981.

[6] J.D. Laney, M.D. Biggin, Redundant control of Ultrabithorax by zeste involves functional levels of zeste protein binding at the Ultrabithorax promoter, Development 122 (1996) 2303–2311 (http://www.ncbi.nlm.nih.gov/pubmed/8681810 (accessed November 7, 2016)).

[7] A. Joyner, K. Herrup, B. Auerbach, C. Davis, J. Rossant, Subtle cerebellar phenotype in mice homozygous for a targeted deletion of the En-2 homeobox, Science (80-.) (1991) 251.

[8] Y. Saga, T. Yagi, Y. Ikawa, T. Sakakura, S. Aizawa, Mice develop normally without tenascin, Genes Dev. 6 (1992) 1821–1831 (http://www.ncbi.nlm.nih.gov/pubmed/1383086 (accessed November 7, 2016)).

[9] M. Jinek, A. East, A. Cheng, S. Lin, E. Ma, J. Doudna, RNA-programmed genome editing in human cells, elife 2 (2013), e00471. https://doi.org/10.7554/eLife.00471.

[10] D. Li, Z. Qiu, Y. Shao, Y. Chen, Y. Guan, M. Liu, Y. Li, N. Gao, L. Wang, X. Lu, Heritable gene targeting in the mouse and rat using a CRISPR–Cas system, Nat. Biotechnol. 31 (2013) 681–683.

[11] W.Y. Hwang, Y. Fu, D. Reyon, M.L. Maeder, S.Q. Tsai, J.D. Sander, R.T. Peterson, J.R.J. Yeh, J.K. Joung, Efficient genome editing in zebrafish using a CRISPR–Cas system, Nat. Biotechnol. 31 (2013) 227–229.

[12] J.E. DiCarlo, J.E. Norville, P. Mali, X. Rios, J. Aach, G.M. Church, Genome engineering in *Saccharomyces cerevisiae* using CRISPR–Cas systems, Nucleic Acids Res. 41 (7) (2013) 4336–4343

[13] C. Brooks, V. Nekrasov, Z.B. Lippman, J. Van Eck, Efficient gene editing in tomato in the first generation using the clustered

regularly interspaced short palindromic repeats/CRISPR-associated9 system, Plant Physiol. 166 (2014) 1292–1297.

[14] K. Belhaj, A. Chaparro-Garcia, S. Kamoun, V. Nekrasov, Plant genome editing made easy: targeted mutagenesis in model and crop plants using the CRISPR/Cas system, Plant Methods 9 (1) (2013).

[15] J. Travis, Making the cut, Science (80-.) 350 (2015) 1456–1457.

[16] W. Jiang, B. Yang, D.P. Weeks, Efficient CRISPR/Cas9-mediated gene editing in *Arabidopsis thaliana* and inheritance of modified genes in the T2 and T3 generations, PLoS One 9 (2014), e99225.

[17] J.A. Doudna, E. Charpentier, The new frontier of genome engineering with CRISPR–Cas9, Science (80-.) 346 (2014) 1258096.

[18] Z. Feng, Y. Mao, N. Xu, B. Zhang, P. Wei, D.-L. Yang, Z. Wang, Z. Zhang, R. Zheng, L. Yang, Multigeneration analysis reveals the inheritance, specificity, and patterns of CRISPR/Cas-induced gene modifications in Arabidopsis, Proc. Natl. Acad. Sci. 111 (2014) 4632–4637.

[19] S.Q. Tsai, Z. Zheng, N.T. Nguyen, M. Liebers, V.V. Topkar, V. Thapar, N. Wyvekens, C. Khayter, A.J. Iafrate, L.P. Le, M.J. Aryee, J.K. Joung, GUIDE-seq enables genome-wide profiling of off-target cleavage by CRISPR–Cas nucleases, Nat. Biotechnol. 33 (2014) 187–197, https://doi.org/10.1038/nbt.3117.

[20] P.D. Hsu, D. a Scott, J. a Weinstein, F.A. Ran, S. Konermann, V. Agarwala, Y. Li, E.J. Fine, X. Wu, O. Shalem, T.J. Cradick, L. a Marraffini, G. Bao, F. Zhang, DNA targeting specificity of RNA-guided Cas9 nucleases, Nat. Biotechnol. 31 (2013) 827–832, https://doi.org/10.1038/nbt.2647.

[21] R.L. Frock, J. Hu, R.M. Meyers, Y. Ho, E. Kii, F.W. Alt, Genome-wide detection of DNA double-stranded breaks induced by engineered nucleases, Nat. Biotechnol. 33 (2014) 179–186, https://doi.org/10.1038/nbt.3101.

[22] L. Cong, F.A. Ran, D. Cox, S. Lin, R. Barretto, N. Habib, P.D. Hsu, X. Wu, W. Jiang, L.A. Marraffini, F. Zhang, Multiplex genome engineering using CRISPR/Cas systems, Science 339 (2013) 819–823, https://doi.org/10.1126/science.1231143.

[23] P. Mali, L. Yang, K.M. Esvelt, J. Aach, M. Guell, J.E. DiCarlo, J.E. Norville, G.M. Church, RNA-guided human genome engineering via Cas9, Science 339 (2013) 823–826, https://doi.org/10.1126/science.1232033.

[24] S. Abadi, W.X. Yan, D. Amar, I. Mayrose, A machine learning approach for predicting CRISPR–Cas9 cleavage efficiencies and patterns underlying its mechanism of action, PLoS Comput. Biol. 13 (2017), e1005807. https://doi.org/10.1371/journal.pcbi.1005807.

[25] J.G. Doench, N. Fusi, M. Sullender, M. Hegde, E.W. Vaimberg, K.F. Donovan, I. Smith, Z. Tothova, C. Wilen, R. Orchard, H.W. Virgin, J. Listgarten, D.E. Root, Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR–Cas9, Nat. Biotechnol. 34 (2016) 184–191, https://doi.org/10.1038/nbt.3437.

[26] M. Stemmer, T. Thumberger, M. del Sol Keyer, J. Wittbrodt, J.L. Mateo, CCTop: an intuitive, flexible and reliable CRISPR/Cas9 target prediction tool, PLoS One 10 (2015), e0124633. https://doi.org/10.1371/journal.pone.0124633.

[27] Zhang Lab, Optimized CRISPR Design, Mit, 2013 2013, http://crispr.mit.edu/.

[28] M. Spitzer, J. Wildenhain, J. Rappsilber, M. Tyers, correspondEnce E-CRISP: fast CRISPR target site identification, Nat. Publ. Gr. 11 (2014) 122–123, https://doi.org/10.1038/nmeth.2812.

[29] T.G. Montague, J.M. Cruz, J.A. Gagnon, G.M. Church, E. Valen, CHOPCHOP: a CRISPR/Cas9 and TALEN web tool for genome editing, Nucleic Acids Res. 42 (2014) W401–7, https://doi.org/10.1093/nar/gku410.

[30] T.J. Cradick, P. Qiu, C.M. Lee, E.J. Fine, G. Bao, COSMID: a web-based tool for identifying and validating CRISPR/Cas off-target sites, Mol. Ther. Nucleic Acids 3 (2014), e214. https://doi.org/10.1038/mtna.2014.64.

[31] I.M. Slaymaker, L. Gao, B. Zetsche, D.A. Scott, W.X. Yan, F. Zhang, Rationally engineered Cas9 nucleases with improved specificity, Science (80-.) (2015), https://doi.org/10.1126/science.aad5227.

[32] F.A. Ran, L. Cong, W.X. Yan, D.a. Scott, J.S. Gootenberg, A.J. Kriz, B. Zetsche, O. Shalem, X. Wu, K.S. Makarova, E.V. Koonin, P.a. Sharp, F. Zhang, In vivo genome editing using *Staphylococcus aureus* Cas9, Nature 520 (2015) 186–190, https://doi.org/10.1038/nature14299.

[33] B.P. Kleinstiver, V. Pattanayak, M.S. Prew, S.Q. Tsai, N.T. Nguyen, Z. Zheng, J.K. Joung, High-fidelity CRISPR–Cas9 nucleases with no detectable genome-wide off-target effects, Nature 529 (2016) 490–495, https://doi.org/10.1038/nature16526.

[34] M. Endo, M. Mikami, S. Toki, Multi-gene knockout utilizing off-target mutations of the CRISPR/Cas9 system in rice, Plant Cell Physiol. 56 (1) (2015) 41–47

[35] S.V. Prykhozhij, V. Rajan, D. Gaston, J.N. Berman, CRISPR multitargeter: a web tool to find common and unique CRISPR single guide RNA targets in a set of similar sequences, PLoS One 10 (2015), e0119372.

[36] R. Singh, C. Kuscu, A. Quinlan, Y. Qi, M. Adli, D. S., M. E.H., C. Y., B. J.A., G. D., Cas9-chromatin binding information enables more accurate CRISPR off-target prediction, Nucleic Acids Res. 43 (18) (2015) e118, https://doi.org/10.1093/nar/gkv575.

[37] I. Gronau, S. Moran, Optimal implementations of UPGMA and other common clustering algorithms, Inf. Process. Lett. 104 (2007) 205–210, https://doi.org/10.1016/j.ipl.2007.07.002.

[38] P.J.A. Cock, T. Antao, J.T. Chang, B.A. Chapman, C.J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, M.J.L. De Hoon, Biopython: freely available Python tools for computational molecular biology and bioinformatics, Bioinformatics 25 (2009) 1422–1423, https://doi.org/10.1093/bioinformatics/btp163.

[39] D. Plotree, D. Plotgram, PHYLIP-phylogeny inference package (version 3.2), Cladistics 5 (1989) 6.

[40] K. Katoh, D.M. Standley, MAFFT multiple sequence alignment software version 7: improvements in performance and usability, Mol. Biol. Evol. 30 (2013) 772–780.

[41] R.M. Karp, Reducibility among combinatorial problems, Complex. Comput. Comput, Springer 1972, pp. 85–103.

[42] V. Chvatal, A greedy heuristic for the set-covering problem, Math. Oper. Res. 4 (1979) 233–235, https://doi.org/10.1287/moor.4.3.233.

[43] B.P. Kleinstiver, M.S. Prew, S.Q. Tsai, V.V. Topkar, N.T. Nguyen, Z. Zheng, A.P.W. Gonzales, Z. Li, R.T. Peterson, J.-R.J. Yeh, M.J. Aryee, J.K. Joung, Engineered CRISPR–Cas9 nucleases with altered PAM specificities, Nature 523 (2015) 481–485, https://doi.org/10.1038/nature14592.

[44] L.S. Qi, M.H. Larson, L.A. Gilbert, J.A. Doudna, J.S. Weissman, A.P. Arkin, W.A. Lim, Repurposing CRISPR as an RNA-guided platform for sequence-specific control of gene expression, Cell 152 (2013) 1173–1183, https://doi.org/10.1016/j.cell.2013.02.022.

[45] K.S. Makarova, F. Zhang, E.V. Koonin, Snapshot: class 2 CRISPR–Cas systems, Cell 168 (2017) 328.

[46] Q. Zhang, M.J. Lenardo, D. Baltimore, Bacterial CRISPR–Cas systems utilize sequence-specific RNA-guided nucleases to defend against bacteriophage infection. As a countermeasure, numerous phages are known that produce proteins to block the function of class 1 CRISPR–Cas systems. However, curre, Cell 168 (2017) 1–2.

[47] S. Shmakov, A. Smargon, D. Scott, D. Cox, N. Pyzocha, W. Yan, O.O. Abudayyeh, J.S. Gootenberg, K.S. Makarova, Y.I. Wolf, Diversity and evolution of class 2 CRISPR–Cas systems, Nat. Rev. Microbiol. 15 (2017) 169–182.

[48] I. Sela, H. Ashkenazy, K. Katoh, T. Pupko, GUIDANCE2: accurate detection of unreliable alignment regions accounting for the uncertainty of multiple parameters, Nucleic Acids Res. 43 (2015) W7–14, https://doi.org/10.1093/nar/gkv318.

[49] R.C. Edgar, K. Sjolander, E.R. Tillier, S. Brenner, R. Dunbrack, A comparison of scoring functions for protein sequence profile alignment, Bioinformatics 20 (2004) 1301–1308, https://doi.org/10.1093/bioinformatics/bth090.

[50] J.D. Thompson, B. Linard, O. Lecompte, O. Poch, P. McGettigan, A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives, PLoS One 6 (2011), e18093. https://doi.org/10.1371/journal.pone.0018093.

[51] G. Blackshields, I.M. Wallace, M. Larkin, D.G. Higgins, Analysis and comparison of benchmarks for multiple sequence alignment, In Silico Biol. 6 (2006) 321–339 (http://www.ncbi.nlm.nih.gov/pubmed/16922695 (accessed August 9, 2017)).

[52] M. Haeussler, K. Schönig, H. Eckert, A. Eschstruth, J. Mianné, J.-B. Renaud, S. Schneider-Maunoury, A. Shkumatava, L. Teboul, J. Kent, J.-S. Joly, J.-P. Concordet, Evaluation of off-target and on-target scoring algorithms and integration into the guide RNA selection tool CRISPOR, Genome Biol. 17 (2016), 148. https://doi.org/10.1186/s13059-016-1012-2.

[53] Q. Shan, Y. Wang, J. Li, Y. Zhang, K. Chen, Z. Liang, K. Zhang, J. Liu, J.J. Xi, J.-L. Qiu, C. Gao, Targeted genome modification of crop plants using a CRISPR–Cas system, Nat. Biotechnol. 31 (2013) 686–688, https://doi.org/10.1038/nbt.2650.

[54] H. Wang, H. Yang, C.S. Shivalila, M.M. Dawlaty, A.W. Cheng, F. Zhang, R. Jaenisch, One-step generation of mice carrying mutations in multiple genes by CRISPR/Cas-mediated genome engineering, Cell 153 (2013) 910–918, https://doi.org/10.1016/j.cell.2013.04.025.

[55] H. Zhou, B. Liu, D.P. Weeks, M.H. Spalding, B. Yang, Large chromosomal deletions and heritable small genetic changes induced by CRISPR/Cas9 in rice, Nucleic Acids Res. 42 (2014) 10903–10914, https://doi.org/10.1093/nar/gku806.

[56] K. Xie, B. Minkenberg, Y. Yang, Boosting CRISPR/Cas9 multiplex editing capability with the endogenous tRNA-processing system, Proc. Natl. Acad. Sci. U. S. A. 112 (2015) 3570–3575, https://doi.org/10.1073/pnas.1420294112.

[57] P. Wang, J. Zhang, L. Sun, Y. Ma, J. Xu, S. Liang, J. Deng, J. Tan, Q. Zhang, L. Tu, High efficient multi-sites genome editing in allotetraploid cotton (*Gossypium hirsutum*) using CRISPR/Cas9 system, Plant Biotechnol. J. 16 (1) (2018) 137–150.

[58] D. Kim, S. Bae, J. Park, E. Kim, S. Kim, H.R. Yu, J. Hwang, J.I.S. Kim, J.I.S. Kim, Digenome-seq: genome-wide profiling of CRISPR–Cas9 off-target effects in human cells, Nat. Methods 12 (2015) 237–243 (1 p following 243) https://doi.org/10.1038/nmeth.3284.