CrisprCBEI online version

user manual

Directory

1. Introduction

1.1 Base editor

Clustered regularly interspaced short palindromic repeats (CRISPR) and CRISPR-associated (Cas) systems are a countermeasure in most prokaryotes against foreign DNA invasion (Jiang and Marraffini, 2015). The RNA-guided CRISPR-associated nucleases have been developed as versatile and multi-purpose tools for genome editing in a number of species (Chen, et al., 2017; Chen, et al., 2018; Jinek, et al., 2012; Kondo and Ueda, 2013; Li, et al., 2013; Sander and Joung, 2014; Wang, et al., 2018), through their nature to cleave double-strand DNA at a precise and programmable target location in genomes (Jiang and Marraffini, 2015). Moreover, the catalytically dead Cas-nucleases are used as a programmable DNA-binding protein to manipulate the expression of certain genes (Bikard et al., 2013; Dominguez et al., 2016; Mali et al., 2013). In most eukaryotes and a few prokaryotes, double-strand break (DSB) could be repaired by non-homologous end-joining (NHEJ), introducing insertion, deletion, translocation or other DNA re-arrangement, usually resulting in gene disruptions (Jeggo, 1998; Rouet et al., 1994). While more precise gene editing could be achieved by taking advantage of the direct-homology repair (HDR) mechanism by supplementing a repair arm with sequences homologous to the flanking region of the DSB (Chapman et al., 2012; Rudin et al., 1989).

Recently, the development of “base editors” offered a new means in gene editing, manifesting single-nucleotide mutagenesis inside the genome by precise DNA modification without creating any double-strand break in the process (Komor et al., 2016). To date, two major types of base editors have been developed by fusing the nucleotide deaminase and the catalytically inactivated Cas-nucleases or Cas-nickase (Rees and Liu, 2018). The adenine base editors transform deoxyadenosine (dA) to deoxyguanosine (dG) (Gaudelli et al., 2017). While cytosine base editors change deoxycytidine (dC) to thymidine (dT), potentially converting four types of codons, CAA, CGA, CAG (editing on the sense strand) and TGG (editing on the anti-sense strand) into stop codon, which are further exploited to disrupt specific genes (Komor, et al., 2016; Nishida, et al., 2016). This technology allows gene therapeutics to treat genetic disease caused by single-nucleotide polymorphism, which also opens a new avenue for genome editing in species with weak intrinsic homologous recombination capacity and without NHEJ repair mechanism, such as *Staphylococcus aureus*, *Pseudomonas aeruginosa* and *Klebsiella pneumoniae* (Chen, et al., 2018;

Gu, et al., 2018; Wang, et al., 2018). However, searching potential sites for generating a pre-stop codon by cytosine base editor is considerably complicated and time-consuming with additional restrictions. To our knowledge, currently, there is no software dedicated to predicting the gene inactivation by cytosine base editors.

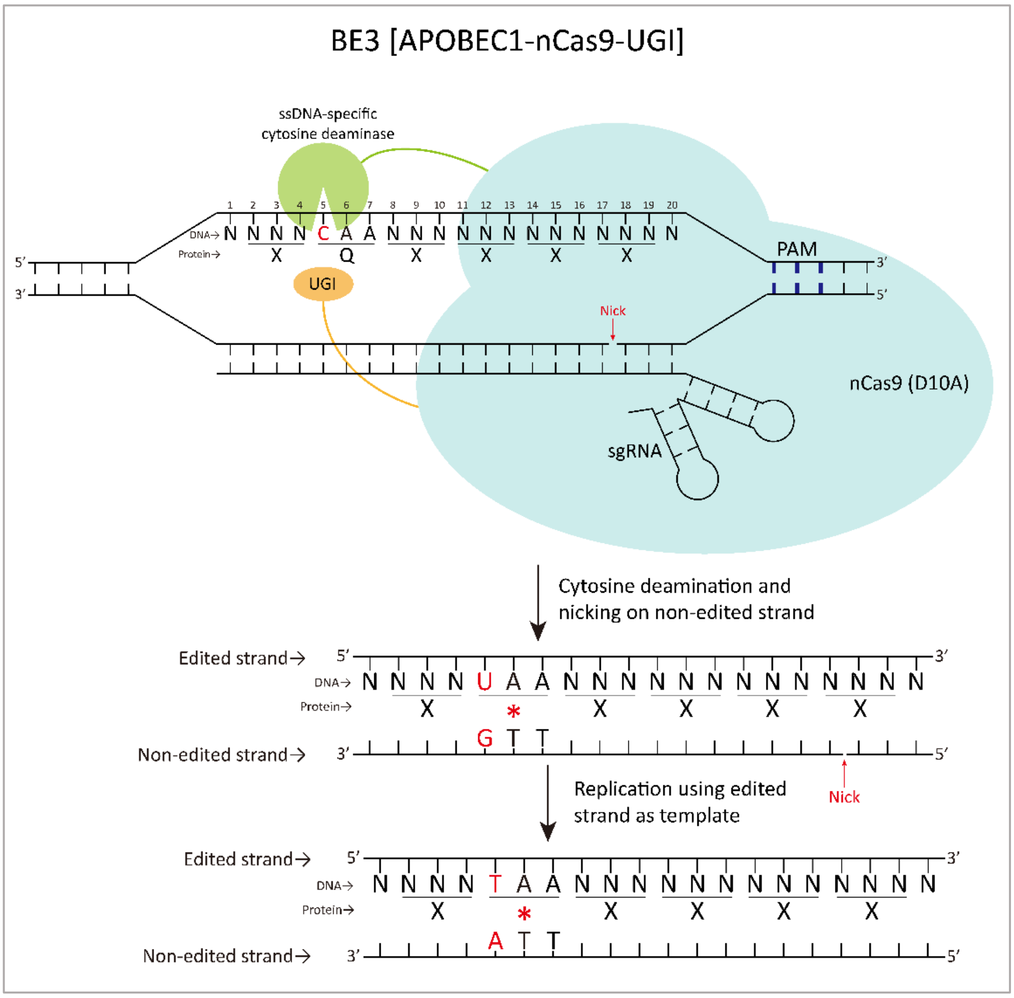

BE3 [APOBEC1-nCas9-UGI]
ssDNA-specific cytosine deaminase
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
DNA→ N N N N C A A N N N N N N N N N N N N N
Protein→ X Q X X X X
5′
3′
UGI
PAM
3′
5′
Nick
nCas9 (D10A)
sgRNA
Cytosine deamination and nicking on non-edited strand
Edited strand→ 5′ 3′
DNA→ N N N N U A A N N N N N N N N N N N N N
Protein→ X * X X X X
Non-edited strand→ 3′ G T T 5′
Replication using edited strand as template
Nick
Edited strand→ 5′ 3′
DNA→ N N N N T A A N N N N N N N N N N N N N
Protein→ X * X X X X
Non-edited strand→ 3′ A T T 5′

Figure 1. Schematic overview of base editing.

Here, we have developed a new, easily accessible, and user-friendly web-based computational web-tool, CrisprCBEI, facilitating the design of sgRNAs for Cytosine Base Editor-mediated gene Inactivation. All currently available CBEs have been included in the built-in choice of CrisprCBEI. Moreover, the length, sequence specificity and location of PAMs are fully customizable. It is worth mentioning that the off-target prediction function of CrisprCBEI is a front-end prediction tool that supports local Fasta format files for off-target prediction without uploading to the server. Through algorithm optimization, CrisprCBEI does not limit the size of Fasta files, and the spacer alignment speed is fast. The whole off-target calculation process is asynchronous and does not occupy much memory (about 200 MB), so it would not affect other applications of the computer.

1.2 CrisprCBEI overview

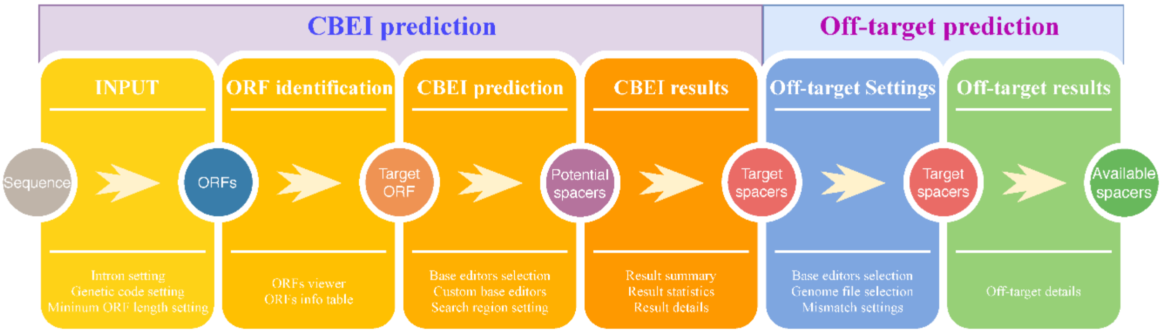


Figure 2. Schematic overview of CrisprCBEI.

The main functions of CrisprCBEI are CBEI-spacer design and off-target prediction. Users should first submit a sequence containing a coding gene. After that, potential ORFs are first detected the software. Once a target ORF has been specified, CBEI-spacers are predicted for gene inactivation. Finally, a local genome file should be selected to verify the potential off-target effect of the predicted spacers. We believe that the default parameters in Crispr-CBEI and the off-target prediction should cover the need of most users. Also, we provide flexible customization options:

Intron settings: The default option of the input sequence is a complete CDS, which should not include introns. Although, when inputting a sequence containing introns, the user can set the position of introns, and CrisprCBEI will remove the introns first before ORF detection. Of Note, the software itself does not predict introns and does not support variable introns splicing. We recommend users to copy the locations of introns directly from a GenBank record (detail in 2.1.2).

ORF detection settings: When we designed ORF recognition, we used the ORF Finder of NCBI as a benchmark. Hence, we include 33 built-in genetic code tables. Moreover, the user can customize the start and end codons, and the minimum ORF length.

Base editor settings: CrisprCBEI has 13 cytosine base editors built-in, and users can also customize their base-editor parameters. We also support customizable PAM sequences, the location of PAM, the length of the spacer, and the edit window. It is fair to say that we support all existing and future base editors.

Moreover, we support the interactive chart presentation and diverse data export methods.

1.3 CrisprCBEI workflow

CrisprCBEI is a pure front-end tool. It adopts HTML5, CSS for the layout, and the calculation process is completed by JavaScripts. Off-target prediction is done by web worker, a HTML5 feature that supports reading local files.

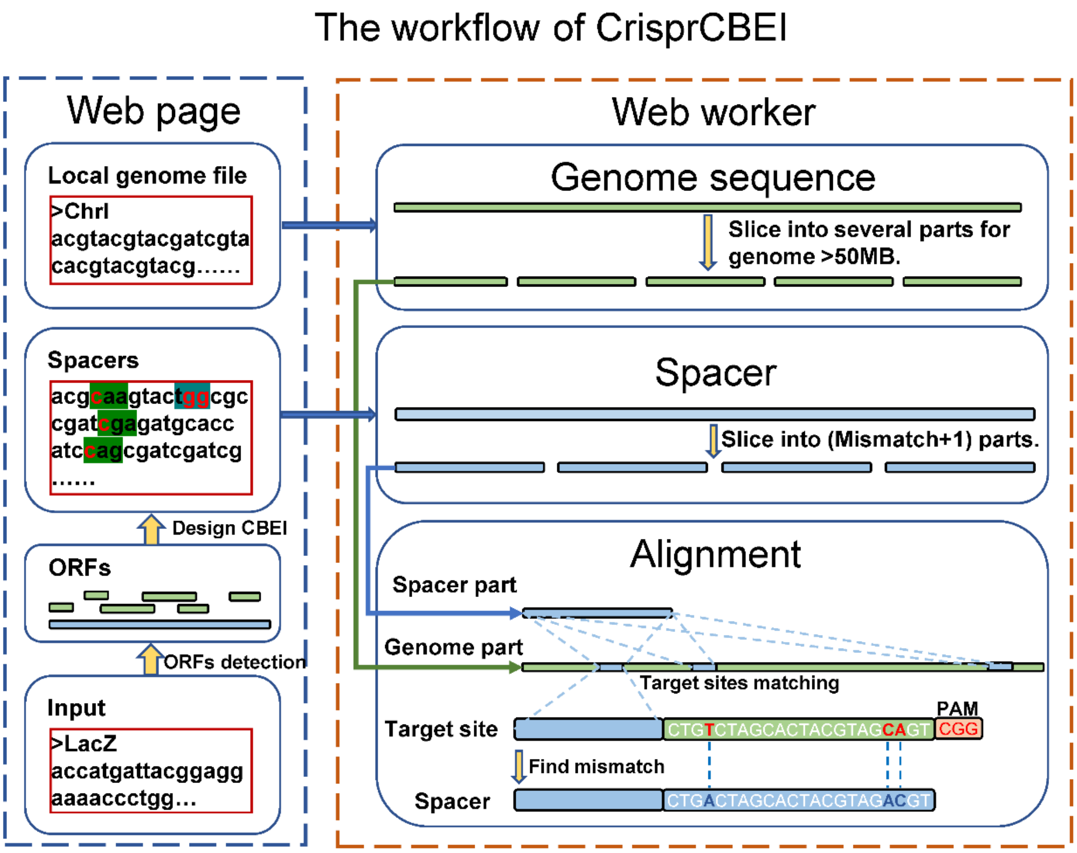


Figure 3. The workflow of CrisprCBEI.

HTML web workers were adopted so that it could be calculated locally without having to upload large genomic files to the server. Genome files larger than 50MB will be cut into 50MB parts for calculation and the off-target near the cutting site was considered in the algorithm. The calculation efficiency depends on CPU performance, genome size and the ‘mismatch value’. If the Mismatch value is set to 0, a regular expression is used to match the genome. And if the Mismatch is greater than 0, the spacer was first divided into (Mismatch + 1) parts, each of which serves as an anchor spacer for matching in the genome. If the anchor spacer matches, then determine whether the mismatch value of other parts is lower than the set value and whether it matches the set base editor.

1.4 CrisprCBEI tool kits

CrisprCBEI is available in three versions to accommodate three usage scenarios: online, offline, and command line. All three versions have the function to design cytosine base editor mediated gene inactivation (CBEI). In addition, the web-based versions (both online and offline) also support ORF detection and off-target prediction on the web page. For the ‘online version’, users can easily access through the website (https://taolab.nwsuaf.edu.cn/CrisprCBEI/). Next came the offline web version for offline computers, where we introduced the ‘HTML version’ and the ‘local server version’. CrisprCBEI is a pure front-end computing tool, so in the HTML version, users can click directly on the HTML file to run it in the browser. However, due to browser security settings, the HTML version only supports a few browsers, so we introduced the local server version. The local server version of CrisprCBEI uses a lightweight server framework, Flask, that allows users to set up a local server in a few simple command lines and to use CrisprCBEI in main browsers. Finally, we have released a command-line version that can efficiently predict a large number of ORFs. The source code and detailed instructions of the above versions are stored in Github for open access (https://github.com/atlasbioinfo/CrisprCBEI).

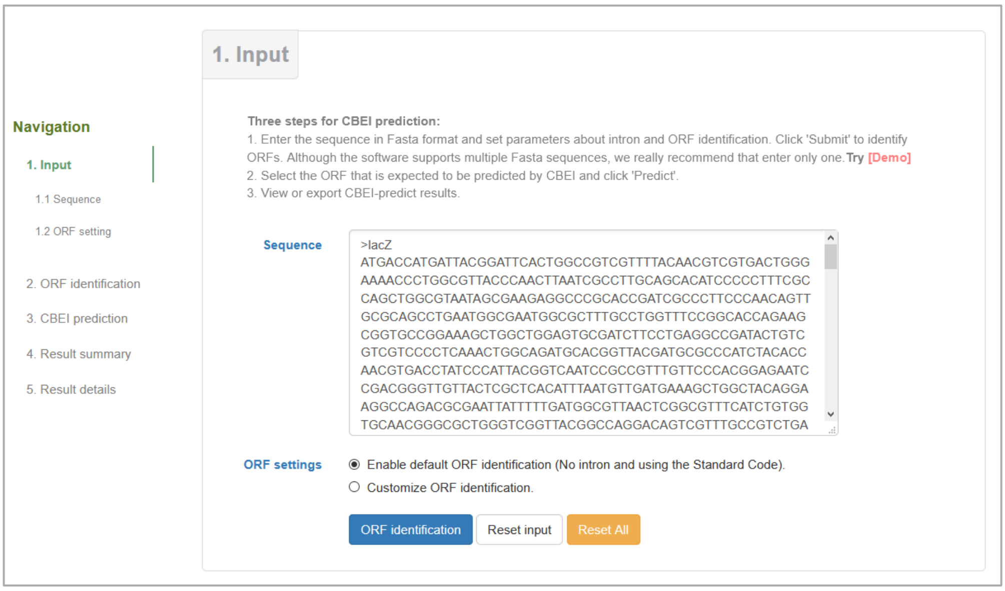
2. CBEI design

Three steps for CBEI design:

- Enter the sequence in Fasta format and set parameters about intron and ORF detection. Click ‘Submit’ to identify ORFs. Although the software supports multiple Fasta sequences, but we recommend that users enter only one.
- Select the ORF of interest and click ‘Predict’.
- View or export CBEI-predict results.

2.1 Input

The ‘Input’ could divide into two parts: Sequence input and ORF recognition setting.



1. Input

Three steps for CBEI prediction:
1. Enter the sequence in Fasta format and set parameters about intron and ORF identification. Click 'Submit' to identify ORFs. Although the software supports multiple Fasta sequences, we really recommend that enter only one. **Try** [Demo]
2. Select the ORF that is expected to be predicted by CBEI and click 'Predict'.
3. View or export CBEI-predict results.

Sequence

```
>lacZ
ATGACCATGATTACGGATTCACTGGCCGTCGTTTTACAACGTCGTGACTGGG
AAAACCCTGGCGTTACCCAACTTAATCGCCTTGCAGCACATCCCCCTTTCGC
CAGCTGGCGTAATAGCGAAGAGGCCCGCACCGATCGCCCTTCCCAACAGTT
GCGCAGCCTGAATGGCGAATGGCGCTTTGCCTGGTTTCCGGCACCAGAAG
CGGTGCCGGAAAGCTGGCTGGAGTGCGATCTTCCTGAGGCCGATACTGTC
GTCGTCCCCTCAAACTGGCAGATGCACGGTTACGATGCGCCCATCTACACC
AACGTGACCTATCCCATTACGGTCAATCCGCCGTTTGTTCCCACGGAGAATC
CGACGGGTTGTTACTCGCTCACATTTAATGTTGATGAAAGCTGGCTACAGGA
AGGCCAGACGCGAATTATTTTTGATGGCGTTAACTCGGCGTTTCATCTGTGG
TGCAACGGGCGCTGGGTCGGTTACGGCCAGGACAGTCGTTTGCCGTCTGA
```

ORF settings

● Enable default ORF identification (No intron and using the Standard Code).
○ Customize ORF identification.

Figure 4. The input interfaces.

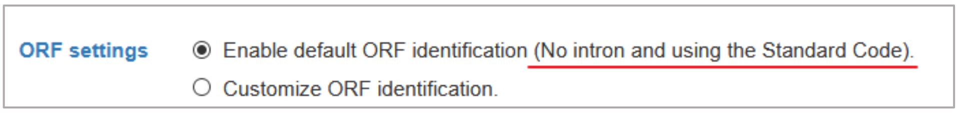
2.1.1 Sequence input

We support Fasta sequences in plain text format. Users can also paste a sequence without Fasta header (‘line starts with >’), but it is not recommended. The entered Fasta sequence would be first processed by ORF detection, so the user can enter the sequence with UTR regions and select the desired ORF in the next step.

The ‘Demo’ sequence built in the software is the CDS of the *lacZ* gene in *E. coli*.

2.1.2 Intron settings

The default setting of an input is a Fasta sequence containing **one complete CDS (*i.e.*, no introns) with the standard genetic codon (start at ATG, and stop at TAA, TGA, and TAG)** for the following ORF recognition.



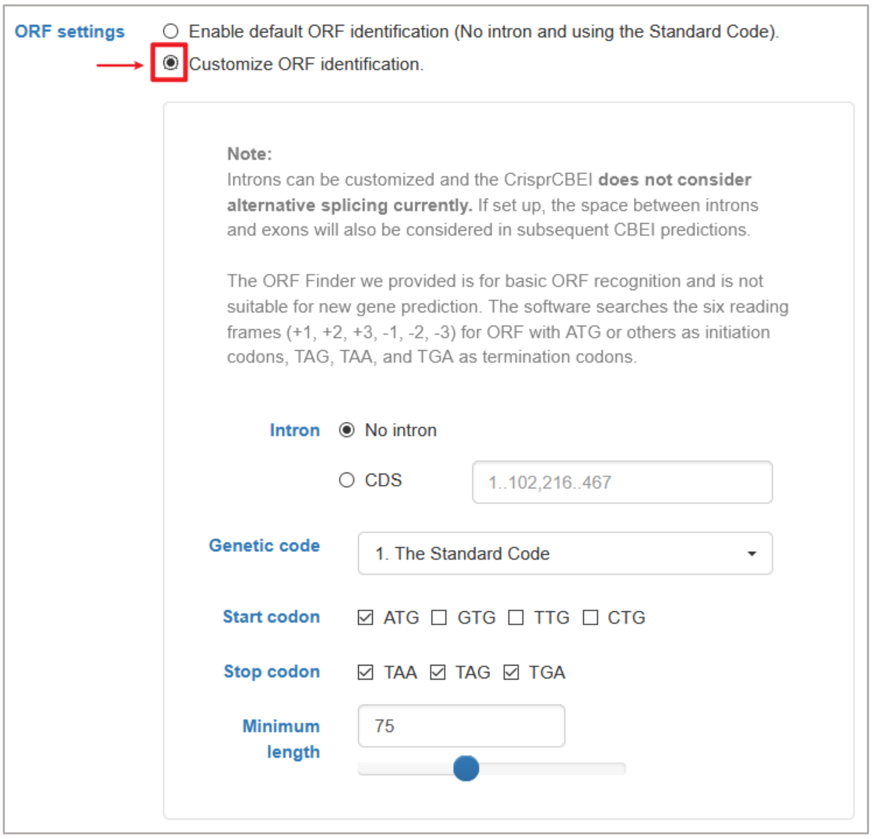
ORF settings

● Enable default ORF identification (No intron and using the Standard Code).

○ Customize ORF identification.

Figure 5. Default options for ORF Settings

If the user has other requirements, they can customize the ORF settings.



ORF settings

○ Enable default ORF identification (No intron and using the Standard Code).

● Customize ORF identification.

Note:
Introns can be customized and the CrisprCBEI **does not consider alternative splicing currently.** If set up, the space between introns and exons will also be considered in subsequent CBEI predictions.

The ORF Finder we provided is for basic ORF recognition and is not suitable for new gene prediction. The software searches the six reading frames (+1, +2, +3, -1, -2, -3) for ORF with ATG or others as initiation codons, TAG, TAA, and TGA as termination codons.

Intron: ● No intron ○ CDS 1..102,216..467

Genetic code: 1. The Standard Code

Start codon: ☑ ATG ☐ GTG ☐ TTG ☐ CTG

Stop codon: ☑ TAA ☑ TAG ☑ TGA

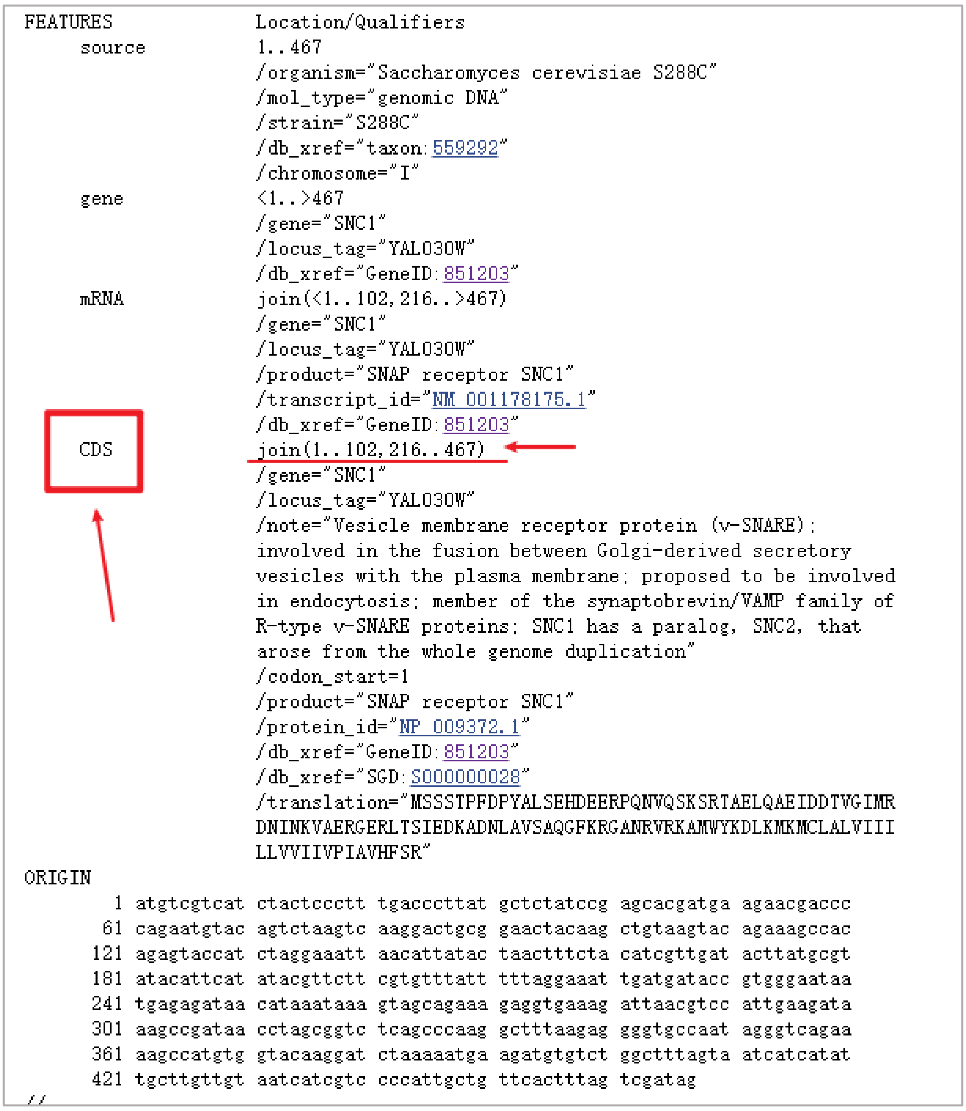
Minimum length: 75

Figure 6. Customize options for ORF detection.

Intron settings: The user can set the position of the CDS so that CrisprCBEI will be splice the sequence before the ORF recognition.

Example (Gene: SNC1, Species: *Saccharomyces cerevisiae*):

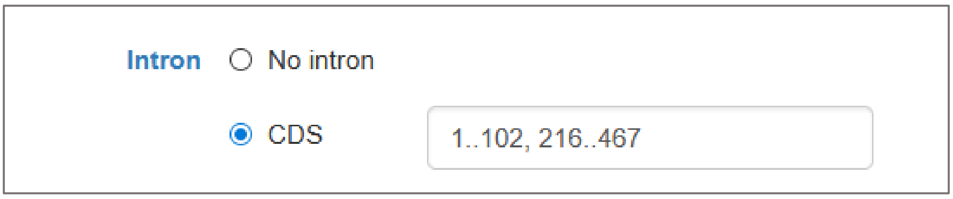
Search **SNC1** in NCBI and open the GenBank entry. After finding the “CDS”, copy its location (1..102, 216..467).



```
FEATURES             Location/Qualifiers
     source          1..467
                     /organism="Saccharomyces cerevisiae S288C"
                     /mol_type="genomic DNA"
                     /strain="S288C"
                     /db_xref="taxon:559292"
                     /chromosome="I"
     gene            <1..>467
                     /gene="SNC1"
                     /locus_tag="YAL030W"
                     /db_xref="GeneID:851203"
     mRNA            join(<1..102,216..>467)
                     /gene="SNC1"
                     /locus_tag="YAL030W"
                     /product="SNAP receptor SNC1"
                     /transcript_id="NM_001178175.1"
                     /db_xref="GeneID:851203"
     CDS             join(1..102,216..467)
                     /gene="SNC1"
                     /locus_tag="YAL030W"
                     /note="Vesicle membrane receptor protein (v-SNARE);
                     involved in the fusion between Golgi-derived secretory
                     vesicles with the plasma membrane; proposed to be involved
                     in endocytosis; member of the synaptobrevin/VAMP family of
                     R-type v-SNARE proteins; SNC1 has a paralog, SNC2, that
                     arose from the whole genome duplication"
                     /codon_start=1
                     /product="SNAP receptor SNC1"
                     /protein_id="NP_009372.1"
                     /db_xref="GeneID:851203"
                     /db_xref="SGD:S000000028"
                     /translation="MSSSTPFDPYALSEHDEERPQNVQSKSRTAELQAEIDDTVGIMR
                     DNINKVAERGERLTSIEDKADNLAVSAQGFKRGANRVRKAMWYKDLKMKMCLALVIII
                     LLVVIIVPIAVHFSR"
ORIGIN
        1 atgtcgtcat ctactccctt tgacccttat gctctatccg agcacgatga agaacgaccc
       61 cagaatgtac agtctaagtc aaggactgcg gaactacaag ctgtaagtac agaaagccac
      121 agagtaccat ctaggaaatt aacattatac taactttcta catcgttgat acttatgcgt
      181 atacattcat atacgttctt cgtgtttatt tttaggaaat tgatgatacc gtgggaataa
      241 tgagagataa cataaataaa gtagcagaaa gaggtgaaag attaacgtcc attgaagata
      301 aagccgataa cctagcggtc tcagcccaag gctttaagag gggtgccaat agggtcagaa
      361 aagccatgtg gtacaaggat ctaaaaatga agatgtgtct ggctttagta atcatcatat
      421 tgcttgttgt aatcatcgtc cccattgctg ttcactttag tcgatag
//
```

Figure 7. GenBank entry of the SNC1 gene in *Saccharomyces cerevisiae*.

User can just copy the location of the CDS in the GenBank entry and paste it into CrisprCBEI.



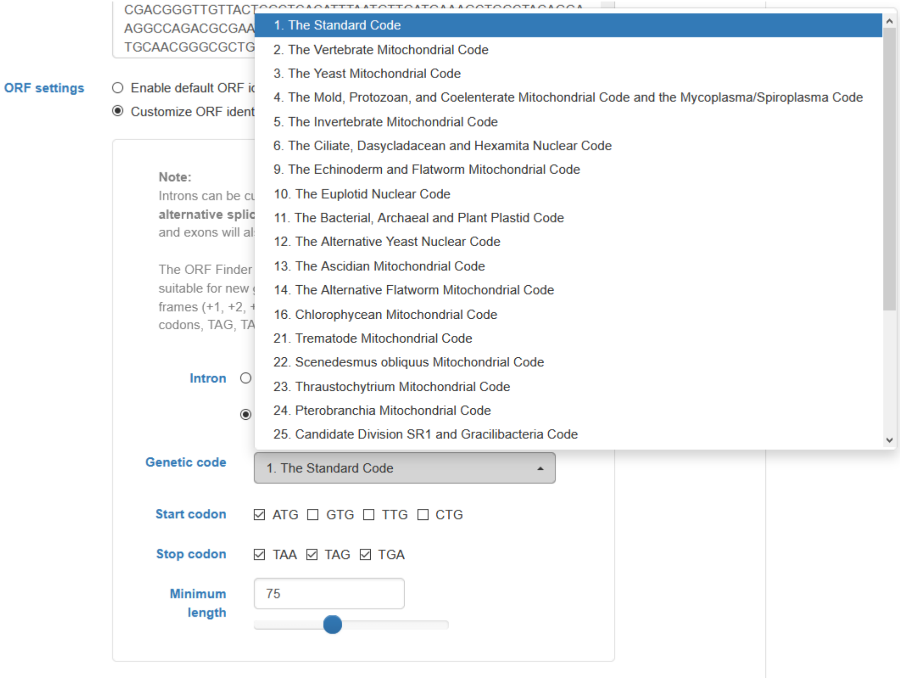
Intron ○ No intron

● CDS 1..102, 216..467

Figure 8. Input the location of CDS.

2.1.3 ORF detection settings

When we design the ORF recognition function, we used the ORF Finder of NCBI as a benchmark. Therefore, we include 33 built-in genetic code tables. Moreover, the user can customize the start and end codons, and the minimum ORF length.



CGACGGGTTGTTACT
AGGCCAGACGCGAA
TGCAACGGGCGCTG

ORF settings ○ Enable default ORF ic
● Customize ORF ident

Note:
Introns can be cu
alternative splic
and exons will al

The ORF Finder
suitable for new
frames (+1, +2, +
codons, TAG, TA

Intron ○
●

1. The Standard Code
2. The Vertebrate Mitochondrial Code
3. The Yeast Mitochondrial Code
4. The Mold, Protozoan, and Coelenterate Mitochondrial Code and the Mycoplasma/Spiroplasma Code
5. The Invertebrate Mitochondrial Code
6. The Ciliate, Dasycladacean and Hexamita Nuclear Code
9. The Echinoderm and Flatworm Mitochondrial Code
10. The Euplotid Nuclear Code
11. The Bacterial, Archaeal and Plant Plastid Code
12. The Alternative Yeast Nuclear Code
13. The Ascidian Mitochondrial Code
14. The Alternative Flatworm Mitochondrial Code
16. Chlorophycean Mitochondrial Code
21. Trematode Mitochondrial Code
22. Scenedesmus obliquus Mitochondrial Code
23. Thraustochytrium Mitochondrial Code
24. Pterobranchia Mitochondrial Code
25. Candidate Division SR1 and Gracilibacteria Code

Genetic code: 1. The Standard Code

Start codon: ☑ ATG ☐ GTG ☐ TTG ☐ CTG

Stop codon: ☑ TAA ☑ TAG ☑ TGA

Minimum length: 75

Figure 9. ORF settings.

2.2 ORF detection

This section presents the ORF detection results in interactive graphics and tables. Users need to select a target ORF and then proceed to the CBEI design.

2.2.1 ORF viewer

The results of ORF detection are presented in six potential open reading boxes (+1, +2, +3, -1, -2, -3).

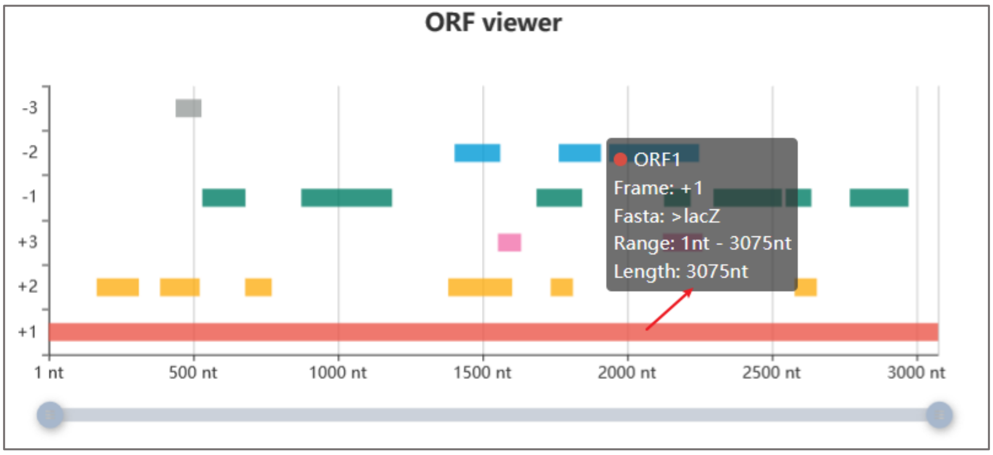
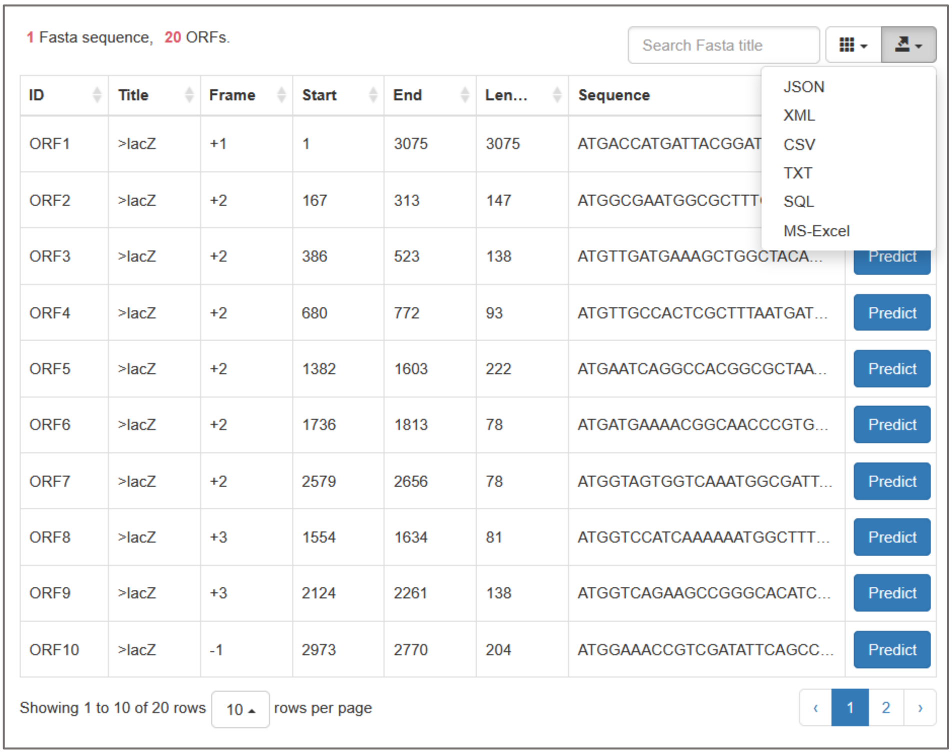


Figure 10. ORF viewer.

Graphics support zoom, mouse hover, mouse click. ORF details can be displayed after the mouse hover. Click on an ORF of interest will go directly to the next part of CBEI design.

2.2.2 ORF info table

The ORF details are displayed in the table.



1 Fasta sequence, 20 ORFs.

ID	Title	Frame	Start	End	Len...	Sequence	
ORF1	>lacZ	+1	1	3075	3075	ATGACCATGATTACGGAT	
ORF2	>lacZ	+2	167	313	147	ATGGCGAATGGCGCTTT	
ORF3	>lacZ	+2	386	523	138	ATGTTGATGAAAGCTGGCTACA...	Predict
ORF4	>lacZ	+2	680	772	93	ATGTTGCCACTCGCTTTAATGAT...	Predict
ORF5	>lacZ	+2	1382	1603	222	ATGAATCAGGCCACGGCGCTAA...	Predict
ORF6	>lacZ	+2	1736	1813	78	ATGATGAAAACGGCAACCCGTG...	Predict
ORF7	>lacZ	+2	2579	2656	78	ATGGTAGTGGTCAAATGGCGATT...	Predict
ORF8	>lacZ	+3	1554	1634	81	ATGGTCCATCAAAAAATGGCTTT...	Predict
ORF9	>lacZ	+3	2124	2261	138	ATGGTCAGAAGCCGGGCACATC...	Predict
ORF10	>lacZ	-1	2973	2770	204	ATGGAAACCGTCGATATTCAGCC...	Predict

Showing 1 to 10 of 20 rows 10 rows per page

Figure 11. ORF info table.

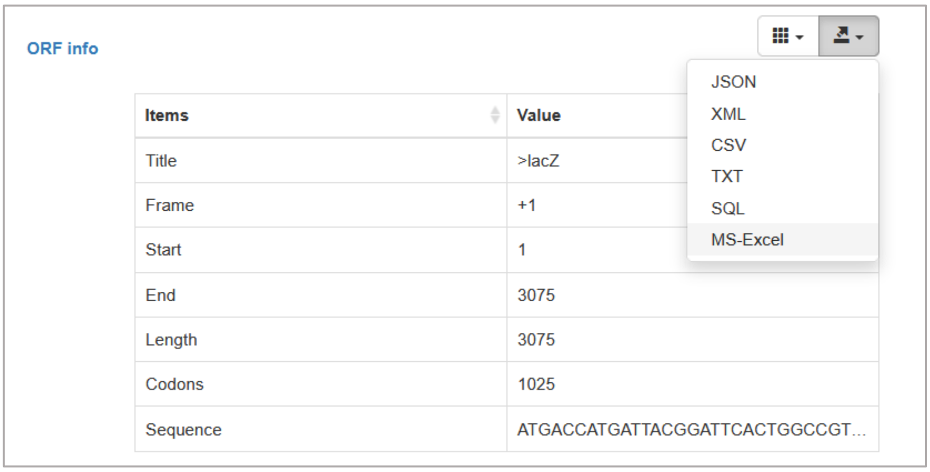
Charts support sorting (click column name), paging, custom columns, and data export. Data can be exported in six standard formats.

2.3 CBEI design

This step uses the ORF you selected in the previous step for CBEI (**C**ytosine **B**ase **E**ditor mediated gene **I**nactivation) design. Since the judgment was made in the previous step, the ORF reading box here is designated as ‘+1’. Through this step, potential editing sites that cause gene inactivation could be predicted.

2.3.1 ORF info

The information about selected ORF is presented in a table.



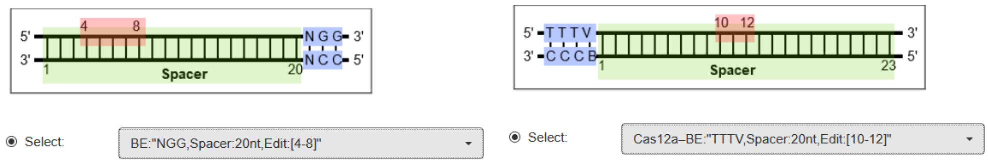
ORF info

Items	Value
Title	>lacZ
Frame	+1
Start	1
End	3075
Length	3075
Codons	1025
Sequence	ATGACCATGATTACGGATTCACTGGCCGT...

Figure 12. ORF info.

2.3.2 Base editor settings

We have built-in 13 common cytosine base editors (CBEs). The diagram above changes when the user selects different CBEs.



● Select: BE:"NGG,Spacer:20nt,Edit:[4-8]"

● Select: Cas12a–BE:"TTTV,Spacer:20nt,Edit:[10-12]"

Figure 13. Base editor selection.

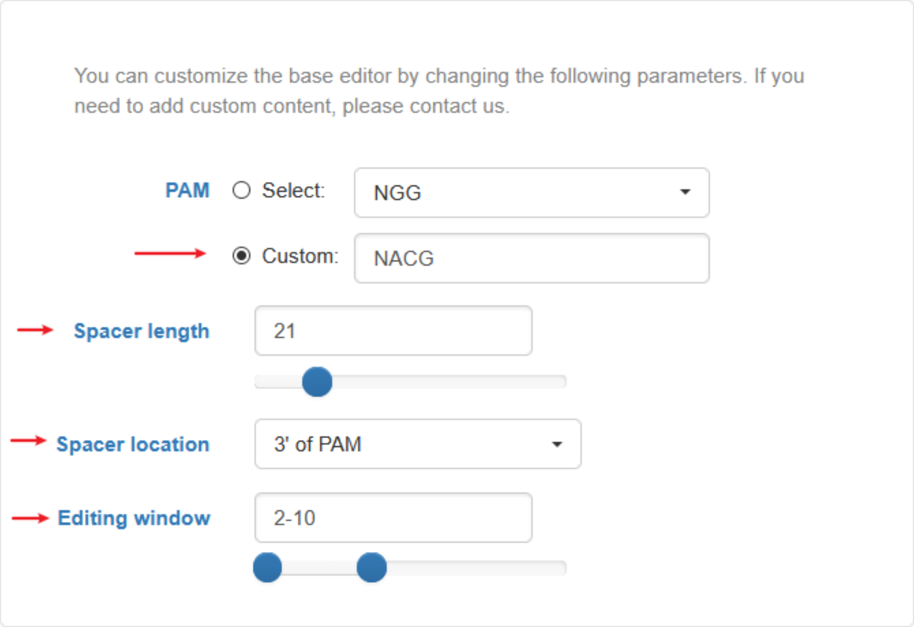
In addition, the essential feature of this software is that we support custom base editor.

For example, a hypothetic base editor has been invented with the following parameters:

Table 1. Hypothetical parameters.

PARAMETERS	VALUES
PAM	NACG
SPACER LENGTH	21
SPACER LOCATION	3' of PAM
EDITING WINDOW	2-10

The above parameters are only examples and do not necessarily exist. Users can adjust these parameters according to their own needs.



You can customize the base editor by changing the following parameters. If you need to add custom content, please contact us.

PAM ○ Select: NGG

● Custom: NACG

Spacer length 21

Spacer location 3' of PAM

Editing window 2-10

Figure 14. Customize base editor.

Since ‘NACG’ does not belong to the common PAM (we have 7 standard PAM accumulators built-in), ‘NACG’ needs to be input. Spacer length can be achieved by dragging the slider or entering the corresponding value, such as 21. Spacer location has two types, ‘3' of PAM’ and ‘5' of PAM’. Finally, you can enter or drag the slider to resize Edit Windows. Updates of the corresponding element immediately show on the schematic.

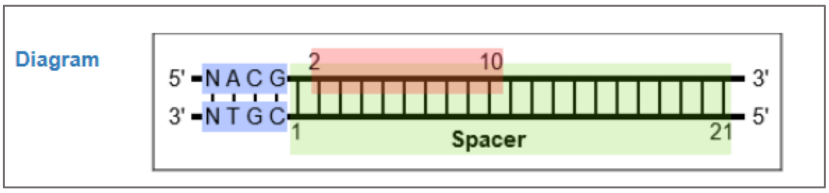
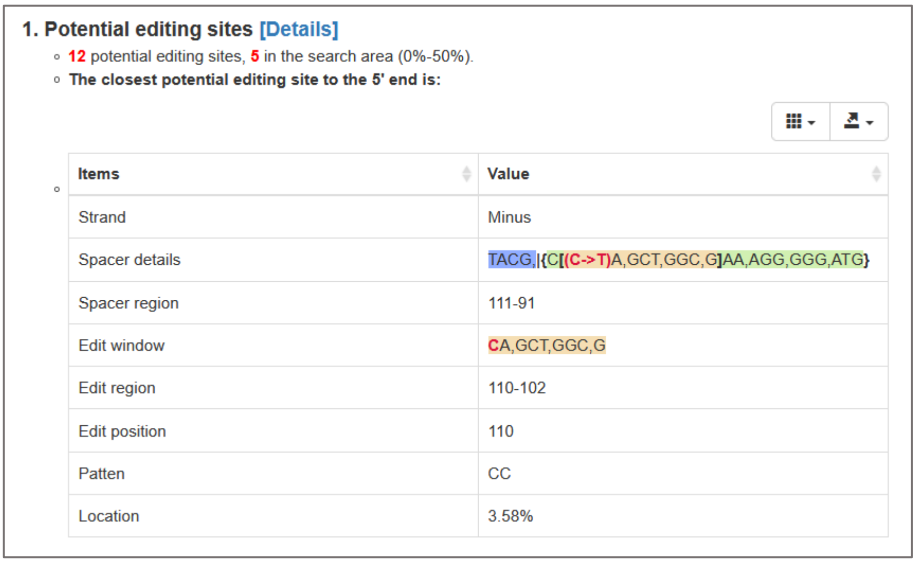


Figure 15. Diagram of the customized base editor.

Theoretically, users can customize any type of base editor with a simple setup.

2.3.3 Result of CBEI design

Click on the ‘CBEI-Predict’ button, and the results will be displayed.



1. Potential editing sites [Details]

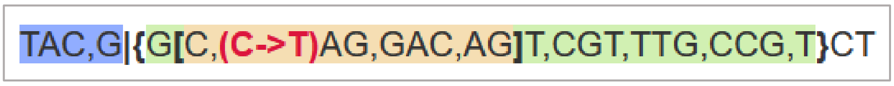
- 12 potential editing sites, 5 in the search area (0%-50%).
- **The closest potential editing site to the 5' end is:**

Items	Value
Strand	Minus
Spacer details	TACG,|{C[(C->T)A,GCT,GGC,G]AA,AGG,GGG,ATG}
Spacer region	111-91
Edit window	CA,GCT,GGC,G
Edit region	110-102
Edit position	110
Patten	CC
Location	3.58%

Figure 16. The best result.

The best result is showed (a potential editing site closest to the 5'end). All potential edit bit points and edit bit points in the search area are marked in red (12 potential editing sites, 5 in the search area). Because the default search range is the first 50% of an ORF. **In some instances, potential edit sites have been identified, but no result appears, which indicated that all potential edit sites lay outside the default search region. Hence, users should expand the search area.**

The potential spacers were marked by background color and symbols.



TAC,G|{G[C,(C->T)AG,GAC,AG]T,CGT,TTG,CCG,T}CT

Figure 17. Potential spacer.

The codon in the spacer sequence is separated by commas. The '|' symbol is used to separate spacer and PAM sequences. Curly braces are used to mark the spacer sequence; square brackets are used to mark the 'edit window'. Brackets indicate the cytosine, which could be potentially changed to thymidine. In this case, a CAG codon is changed to a termination codon, TAG, thus causing gene inactivation.

We also made a statistical analysis of the results.

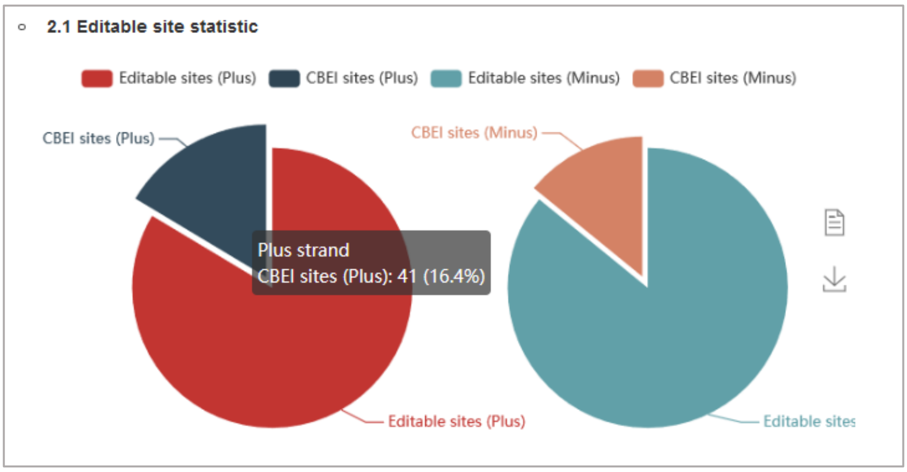


Figure 18. Editable site statistics.

We use a pie chart to show the ratio of the potential CBEI site to the total editable site in the plus/minus chain of ORF. In this case, the potential CBEI account for 16.4% of the total number of total editable sites.

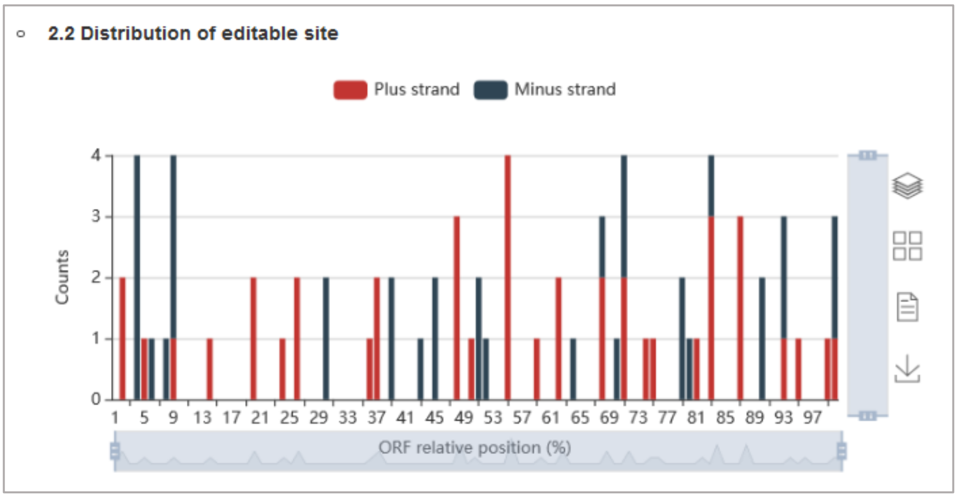
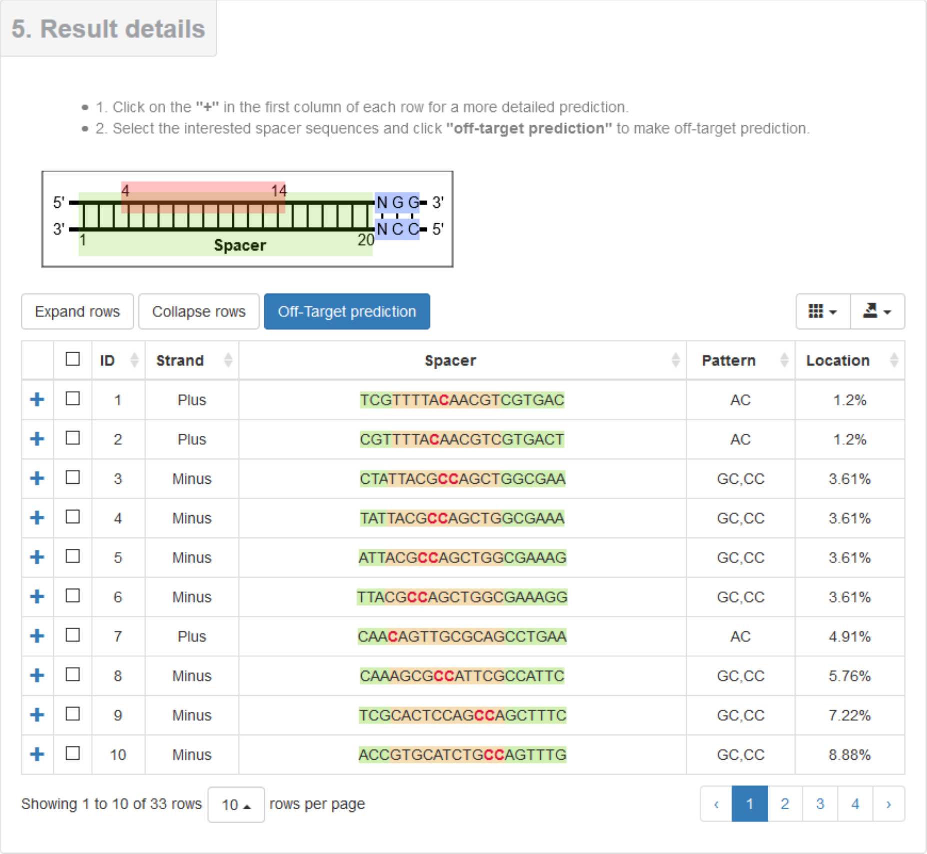


Figure 19. Distribution of CBEI site.

The distribution of CBEI sites is also shown in the figure.



5. Result details

- 1. Click on the **"+"** in the first column of each row for a more detailed prediction.
- 2. Select the interested spacer sequences and click **"off-target prediction"** to make off-target prediction.

	□	ID	Strand	Spacer	Pattern	Location
+	□	1	Plus	TCGTTTTACAACGTCGTGAC	AC	1.2%
+	□	2	Plus	CGTTTTACAACGTCGTGACT	AC	1.2%
+	□	3	Minus	CTATTACGCCAGCTGGCGAA	GC,CC	3.61%
+	□	4	Minus	TATTACGCCAGCTGGCGAAA	GC,CC	3.61%
+	□	5	Minus	ATTACGCCAGCTGGCGAAAG	GC,CC	3.61%
+	□	6	Minus	TTACGCCAGCTGGCGAAAGG	GC,CC	3.61%
+	□	7	Plus	CAACAGTTGCGCAGCCTGAA	AC	4.91%
+	□	8	Minus	CAAAGCGCCATTCGCCATTC	GC,CC	5.76%
+	□	9	Minus	TCGCACTCCAGCCAGCTTTC	GC,CC	7.22%
+	□	10	Minus	ACCGTGCATCTGCCAGTTTG	GC,CC	8.88%

Showing 1 to 10 of 33 rows 10 rows per page

Figure 20. Details of CBEI design.

Finally, the detail info of CBEI design results is shown. Each plus "+" click displays details about the spacer. The table supports export in various formats.

3. Off-target prediction

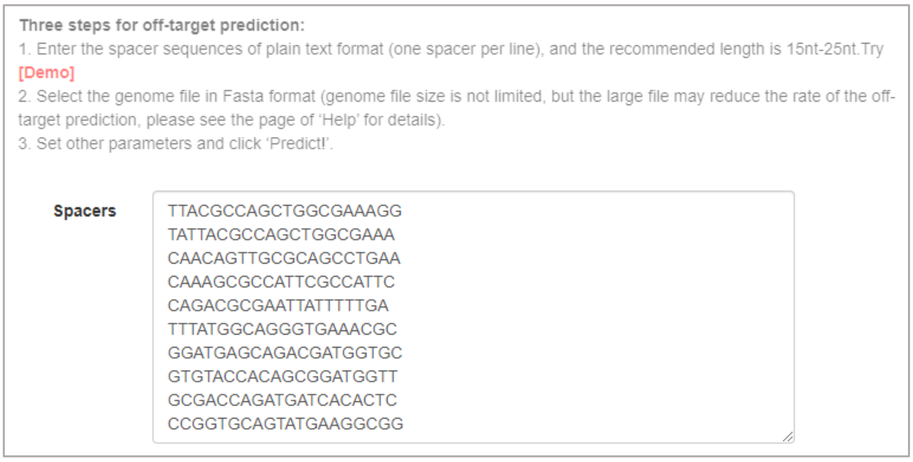
The Crispr-CBEI off-target prediction was based on local computing without uploading it to the server. Therefore, the user could flexibly select any size of Fasta file for off-target prediction. The calculation process was completed with JavaScript script, the "FileReader" and the "Web worker" of HTML5 features was used for file reading and off-target background calculation. Crispr-CBEI does not limit the size of the Fasta file. Because the selected Fasta format file is larger than 50 MB, it would be divided into several parts of 50 MB size for off-target prediction. We have also optimized the spacer alignment algorithm so that the alignment could be completed quickly. If the off-target prediction parameter, "Mismatch" value, is "0", it used the regular expression to look for an exact match in the Fasta file. If the "Mismatch" value was greater than 0, (*i.e.*, The mismatch number of the potential off-target site needs to be less than or equal to the "Mismatch" value), it splits the spacer sequence into “Mismatch +1” short sequences of equal length, because at least one of these short spacers was a perfect match for the potential off-target sequence. Then, these short spacers would be matched entirely in the Fasta file to obtain the target sites, and the local sequence corresponding to each target site was matched one-to-one with the spacer sequence, subsequently. If the mismatch number less than or equal to set “Mismatch value” and the PAM sequence of target sites matched the selected base editor pattern, the predicted result was shown. We have tested the working time for different size genomes. When the “Mismatch” was set as 3, the working time for a single spacer sequence searching in the *E. coli* genome (4.50 MB) was 0.17 seconds, and the mouse genome (2.58 GB) was 74.09 seconds, and the human genome (2.92 GB) was 84.25 seconds. In the case of a small “Mismatch” value, it also cost less time. When the “Mismatch” value was set to 0, it only took 18.38 seconds for a single spacer sequence searching in the human genome.

The source code can be accessed on Github (https://github.com/atlasbioinfo/CrisprCBEI/).

3.1 Off-target settings

This section includes setting up the spacer sequence, the base editor, and the genome file.

3.1.1 Spacers input



Three steps for off-target prediction:

1. Enter the spacer sequences of plain text format (one spacer per line), and the recommended length is 15nt-25nt.Try [Demo]

2. Select the genome file in Fasta format (genome file size is not limited, but the large file may reduce the rate of the off-target prediction, please see the page of ‘Help’ for details).

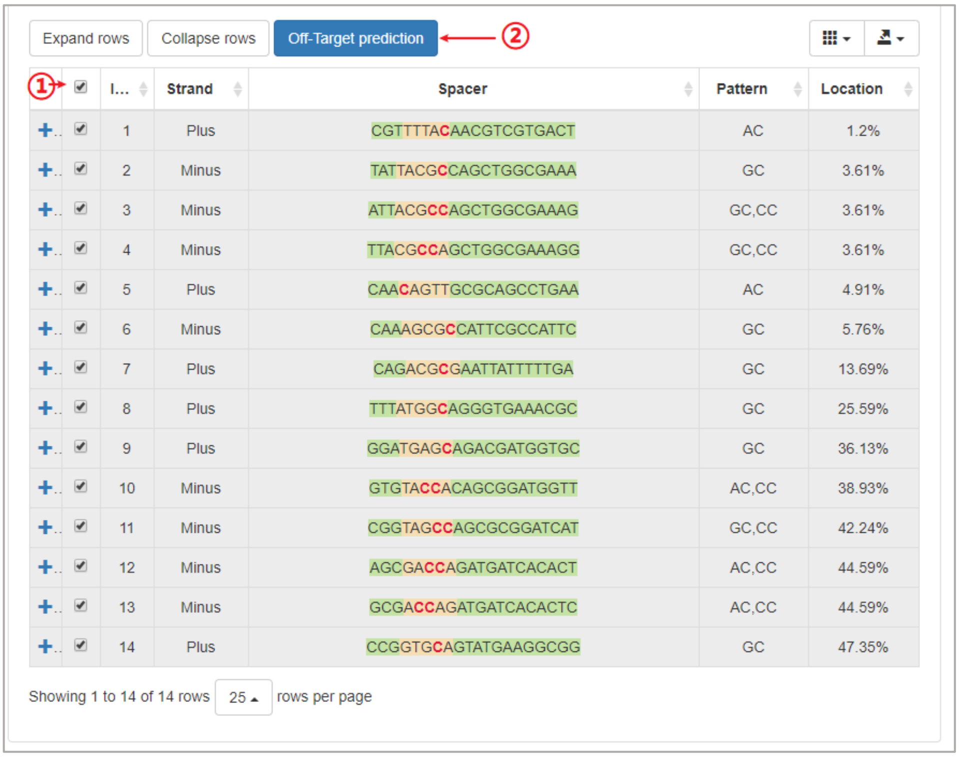
3. Set other parameters and click ‘Predict!’.

Spacers

```
TTACGCCAGCTGGCGAAAGG
TATTACGCCAGCTGGCGAAA
CAACAGTTGCGCAGCCTGAA
CAAAGCGCCATTCGCCATTC
CAGACGCGAATTATTTTTGA
TTTATGGCAGGGTGAAACGC
GGATGAGCAGACGATGGTGC
GTGTACCACAGCGGATGGTT
GCGACCAGATGATCACACTC
CCGGTGCAGTATGAAGGCGG
```

Figure 21. Input spacers.

The spacer sequence is entered as plain text. Multiple spacers are supported, one per line without additional symbols. If the user first uses the CBEI design, the spacer needed to be predicted can be imported directly from the previous step.



	☑	I...	Strand	Spacer	Pattern	Location
+..	☑	1	Plus	CGTTTTACAACGTCGTGACT	AC	1.2%
+..	☑	2	Minus	TATTACGCCAGCTGGCGAAA	GC	3.61%
+..	☑	3	Minus	ATTACGCCAGCTGGCGAAAG	GC,CC	3.61%
+..	☑	4	Minus	TTACGCCAGCTGGCGAAAGG	GC,CC	3.61%
+..	☑	5	Plus	CAACAGTTGCGCAGCCTGAA	AC	4.91%
+..	☑	6	Minus	CAAAGCGCCATTCGCCATTC	GC	5.76%
+..	☑	7	Plus	CAGACGCGAATTATTTTTGA	GC	13.69%
+..	☑	8	Plus	TTTATGGCAGGGTGAAACGC	GC	25.59%
+..	☑	9	Plus	GGATGAGCAGACGATGGTGC	GC	36.13%
+..	☑	10	Minus	GTGTACCACAGCGGATGGTT	AC,CC	38.93%
+..	☑	11	Minus	CGGTAGCCAGCGCGGATCAT	GC,CC	42.24%
+..	☑	12	Minus	AGCGACCAGATGATCACACT	AC,CC	44.59%
+..	☑	13	Minus	GCGACCAGATGATCACACTC	AC,CC	44.59%
+..	☑	14	Plus	CCGGTGCAGTATGAAGGCGG	GC	47.35%

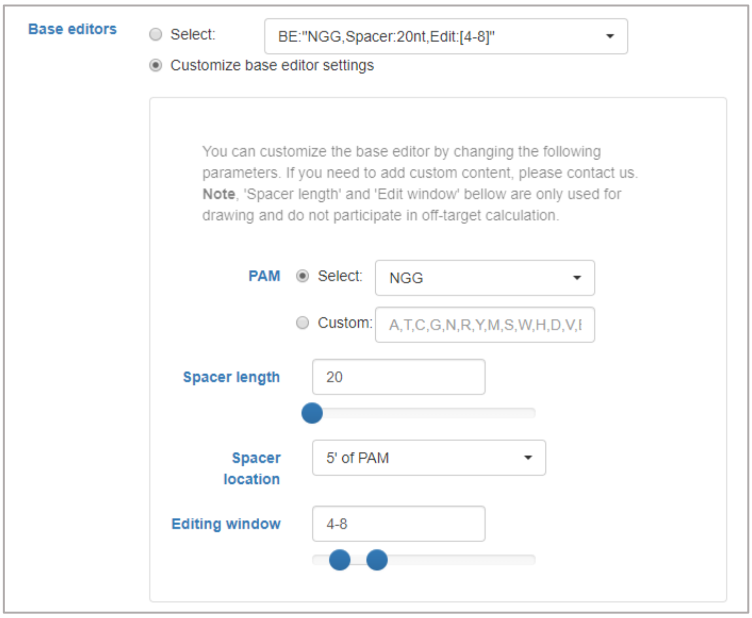
Showing 1 to 14 of 14 rows 25 rows per page

Figure 22. Spacers are selected from CBEI design for off-target prediction.

Tables are front-page pages, and you can change the number of spacers displayed per page by adjusting the paging option. Users can make predictions by selecting all spacers by the click the header or by clicking on a particular spacer row.

3.1.2 Custom base editor

This step is the same setup as in the CBEI design.



Base editors

○ Select: BE:"NGG,Spacer:20nt,Edit:[4-8]"

● Customize base editor settings

You can customize the base editor by changing the following parameters. If you need to add custom content, please contact us. **Note**, 'Spacer length' and 'Edit window' bellow are only used for drawing and do not participate in off-target calculation.

PAM ● Select: NGG

○ Custom: A,T,C,G,N,R,Y,M,S,W,H,D,V,I

Spacer length: 20

Spacer location: 5' of PAM

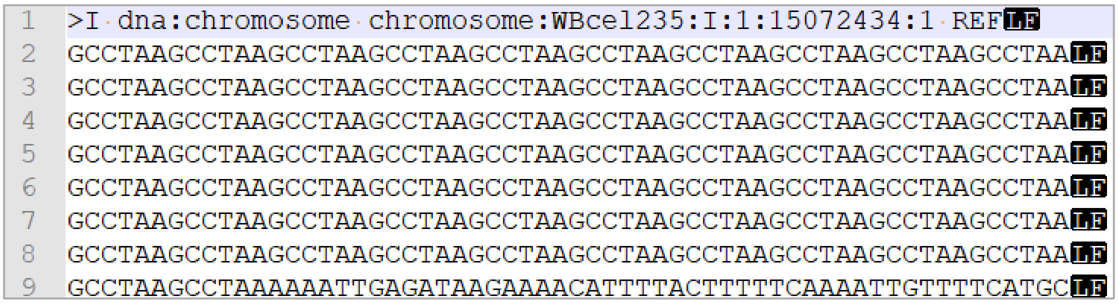
Editing window: 4-8

Figure 23. Custom base editor.

Users can select our built-in base editors or type in the appropriate arguments to get the customized base editor.

3.1.3 Selection of genome files

We support genome files in FASTA format.



```
>I dna:chromosome chromosome:WBcel235:I:1:15072434:1 REF
GCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAA
GCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAA
GCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAA
GCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAA
GCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAA
GCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAA
GCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAA
GCCTAAGCCTAAAAAATTGAGATAAGAAAACATTTTACTTTTTCAAAATTGTTTTCATGC
```

Figure 24. Genome in Fasta format.

In bioinformatics, the FASTA format is a text-based format used to represent nucleotide or amino acid sequences, with the sequence name and annotation preceded by a “>” symbol.

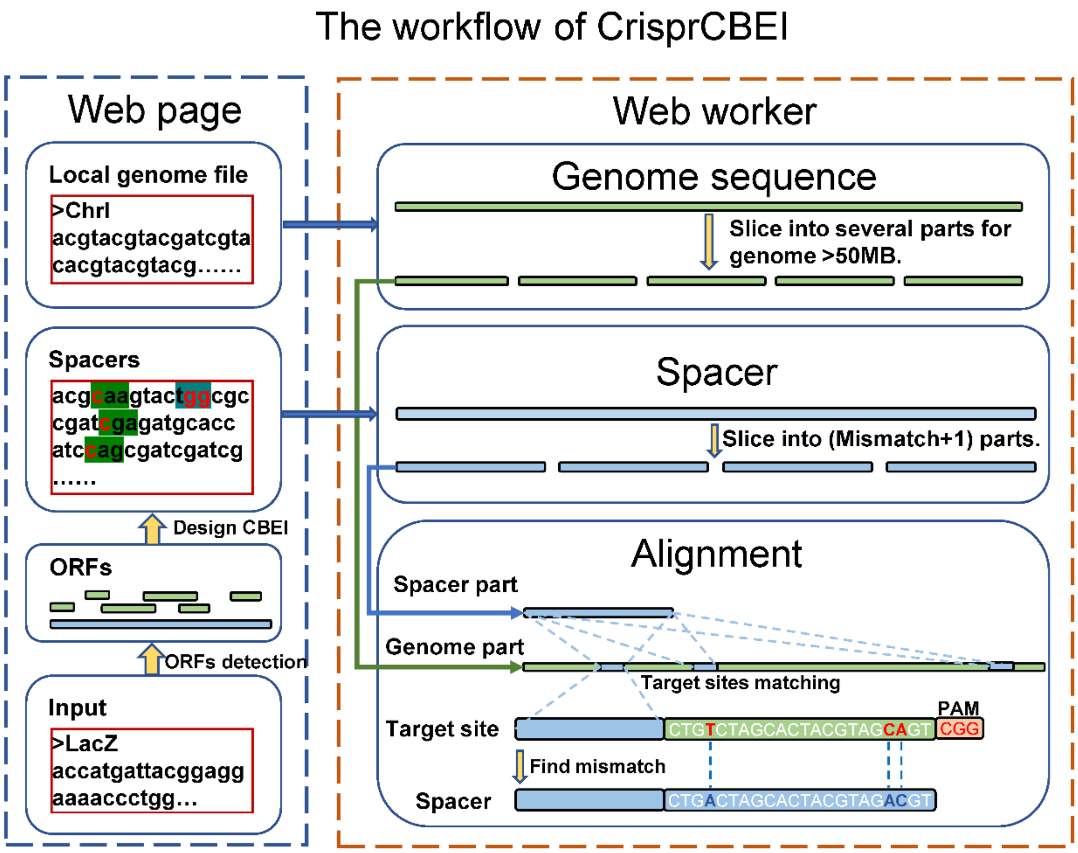
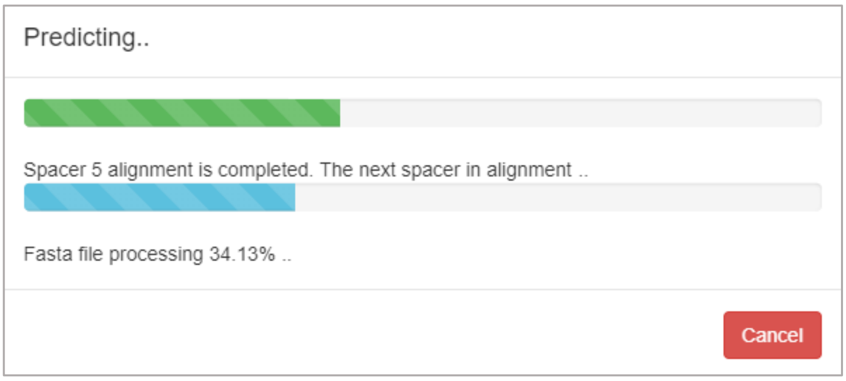


Figure 25. Schematic diagram of the off-target prediction process (right part).

Users can select any genome file in FASTA format that needs for off-target prediction. Our algorithm does not limit the size of the genome file because the genome file is first sliced into 50 MB chunks for calculation. Of course, the upstream and downstream sequence of the cutting site has been considered in the algorithm.


Predicting..

Spacer 5 alignment is completed. The next spacer in alignment ..

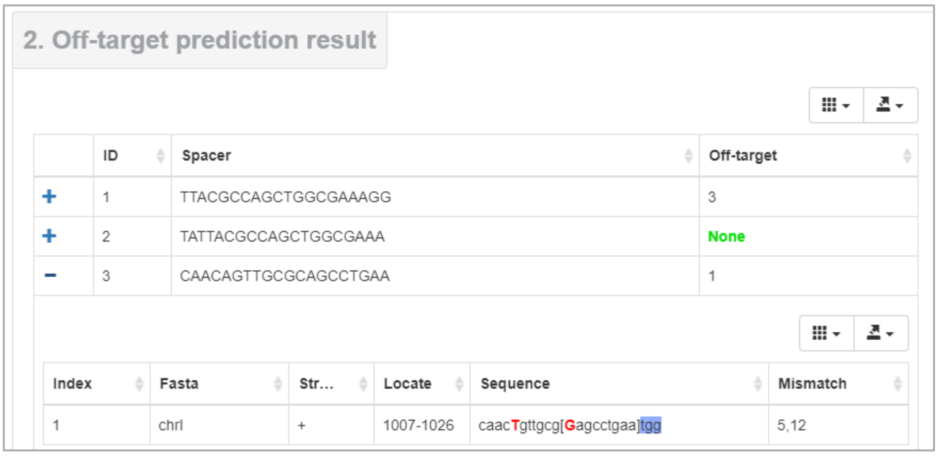
Fasta file processing 34.13% ..

Figure 26. Schematic diagram of the calculation process.

The off-target calculation takes a while, depending on the size of the genome. The corresponding progress bar we designed indicates the progress of the calculation. The process starts by slicing the genome into 50 MB blocks. Each block calculates the off-target sites of each spacer. The two progress bars show genome processing and spacer processing, respectively. If both progress bars reach 100% at the same time, the calculation is complete.

3.2 Results of off-target prediction

The off-target sites with color-labeled mismatch sites are shown in the table. We labeled the 9-nt sequence near the PAM sequence.



2. Off-target prediction result

	ID	Spacer	Off-target
+	1	TTACGCCAGCTGGCGAAAGG	3
+	2	TATTACGCCAGCTGGCGAAA	None
−	3	CAACAGTTGCGCAGCCTGAA	1

Index	Fasta	Str...	Locate	Sequence	Mismatch
1	chrI	+	1007-1026	caacTgttgcg[Gagcctgaa]tgg	5,12

Figure 27. Schematic diagram of the calculation process.

Primary and child tables also support data export.

3.3 The time cost of off-target prediction

Crispr-CBEI off-target prediction function is a front-end prediction tool that supports local Fasta format files for off-target prediction without uploading to the server. Through algorithm optimization, Crispr-CBEI does not limit the size of Fasta files, and the spacer alignment speed is fast. The whole off-target calculation process is asynchronous and does not occupy much memory (about 200 MB), so it would not affect other applications of the computer.

Table 2. The time cost for 10 spacers off-target prediction.

Species	*Size (MB)*	Mismatch			
		⩽3	⩽2	⩽1	⩽0
E.coli CDS	4.66	0m1s±0.01s	0m1s±0.02s	0m0s±0.01s	0m0s±0.02s

E.coli DNA	4.50	0m1s±0.01s	0m0s±0.02s	0m0s±0.02s	0m0s±0.01s
S. cerevisiae CDS	11.10	0m1s±0.04s	0m1s±0.03s	0m1s±0.03s	0m0s±0.01s
S. cerevisiae DNA	11.79	0m1s±0.04s	0m1s±0.03s	0m1s±0.03s	0m0s±0.01s
C.elegans CDS	52.00	0m6s±0.20s	0m4s±0.13s	0m3s±0.19s	0m2s±0.01s
D.rerio CDS	90.86	0m11s±0.23s	0m7s±0.04s	0m5s±0.21s	0m3s±0.01s
C.elegans DNA	97.24	0m12s±0.24s	0m8s±0.24s	0m6s±0.26s	0m4s±0.02s
M.musculus CDS	98.64	0m12s±0.41s	0m8s±0.25s	0m6s±0.25s	0m3s±0.01s
A.thaliana CDS	115.57	0m13s±0.32s	0m9s±0.28s	0m6s±0.25s	0m4s±0.01s
H.sapiens CDS	146.49	0m18s±0.01s	0m12s±0.30s	0m8s±0.36s	0m5s±0.03s
D.rerio DNA	1304.19	2m35s±3.57s	1m46s±3.12s	1m13s±0.28s	0m47s±0.25s
M.musculus DNA	2642.60	5m7s±7.03s	3m33s±6.42s	2m32s±6.24s	1m35s±0.05s
H.sapiens DNA	2994.31	5m52s±7.89s	4m2s±7.02s	2m52s±7.04s	1m48s±0.20s

The addition of the front-end off-target prediction function has become a highlight of the CrisprCBEI, enabling off-target prediction to be carried out in web pages. Nonetheless, due to the limitation of multi-threading in the browser, the front-end off-target prediction is less efficient than the command line multi-threaded version. But after optimization of the algorithm, the time cost of front-end off-target prediction can be controlled within an acceptable range. We performed off-target prediction and calculated time cost for seven species genomes and CDSs (Table 1). The efficiency of the calculation should affect by mismatch value, genome size and CPU power. The mismatch is most efficient if it is set to 0, which is only 108 seconds for 10 spacers in the largest human genome (10.80s per spacer). By default, the mismatch value is set to less than or equal to three and ten spacers took 11.91s per 100MB of data (1.19s per spacer). Therefore, for small genomes such as yeast, the front-end off-target prediction could be calculated in seconds, while for the human genome (2994.31MB), ten spacers took 5m52s (36.36s per spacer). The results of the front-end off-target prediction are exactly the same as those calculated by the existing command line version of off-target prediction software.

4. Acknowledgements

4.1 Institutions and organizations.

This work was supported by the National Natural Science Foundation of China (grants number 3177080732, to S. Tao). This work was also supported by the National Natural Science Foundation of China (21922705, 91753127, and 31700123), the Shanghai Committee of Science and Technology, China (19QA1406000 and 17ZR1449200) to Q. Ji, and the China Postdoctoral Science Foundation (2019M651627) to Z. Wu.

The authors would like to thank the Network & Education Technology Center of NWAFU for their support for the server and the networks.

4.2 Open source projects used

Thanks to the open-source projects, we can save a lot of time to implement the CrisprCBEI online software together. The list of open-source software and links were shown below, if we have any missing, please contact us.

Table 3. Open source projects used

Project name	Links	Usage
Flask	http://flask.pocoo.org/	Website back-end
Nginx	http://nginx.org	
Python 3.7	https://www.python.org/	
uWSGI	https://github.com/unbit/uwsgi	
SQLalchemy	https://www.sqlalchemy.org/	
Bootstrap3	https://getbootstrap.com/	Page layout
Jquery	https://jquery.com/	Easy use for JS

Echarts	https://echarts.baidu.com/	Chart show
Bootstrap-table	https://github.com/wenzhixin/bootstrap-table	Table show
Bootstrap-select	https://github.com/snapappointments/bootstrap-select	Select box
Table export	https://github.com/kayalshri/tableExport.jquery.plugin	Export table
Back2top	https://github.com/zxlie/back2top	Back to top button

4.3 Peoples

We would like to thank Shuo Gao, Xuanyan Li, and Jingjing Song of MTPT technology group for their advice on this project in terms of server operation and webpage technology. We would like to thank the members of Prof. Quanjiang Ji lab for their testing and suggestions on the website. The authors want to thank Prof. Tao and Prof. Ji for their revision and suggestions on the paper. Finally, we would like to thank the vast number of users of the CrisprCBEI, because your concerns and suggestions have made it possible to continue to improve the CrisprCBEI.

5. Contact us

Thank you for choosing and using CrisprCBEI. If you encounter problems in use, please contact us.

5.1 Shiheng Tao Lab

Address：College of Life Sciences and State Key Laboratory of Crop Stress Biology in Arid Areas, Bioinformatics Center, Northwest A&F University, Yangling, Shaanxi, China, 712100

Contact: Tel: (+86-029) 87091060

Mail: shihengt@nwsuaf.edu.cn

5.2 Quanjiang Ji Lab

Address: School of Physical Science and Technology, ShanghaiTech University, Shanghai, China, 201210

Mail: quanjiangji@shanghaitech.edu.cn

5.3 Bug report

Dr. Haopeng Yu: atlasbioin4@gmail.com

Dr. Zhaowei Wu: wuzw1@shanghaitech.edu.cn