Designing genome editing experiments with EditABLE

Demetrios S. Maxim[1,2,3,4], Juliet Sostena[3,5], Najani Shanee Johnson[3,6], David Wei Wu[7,8], Vivek Charu[9], Jennefer N. Carter[10], Shuchi Anand[1], George M. Church[3,11,12], Vivek Bhalla[1,3,13,14,*]

[1] Division of Nephrology, Department of Medicine, Stanford University School of Medicine, Stanford, CA

[2] Department of Biology, Stanford University, Stanford School of Humanities and Sciences, Stanford, CA

[3] Nephrogen Inc., New York, NY

[4]@DemetriMxm (Twitter X)

[5] Department of Bioengineering, Stanford University, Stanford School of Engineering, Stanford, CA

[6] Department of Electrical and Computer Engineering, University of Albany, Albany, NY

[7] Medical Scientist Training Program, Perelman School of Medicine at the University of Pennsylvania, Philadelphia, PA

[8] Genomics and Computational Biology Graduate Program, University of Pennsylvania, Pennsylvania, Philadelphia, PA

[9] Department of Pathology, Stanford University School of Medicine, Stanford, CA

[10]Division of Cardiovascular Medicine, Department of Medicine, Stanford University School of Medicine, Stanford, California

[11] Department of Genetics, Harvard Medical School, Boston, MA

[12] Wyss Institute for Biologically Inspired Engineering at Harvard University, Boston, MA

[13]@BhallaResearch (Twitter X)

[14]@bhallaresearch.bsky.social (Bluesky)

* To whom correspondence should be addressed. Tel: +1 (650) 721-2471; Fax: +1 (650) 721-2471; Email: vbhalla@stanford.edu

ADDITIONAL FILE 1

1. SUPPLEMENTARY FIGURES

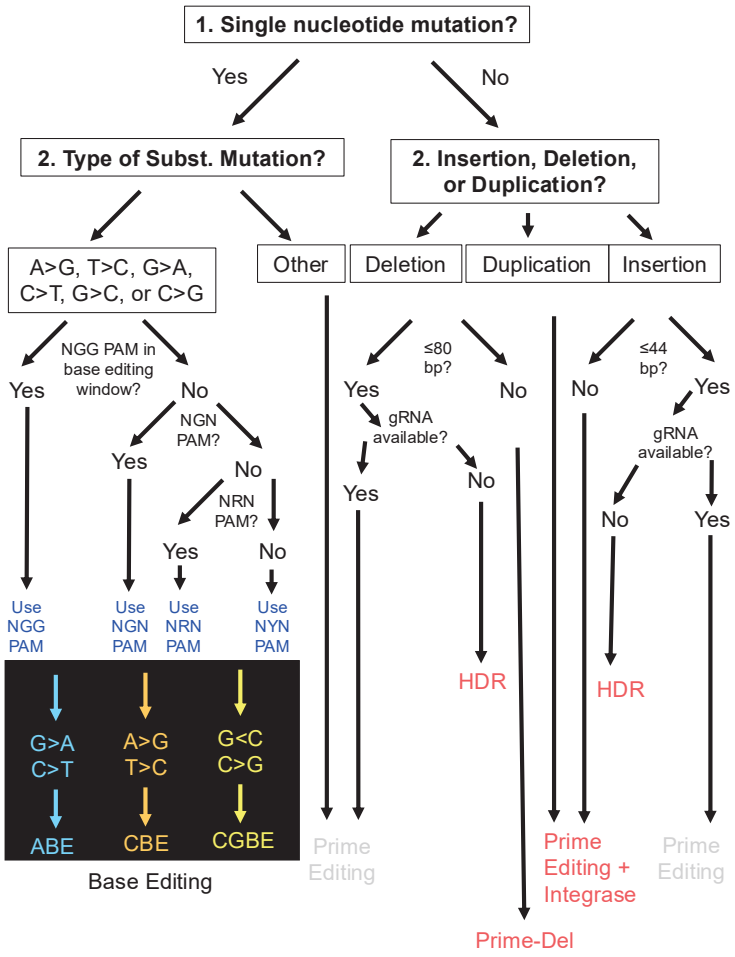


Fig. S1: Schematic diagram of EditABLE algorithm. Mutations are classified as amenable to base editing, prime editing (grey), or other (red). For example, A>G stands for an A>G substitution mutation, which requires a G>A correction via a CBE. For the ADPKD mutation analysis, available PAM sites for ABE and CBE base editors were defined as the presence of NGN nucleotide sequence that positioned the substitution mutation 12-17 bp upstream of the PAM in the 5' direction, which was identified previously as the most efficient base editing window (4) (5). Available PAM sites for transversion base editors (CGBE) were defined as when, starting from the 5' direction, the C nucleotide to be edited is in position 6 and an NGG, NGA, or NGN PAM is in positions 21-23 (11). BP = base pair, PAM = protospacer-adjacent motif, gRNA = guide RNA, ABE = adenine base editor, CBE = cytosine base editor, CGBE = transversion base editor, HDR = homology directed repair.

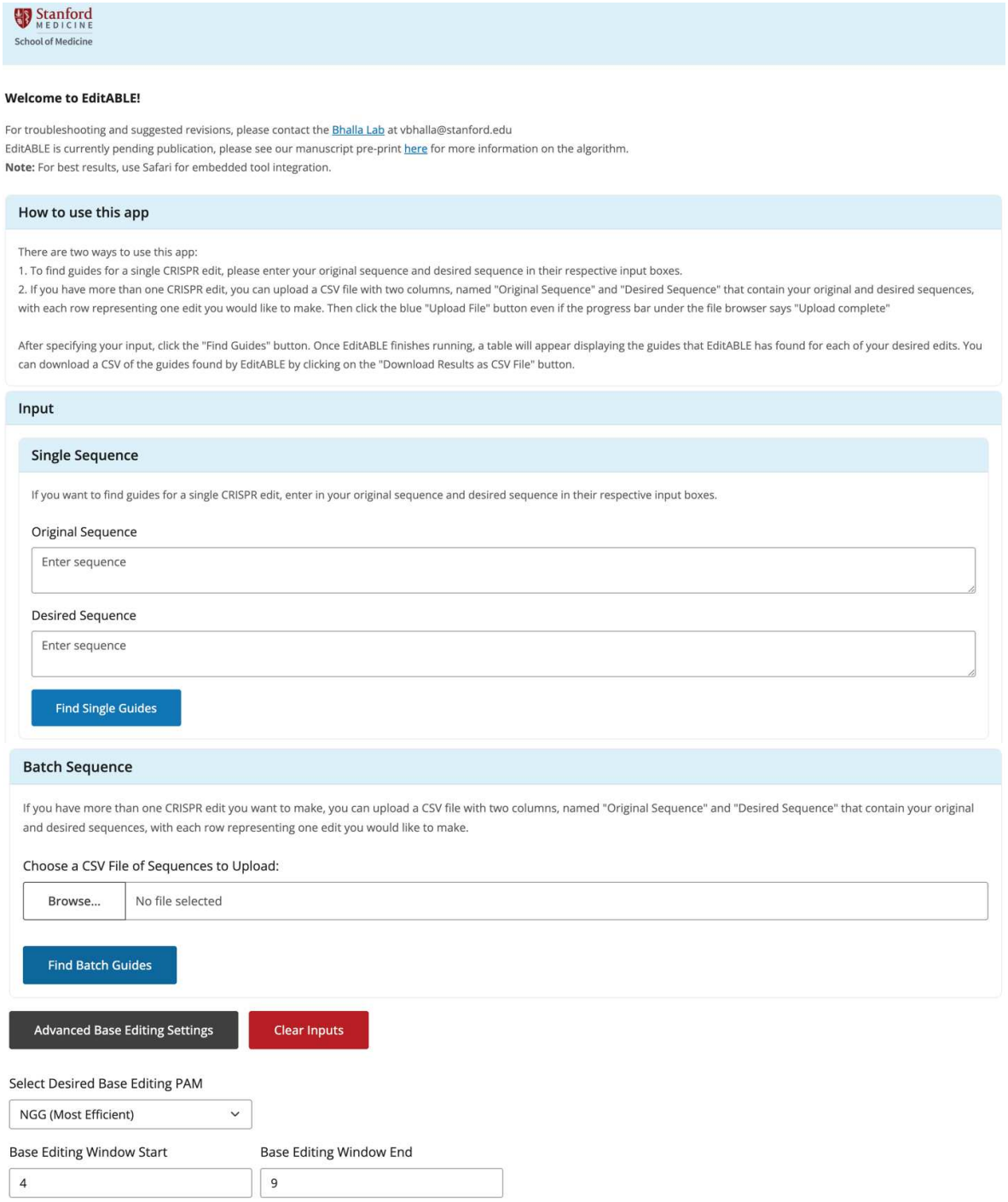


Fig. S2: Screenshots of the EditABLE web application. EditABLE accepts both single sequence data inputted directly into the web application or batch sequence data uploaded as a CSV. Batch sequences should be uploaded into two columns: original sequence and desired sequence. EditABLE also allows advanced users to modify specific base editing settings such as the editing window and PAM. The default PAM setting is NGG and the default base editing window is bases 4 to 9. The base editing window start and end sites are inclusive. Web page accessed March 13, 2026.

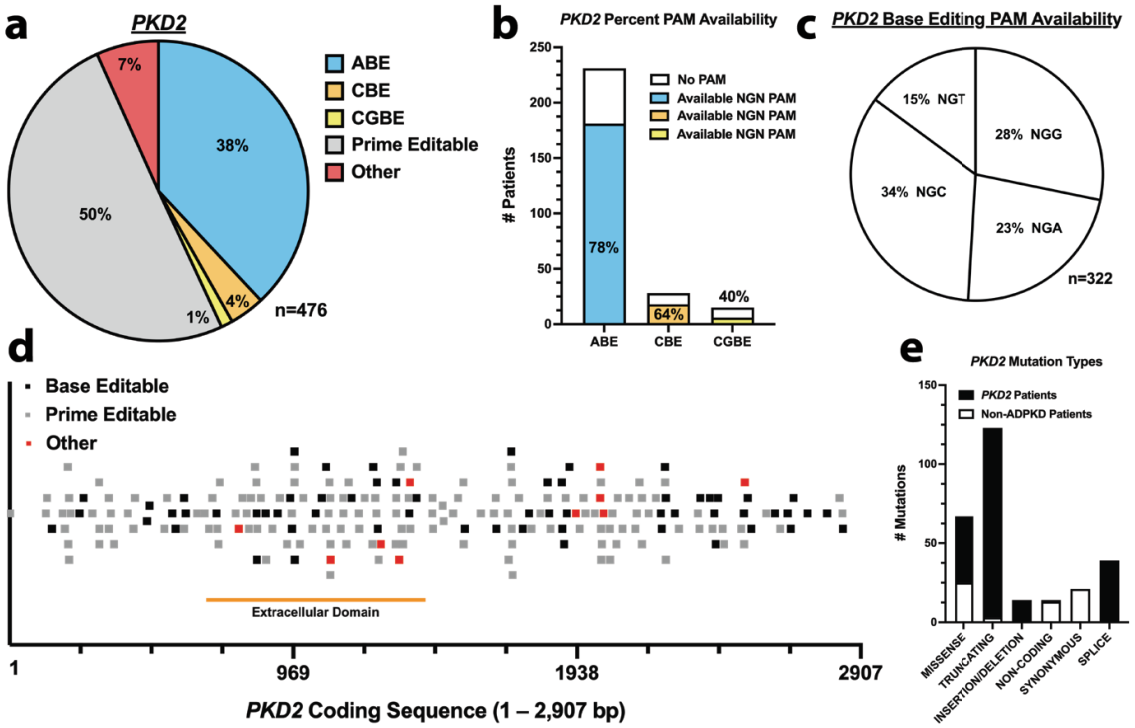


Fig. S3: 93% of *PKD2* patients carry mutations correctable via base or prime editing. All *PKD2* variants from patients without ADPKD in the Mayo Clinic database were excluded from the analysis. **(a)** Pie chart showing percentage of patients with *PKD2* amenable to each base or prime editing technology. Patients were classified as amenable to base editing (ABE, CBE, or CGBE) if their mutation possessed an NGN PAM sequence that positioned the substitution mutation 12-17 bp upstream of the PAM in the 5' direction. **(b)** Bar graph showing percent of patients who possess the appropriate substitution mutations for base editing but do not have NGN PAM sequences available in the optimal editing window. **(c)** Pie chart showing which third nucleotides are most common in NGN PAM sequences. The data from part **(a)** are graphed as dotplots against the **(d)** *PKD2* coding region. Each dot represents an individual mutation from the Mayo Clinic ADPKD Mutation Database. All mutations from the Mayo database are shown except large deletion or duplication mutations spanning multiple exons. **(e)** Bar graph showing a breakdown of mutation types in the ADPKD mutation database with the percent of pathogenic mutations vs. non-ADPKD mutations for each mutation type. ABE, adenine base editor; CBE, cytosine base editor; CGBE, transversion base editor; PAM, protospacer-adjacent-motif.

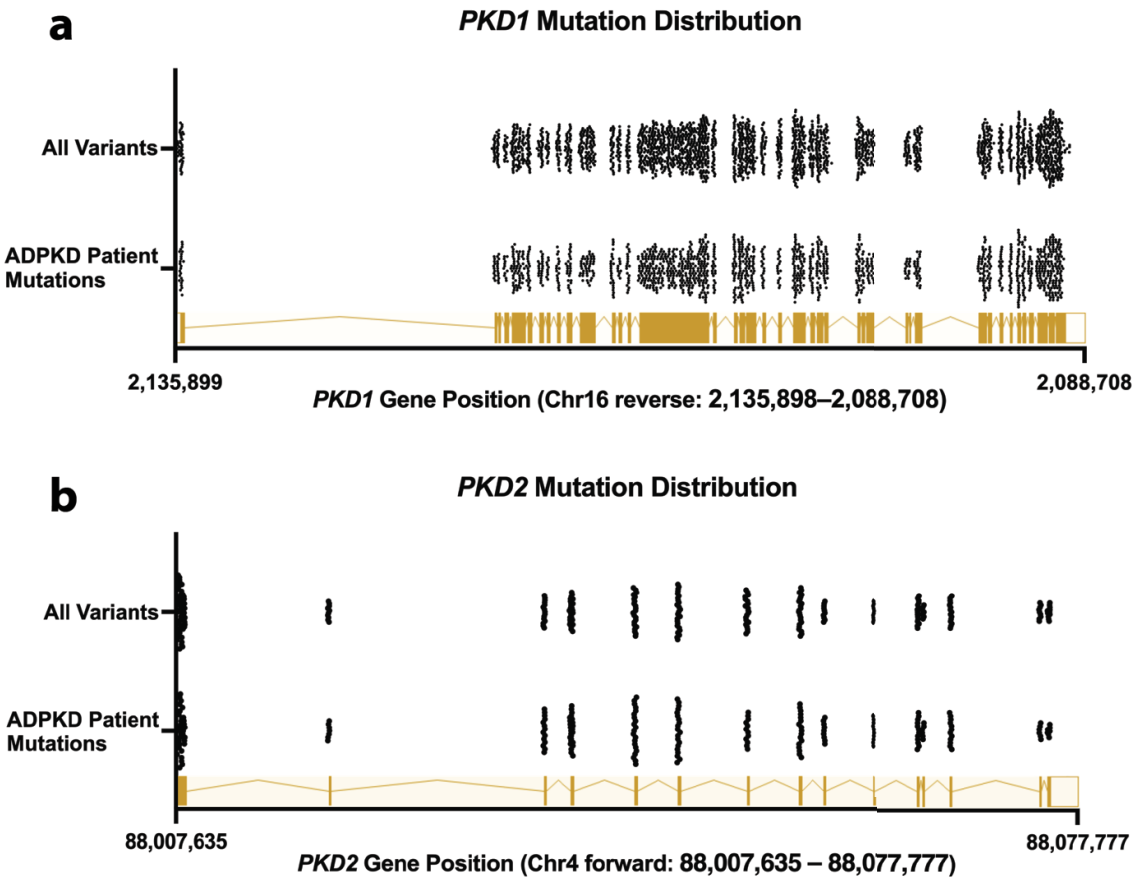


Fig. S4: PKD1 (a) and PKD2 (b) mutations are scattered throughout the genomic region and there are no clear hotspot mutation regions. The full *PKD1* (Chr16: 2,135,898 – 2,088,708) and *PKD2* (Chr4: 88,007,635 – 88,077,777) genomic sequences are graphed in sense orientation, superimposed with all mutations from the Mayo Clinic ADPKD Database. Mutations from confirmed ADPKD patients are shown in a separate dot plot below all mutation data for *PKD1* and *PKD2*.

Fig. S5: Screenshots of the direct PrimeDesign integration within the EditABLE web application. The EditABLE website (available at editable-app.stanford.edu) directly integrates the PrimeDesign algorithm into the webpage to allow advanced users to further customize the default prime editing guide RNAs that EditABLE recommends. This PrimeDesign integration includes visualization of the prime editing guide RNA structure and offers various options to the user for modifying the prime editing guide RNA such as the PBS and RTT lengths. Accessed on March 13, 2026.

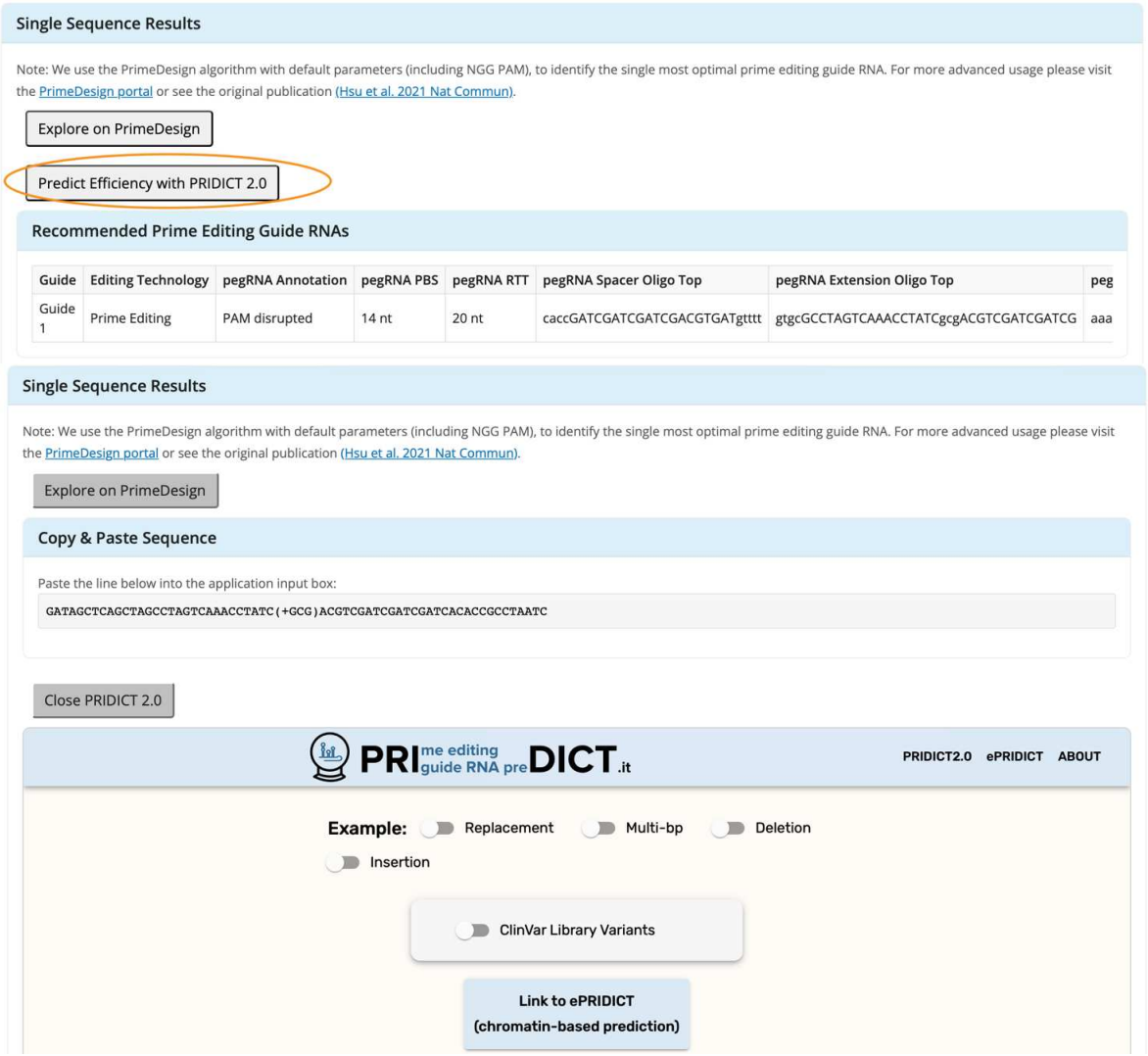


Fig. S6. Screenshots of the direct PRIDICT2.0/ePRIDICT integration within the EditABLE web application. The EditABLE website (available at editable-app.stanford.edu) directly integrates the PRIDICT2.0 and ePRIDICT algorithms into the webpage to calculate on-target and off-target efficiency scores for prime editing guide RNAs; thus, allowing advanced users to further customize the prime editing guide RNA designs initially recommended by the EditABLE algorithm. When the user clicks the button circled in orange in this figure and entitled “Predict Efficiency with PRIDICT2.0,” the PRIDICT2.0 interface appears along with a sequence that can be copied and pasted directly into the PRIDICT2.0 window to begin using the algorithm. Accessed on March 13, 2026.

2. SUPPLEMENTARY TABLES

Feature	Edit ABLE	PnB Designer	Benchling	Prime Design	CHOPCHOP	Cas-OFFinder	CRISPOR	CRISPick	GenCRISR sgRNA Design Tool	CRISPR GPT	PRIDICT2.0 & ePRIDICT	bEditor
Customize base editing window	✓	✓	X	X	X	X	X	X	X	✓	X	✓
Customize PAM selection	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ranks guide RNAs based on predicted on-target and off-target efficiency scores	✓	X	✓	X	✓	✓	✓	✓	✓	✓	✓	✓
Visualize prime editing guide RNAs	✓	X	X	✓	X	X	X	X	X	✓	X	X
Visualize base editing guide RNAs (editing window, PAM, bystander edits)	✓	X	X	X	X	X	X	X	X	✓	X	✓
Identify and visualize base editing bystander edits	✓	X	X	X	X	X	X	X	X	X	X	✓
Links to Addgene pages for plasmids for recommended editors and guide RNAs	✓	X	X	X	X	X	X	X	X	X	X	X
Step-by-step cloning protocol to experimentally validate guide RNAs for base and prime editing.	✓	X	X	X	X	X	X	X	X	✓	X	X
Guides users on which editing technology to use (prime, base, integrases, etc.)	✓	X	X	X	X	X	X	X	X	X	X	X
Identify reagents for large deletions using PRIMEdel	✓	X	X	X	X	X	X	X	X	X	X	X
Identify editing reagents for large insertions with Bxbl integrases	✓	X	X	X	X	X	X	X	X	X	X	X

Table S1. Full list of EditABLE features compared to existing guide RNA design tools. Although several web-based tools exist for each feature, only the primary references are shown. This table describes 12 relevant computational CRISPR tools covering base editing, prime editing, off-target prediction, PAM flexibility, editing outcome analysis, and guide cloning resources. Most existing tools focus on one or a few dedicated functions, for example, CHOPCHOP emphasizes flexible gRNA design and PAM selection, Cas-OFFinder provides fast off-target searches, CRISPOR offers combined design and scoring, CRISPResso focuses on outcome analysis, PegFinder, PRIDICT2.0, and ePRIDICT target

prime editing, and PrimeDEL focuses on large deletions. In contrast, EditABLE integrates tasks and acts as a gateway interface, increasing the accessibility and streamlining processes. This directly addresses the needs outlined by several recent reviews, including Jasieniecka & Domingues (2025) and Saraswat & Ranjan (2025), which call for “integrated platforms that combine functionalities, reducing reliance on fragmented workflows” (25–28,36,37).

Table S2 (XLS FILE). List of *PKD1* mutations for clinical base editing proof-of-concept. Mutations were identified as candidates if they resulted in truncation of *PKD1* protein, had been identified in at least 5 unique pedigrees, and were amenable to correction via base editing with an NGN PAM site. The mutations are listed in order of appearance in the *PKD1* coding region starting with exon 1. *PKD1*^{Q2556X} mutation used for *in vitro* validation in Figure 2 is in **bold**. EX = exon; Refs. = number of independent sources that cite the mutation.

Table S3 (XLS FILE). List of *PKD1* mutations amenable to prime editing. Mutations were identified as candidates if they resulted in truncation of *PKD1* protein, found in at least 5 unique pedigrees, and amenable to correction via prime editing with an NGG PAM site. Guide RNA sequences were excluded for simplicity since prime editing gRNA design is highly complex; however, the guides can be easily obtained through the EditABLE website. The “Edit Type” column is used to indicate the type of prime edit needed to correct the mutation. The mutations are listed in order of appearance in *PKD1* starting with exon 1. EX = exon; Refs. = number of independent sources that cite the mutation.

Table S4 (XLS FILE). Example batch sequence input file for the EditABLE. This batch file contains all the example mutations described in Supplementary Note 2 and highlights the scalability of the EditABLE web tool.

Table S5 (XLS FILE). Expected results from the EditABLE web application for the batch sequence input file in Supplementary Table 4. These results were directly downloaded from EditABLE in CSV format.

Table S6 (XLS FILE). Full list of *PKD1* germline mutations used for the analyses in this study. This list was taken directly from the Mayo Clinic PKD Mutation Database website and converted into CSV for further processing. Records are current as of March 13, 2026.

Table S7 (XLS FILE). Full list of *PKD2* germline mutations used for the analyses in this study. This list was taken directly from the Mayo Clinic PKD Mutation Database website and converted into CSV for further processing. Records are current as of March 13, 2026.

3. SUPPLEMENTARY NOTES

Note S1. Python source code for analyzing the ADPKD mutation database using the EditABLE algorithm. Requires installation of pandas, numpy, regex, and Bio packages.

Note S2. Example inputs for the EditABLE website. Each example includes the reference and edited sequences entered as well as any advanced user settings. If no advanced base editing settings are shown, the default EditABLE parameters were used (NGG = PAM, 4 = start base editing window, 9 = end base editing window). The expected output for each example is shown as a screenshot below the inputs.

Note S3. Full protocols for long-range and short-range PCR on the *PKD1* genomic region containing the candidate *PKD1*^{Q2556X} mutation.

NOTE S1

Python code for the EditABLE analysis of the Mayo Clinic ADPKD Mutation Database:

```
import pandas as pd
import numpy as np
import regex as re
from Bio.Seq import Seq
from Bio.Seq import MutableSeq

"""
This code was written to parse the Mayo Clinic ADPKD Database and
count the number of PKD1 and PKD2 Mutations that are amenable to
base editing.

IMPORTANT: Before running this script, all deletion mutations (in
the cDNA column) must be formatted as firstBase_lastBase with an
UNDERSCORE and not a dash. Deletion mutations with a dash will not
be counted properly.

"""

# Initializes the reference genome
gene_name = "pkd1"
f = open(gene_name + "_ref_seq.txt", "r")
ref_seq = Seq(f.read()).upper()
f.close()

# Created dictionary with exon start sites in PKD1 or PKD2 genes
genom_exon_ss = pd.read_csv(gene_name +
'_genomic_exon_start_sites.csv', header=None,
index_col=0).squeeze(1).to_dict()
cds_exon_ss = pd.read_csv(gene_name + '_cds_exon_start_sites.csv',
header=None, index_col=0).squeeze(1).to_dict()
exon_sizes = pd.read_csv(gene_name + '_exon_sizes.csv', header=None,
index_col=0).squeeze(1).to_dict()
PAMs = re.compile("[A|T|G|C]G")
reverse_PAMs = re.compile("C[A|T|G|C]")
edit_start = 4
edit_end = 9
gRNA_size = 20
pam_length = 2

############ DO NOT CHANGE ANY PARAMETERS BELOW THIS LINE
############

# Helper function to reformat exons and introns as numbers to be
used for finding gene positions
def exon_numbering(region, mutation_type):
     if "LARGE" in mutation_type:
          return "LARGE"
     elif ("UTR" in region) and ("5" in region):
          return "5UTR"
     elif ("UTR" in region) and ("3" in region):
          return "3UTR"
```

```
      elif ("-" in region) or ("_" in region):
            return "COMPLEX"
      elif "EX" in region:
            return int(re.findall(r'\d+', region)[0])
      elif "IVS" in region:
            return re.findall(r'\d+', region)[0] + "I"
      else:
            return "ERROR"

# Helper function to add nt position within full gene sequence to
initial dataset
def full_gene_position(cDNA, exon):
      if "LARGE" in str(exon):
            return "LARGE"
      elif str(exon).isdigit():
            return genom_exon_ss[exon] + (int(re.findall('\d+',
cDNA)[0]) - cds_exon_ss[exon])
      elif "I" in str(exon):
            intron_num = int(re.findall(r'\d+', exon)[0])
            if "+" in cDNA:
                  return genom_exon_ss[intron_num] +
exon_sizes[intron_num] + int(re.findall('\d+',
cDNA.split('+')[1])[0])
            elif "-" in cDNA:
                  return genom_exon_ss[intron_num + 1] + 1 -
int(re.findall('\d+', cDNA.split('-')[1])[0])
            elif "_" in cDNA:
                  return genom_exon_ss[intron_num + 1]
            else:
                  return "FALSE"
      elif "5UTR" in exon:
            coding_region_start = min(genom_exon_ss,
key=genom_exon_ss.get)
            return genom_exon_ss[coding_region_start] -
int(re.findall('\d+', cDNA.split('-')[1])[0])
      elif "3UTR" in exon:
            coding_region_end = max(genom_exon_ss,
key=genom_exon_ss.get)
            return exon_sizes[coding_region_end] +
genom_exon_ss[coding_region_end] + int(re.findall('\d+',
cDNA.split('*')[1])[0])
      elif "COMPLEX" in str(exon):
            return "COMPLEX"
      else:
            return "ERROR"

# Helper function to add coding sequence nt position to initial
dataset
def cds_position(cDNA, exon):
      if "LARGE" in str(exon):
            return "LARGE"
      elif str(exon).isdigit():
            return  int(re.findall('\d+', cDNA)[0])
      elif "I" in str(exon):
            intron_num = int(re.findall(r'\d+', exon)[0])
            if "+" in cDNA:
```

```
                return cds_exon_ss[intron_num] +
exon_sizes[intron_num] + int(re.findall('\d+',
cDNA.split('+')[1])[0])
          elif "-" in cDNA:
                return cds_exon_ss[intron_num + 1] -
int(re.findall('\d+', cDNA.split('-')[1])[0])
          elif "_" in cDNA:
                return cds_exon_ss[intron_num + 1]
          else:
                return "ERROR"
     elif "5UTR" in str(exon):
          return
     elif "3UTR" in str(exon):
          return
     elif "COMPLEX" in str(exon):
          return "COMPLEX"
     else:
          return "ERROR"

# Helper function to add mutations definitions to initial dataset
def mutation_types(cDNA):
     if ("del" in cDNA) & ("ins" in cDNA):
          return "COMPLEX"
     elif "dup" in cDNA:
          return "FALSE"
     elif ("C>T" in cDNA) | ("G>A" in cDNA):
          if "," in cDNA:
                return "COMPLEX"
          else:
                return "ABE"
     elif ("T>C" in cDNA) | ("A>G" in cDNA):
          if "," in cDNA:
                return "COMPLEX"
          else:
                return "CBE"
     elif ("C>G" in cDNA) | ("G>C" in cDNA):
          if "," in cDNA:
                return "COMPLEX"
          else:
                return "TRANS"
     elif "del" in cDNA:
          if ("-" in cDNA) | ("+" in cDNA):
                return "COMPLEX"
          if "_" in cDNA:
                start = cDNA.split('_')[0]
                end = (cDNA.split('_')[1]).split('del')[0]
                if end == "3'UTR":
                      end = 14138
                if (int(end) - int(start)) <= 44:
                      return "PRIME INSERTION"
                else:
                      return "FALSE"
          else:
                if len(cDNA.split('del')[1]) <= 44:
                      return "PRIME INSERTION"
                else:
```

```
                    return "ERROR"
     elif "ins" in cDNA:
          if "_" in cDNA:
               num = cDNA.split('ins')[1]
               if num.isdigit():
                    if int(num) <= 80:
                         return "PRIME DELETION"
                    else:
                         return "FALSE"
               elif len(num) <= 80:
                    return "PRIME DELETION"
               else:
                    return "FALSE"
          else:
               if len(cDNA.split('ins')[1]) <= 80:
                    return "PRIME DELETION"
               else:
                    return "ERROR"
     elif ("T>G" in cDNA) | ("G>T" in cDNA) | ("A>C" in cDNA) |
("C>A" in cDNA) | ("A>T" in cDNA) | ("T>A" in cDNA):
          return "PRIME SUBST EDIT"
     else:
          return "ERROR"

# Helper function to find guide RNAs for ABE and CBE EditABLE
mutations on forward/reverse strands
def find_BE_guide_rnas(direction, mutant_seq, genomic_location):
     gen_index = genomic_location - 1

     if direction == "forward":
          start_pams = gen_index + (gRNA_size - edit_end) + 1
          end_pams = gen_index + (gRNA_size - edit_start) +
pam_length + 1
          possible_pams = mutant_seq[start_pams : end_pams]
          if len(re.findall(PAMs, str(possible_pams))) == 0:
               return "NO PAM"
          else:
               guideRNAs = []
               for match in re.finditer(PAMs, str(possible_pams),
overlapped=True):
                    gRNA = mutant_seq[start_pams + match.start() -
gRNA_size: start_pams + match.start()]
                    guideRNAs.append(str(gRNA))
               return guideRNAs

     elif direction == "reverse":
          start_pams = gen_index - gRNA_size + edit_start -
pam_length
          end_pams = gen_index - gRNA_size + edit_end
          possible_pams = mutant_seq[start_pams : end_pams]
          if len(re.findall(reverse_PAMs, str(possible_pams))) ==
0:
               return "NO PAM"
          else:
               guideRNAs = []
```

```
                for match in re.finditer(reverse_PAMs,
str(possible_pams), overlapped=True):
                    gRNA = mutant_seq[start_pams + match.start() +
pam_length : start_pams + match.start() + pam_length + gRNA_size]
                    gRNA.reverse_complement()
                    guideRNAs.append(str(gRNA))
                return guideRNAs
     else:
          return "ERROR"

# Helper function to find guide RNAs for transversion (C>G and G>C)
mutations on forward/reverse strands
def find_trans_guide_rnas(direction, mutant_seq, genomic_location):
     gen_index = genomic_location - 1

     if direction == "forward":
          pam = mutant_seq[gen_index + 15 : gen_index + 15 +
pam_length]
          if len(re.findall(PAMs, str(pam))) == 0:
               return "NO PAM"
          else:
               return mutant_seq[gen_index + 15 - gRNA_size :
gen_index + 15]
     elif direction == "reverse":
          pam = mutant_seq[gen_index - 15 - 1: gen_index - 15 - 1 +
pam_length]
          if len(re.findall(reverse_PAMs, str(pam))) == 0:
               return "NO PAM"
          else:
               gRNA = mutant_seq[gen_index - 15 - 1 + pam_length:
gen_index - 15 - 1 + pam_length + gRNA_size]
               gRNA.reverse_complement()
               return gRNA
     else:
          return "ERROR"

# Adds the base EditABLE guide RNAs to the data table for export
def add_guide_RNAs(cDNA, edit_type, genomic_location):
     mutant_seq = MutableSeq(ref_seq)
     if "A>G" in cDNA and edit_type == "CBE":
          mutant_seq[genomic_location - 1] = "G"
          return find_BE_guide_rnas("reverse", mutant_seq,
genomic_location)
     elif "T>C" in cDNA and edit_type == "CBE":
          mutant_seq[genomic_location - 1] = "C"
          return find_BE_guide_rnas("forward", mutant_seq,
genomic_location)
     elif "G>A" in cDNA and edit_type == "ABE":
          mutant_seq[genomic_location - 1] = "A"
          return find_BE_guide_rnas("forward", mutant_seq,
genomic_location)
     elif "C>T" in cDNA and edit_type == "ABE":
          mutant_seq[genomic_location - 1] = "T"
          return find_BE_guide_rnas("reverse", mutant_seq,
genomic_location)
     elif "G>C" in cDNA and "TRANS" in edit_type:
```

```
            mutant_seq[genomic_location - 1] = "C"
            return find_trans_guide_rnas("forward", mutant_seq,
genomic_location)
      elif "C>G" in cDNA and "TRANS" in edit_type:
            mutant_seq[genomic_location - 1] = "G"
            return find_trans_guide_rnas("reverse", mutant_seq,
genomic_location)

# Formats raw data from the Mayo Clinic Database, adds mutations
definitions, and finds guide RNAs
def format_dataset(df):
      df.rename(columns = {'Score' : 'Variant Score'}, inplace=True)
      df['# Patients'] = df['#'].apply(lambda x: 0 if '-' in x else
x.split('(')[0])
      df['# Publications'] = df['#'].apply(lambda y: y.split('-')[1]
if '-' in y else y[y.find("(")+1:y.find(")")])
      df.drop(columns=['Percent', '#'], inplace=True, axis=1)

      # Formats data as integers
      df[['# Patients', '# Publications']] = df[['# Patients', '#
Publications']].apply(pd.to_numeric, downcast='integer', axis = 1)
      df['# Publications'] = df['# Publications'].astype('Int64')
      df['# Patients'] = df['# Patients'].astype('Int64')
      df['Codon'] = df['Codon'].astype('Int64')

      # Adds additional fields to dataset to help with guide RNA
finding
      df['Edit Type'] = df.apply(lambda x: mutation_types(x['cDNA
Change']),axis=1)
      df['Exon'] = df.apply(lambda x: exon_numbering(x['Region'],
x['Mutation Type']),axis=1)
      df['Genomic Location'] = df.apply(lambda y:
full_gene_position(y['cDNA Change'], y['Exon']), axis=1)
      df['CDS Location'] = df.apply(lambda z: cds_position(z['cDNA
Change'], z['Exon']), axis=1)
      df['gRNA'] = df.apply(lambda x: add_guide_RNAs(x['cDNA
Change'], x['Edit Type'], x['Genomic Location']),axis=1)

      return df

# Optional helper function that removes all mutations not found in
pedigrees
def remove_null_mutations(df):
      df.drop(df.loc[df['# Patients']== 0].index, inplace=True)
      return(df)

def main():
      # Creates dataset and outputs to CSV
      df =
remove_null_mutations(format_dataset(pd.read_csv(gene_name +
"_germline_mutations.csv")))
      #df = df.loc[df["Edit Type"].str.contains("ABE") | df["Edit
Type"].str.contains("CBE") | df["Edit Type"].str.contains("TRANS")]
      df.to_csv('output.csv')

if __name__ == "__main__":
```

```
main()
```

NOTE S2

All examples in this section use default settings for base editing in the EditABLE web application (NGG PAM and a base editing window of 4 to 9 within a 20 bp guide RNA).

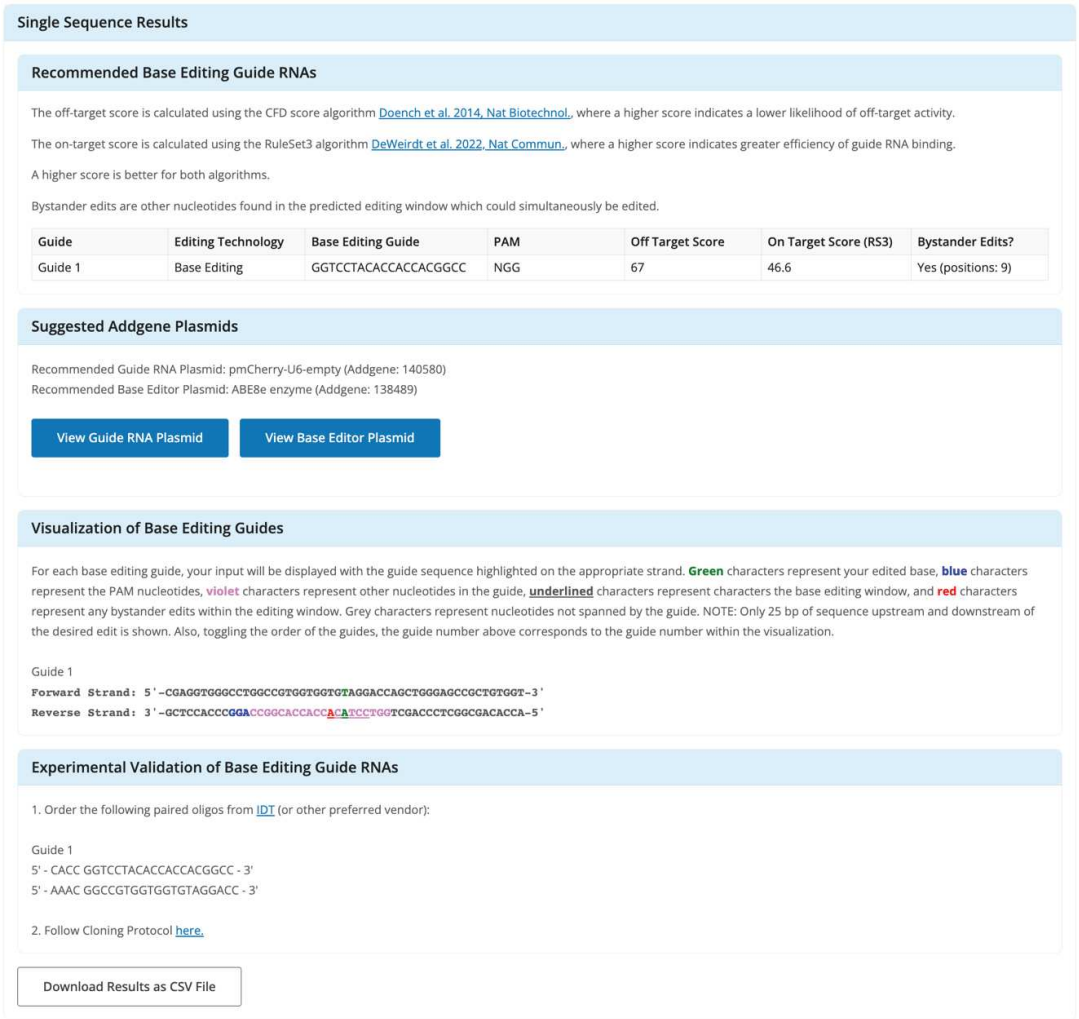
Example 1 – Human PKD1 Base Editing with NGG PAM site

Sequence change: Correction of *PKD1*^{Q2556X} nonsense substitution mutation (32). Revert T (mutant) to C (wildtype).

Original Sequence: GTTTCAGGCCACACTTCGAGGTGGGCCTGGCCGTGGTGGTG**T**A GGACCAGCTGGGAG CCGCTGTGGTCGCCCTCAACAGGTGAGCCAGGCCGTGGGA GGGCGCCC

Desired Sequence: GTTTCAGGCCACACTTCGAGGTGGGCCTGGCCGTGGTGGTG**C**A GGACCAGCTGGGAG CCGCTGTGGTCGCCCTCAACAGGTGAGCCAGGCCGTGGGA GGGCGCCC

Expected Output:



Single Sequence Results

Recommended Base Editing Guide RNAs

The off-target score is calculated using the CFD score algorithm Doench et al. 2014, Nat Biotechnol., where a higher score indicates a lower likelihood of off-target activity.

The on-target score is calculated using the RuleSet3 algorithm DeWeirdt et al. 2022, Nat Commun., where a higher score indicates greater efficiency of guide RNA binding.

A higher score is better for both algorithms.

Bystander edits are other nucleotides found in the predicted editing window which could simultaneously be edited.

Guide	Editing Technology	Base Editing Guide	PAM	Off Target Score	On Target Score (RS3)	Bystander Edits?
Guide 1	Base Editing	GGTCCTACACCACCACGGCC	NGG	67	46.6	Yes (positions: 9)

Suggested Addgene Plasmids

Recommended Guide RNA Plasmid: pmCherry-U6-empty (Addgene: 140580)
Recommended Base Editor Plasmid: ABE8e enzyme (Addgene: 138489)

View Guide RNA Plasmid | View Base Editor Plasmid

Visualization of Base Editing Guides

For each base editing guide, your input will be displayed with the guide sequence highlighted on the appropriate strand. **Green** characters represent your edited base, **blue** characters represent the PAM nucleotides, violet characters represent other nucleotides in the guide, **underlined** characters represent characters the base editing window, and **red** characters represent any bystander edits within the editing window. Grey characters represent nucleotides not spanned by the guide. NOTE: Only 25 bp of sequence upstream and downstream of the desired edit is shown. Also, toggling the order of the guides, the guide number above corresponds to the guide number within the visualization.

Guide 1
Forward Strand: 5'-CGAGGTGGGCCTGGCCGTGGTGGTGTAGGACCAGCTGGGAGCCGCTGTGGT-3'
Reverse Strand: 3'-GCTCCACCCGGACCGGCACCACCACATCCTGGTCGACCCTCGGCGACACCA-5'

Experimental Validation of Base Editing Guide RNAs

1. Order the following paired oligos from IDT (or other preferred vendor):

Guide 1
5' - CACC GGTCCTACACCACCACGGCC - 3'
5' - AAAC GGCCGTGGTGGTGTAGGACC - 3'

2. Follow Cloning Protocol here.

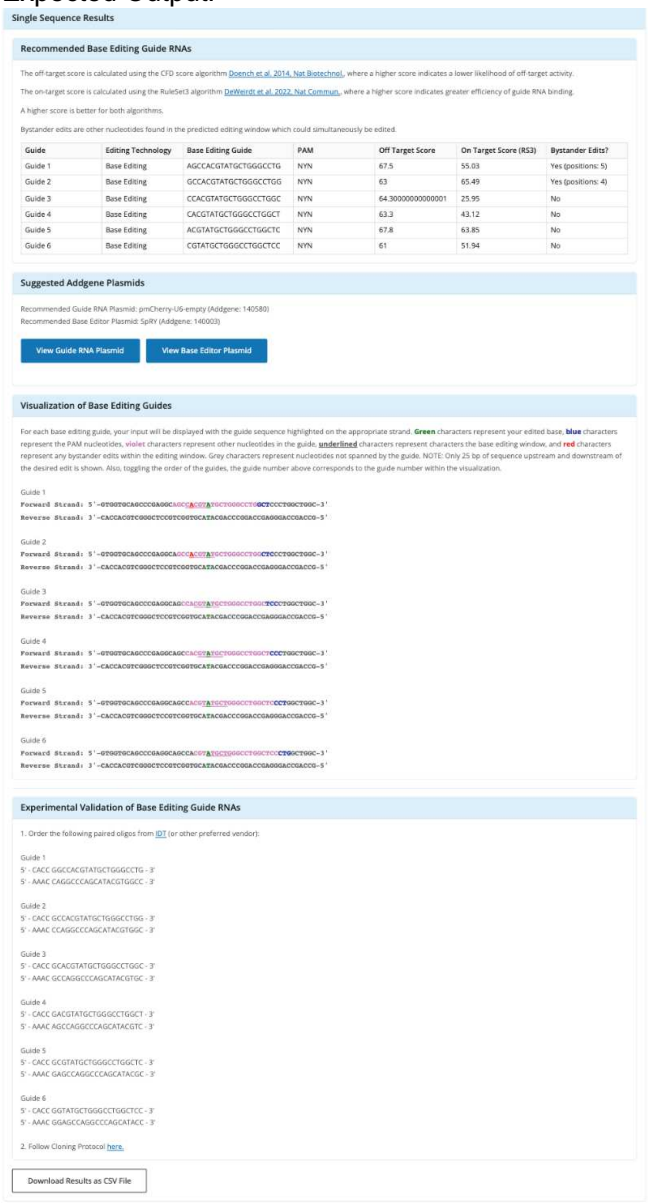
Download Results as CSV File

Example 2 – Human PKD1 Base Editing with NYN PAM site
Sequence change: Correction of *PKD1*^{C155Y} substitution, non-truncating mutation. Revert A (mutant) to G (wildtype) (38).

Original Sequence:
ACTGTGGCCTGGCGTGGCTGCCGCGATGGGCGGAGGAGCAGCAGGTGCGGGTGGTG
CAGCCCGAGGCAGCCACGT**A**TGCTGGGCCTGGCTCCCTGGCTGGCCAGCCTCTGCTT
GGCA

Desired Sequence:
ACTGTGGCCTGGCGTGGCTGCCGCGATGGGCGGAGGAGCAGCAGGTGCGGGTGGTG
CAGCCCGAGGCAGCCACGT**G**TGCTGGGCCTGGCTCCCTGGCTGGCCAGCCTCTGCTT
GGCA

Expected Output:



Single Sequence Results

Recommended Base Editing Guide RNAs

The off-target score is calculated using the CFD score algorithm Doench et al. 2014, Nat Biotechnol, where a higher score indicates a lower likelihood of off-target activity.

The on-target score is calculated using the RuleSet3 algorithm DeWeirdt et al. 2022, Nat Commun, where a higher score indicates greater efficiency of guide RNA binding.

A higher score is better for both algorithms.

Bystander edits are other nucleotides found in the predicted editing window which could simultaneously be edited.

Guide	Editing Technology	Base Editing Guide	PAM	Off Target Score	On Target Score (RS3)	Bystander Edits?
Guide 1	Base Editing	AGCCACGTATGCTGGGCCTG	NYN	67.5	55.03	Yes (positions: 5)
Guide 2	Base Editing	GCCACGTATGCTGGGCCTGG	NYN	63	65.49	Yes (positions: 4)
Guide 3	Base Editing	CCACGTATGCTGGGCCTGGC	NYN	64.30000000000001	25.95	No
Guide 4	Base Editing	CACGTATGCTGGGCCTGGCT	NYN	63.3	43.12	No
Guide 5	Base Editing	ACGTATGCTGGGCCTGGCTC	NYN	67.8	63.85	No
Guide 6	Base Editing	CGTATGCTGGGCCTGGCTCC	NYN	61	51.94	No

Suggested Addgene Plasmids

Recommended Guide RNA Plasmid: pmCherry-U6-empty (Addgene: 140580)
Recommended Base Editor Plasmid: SpRY (Addgene: 140003)

View Guide RNA Plasmid | View Base Editor Plasmid

Visualization of Base Editing Guides

For each base editing guide, your input will be displayed with the guide sequence highlighted on the appropriate strand. Green characters represent your edited base, blue characters represent the PAM nucleotides, violet characters represent other nucleotides in the guide, underlined characters represent characters the base editing window, and red characters represent any bystander edits within the editing window. Grey characters represent nucleotides not spanned by the guide. NOTE: Only 25 bp of sequence upstream and downstream of the desired edit is shown. Also, toggling the order of the guides, the guide number above corresponds to the guide number within the visualization.

Guide 1
Forward Strand: 5'-GTGGTGCAGCCCGAGGCAGCCACGTATGCTGGGCCTGGCTCCCTGGCTGGC-3'
Reverse Strand: 3'-CACCACGTCGGGCTCCGTCGGTGCATACGACCCGGACCGAGGGACCGACCG-5'

Guide 2
Forward Strand: 5'-GTGGTGCAGCCCGAGGCAGCCACGTATGCTGGGCCTGGCTCCCTGGCTGGC-3'
Reverse Strand: 3'-CACCACGTCGGGCTCCGTCGGTGCATACGACCCGGACCGAGGGACCGACCG-5'

Guide 3
Forward Strand: 5'-GTGGTGCAGCCCGAGGCAGCCACGTATGCTGGGCCTGGCTCCCTGGCTGGC-3'
Reverse Strand: 3'-CACCACGTCGGGCTCCGTCGGTGCATACGACCCGGACCGAGGGACCGACCG-5'

Guide 4
Forward Strand: 5'-GTGGTGCAGCCCGAGGCAGCCACGTATGCTGGGCCTGGCTCCCTGGCTGGC-3'
Reverse Strand: 3'-CACCACGTCGGGCTCCGTCGGTGCATACGACCCGGACCGAGGGACCGACCG-5'

Guide 5
Forward Strand: 5'-GTGGTGCAGCCCGAGGCAGCCACGTATGCTGGGCCTGGCTCCCTGGCTGGC-3'
Reverse Strand: 3'-CACCACGTCGGGCTCCGTCGGTGCATACGACCCGGACCGAGGGACCGACCG-5'

Guide 6
Forward Strand: 5'-GTGGTGCAGCCCGAGGCAGCCACGTATGCTGGGCCTGGCTCCCTGGCTGGC-3'
Reverse Strand: 3'-CACCACGTCGGGCTCCGTCGGTGCATACGACCCGGACCGAGGGACCGACCG-5'

Experimental Validation of Base Editing Guide RNAs

1. Order the following paired oligos from IDT (or other preferred vendor):

Guide 1
5' - CACC GGCCACGTATGCTGGGCCTG - 3'
5' - AAAC CAGGCCCAGCATACGTGGCC - 3'

Guide 2
5' - CACC GCCACGTATGCTGGGCCTGG - 3'
5' - AAAC CCAGGCCCAGCATACGTGGC - 3'

Guide 3
5' - CACC GCACGTATGCTGGGCCTGGC - 3'
5' - AAAC GCCAGGCCCAGCATACGTGC - 3'

Guide 4
5' - CACC GACGTATGCTGGGCCTGGCT - 3'
5' - AAAC AGCCAGGCCCAGCATACGTC - 3'

Guide 5
5' - CACC GCGTATGCTGGGCCTGGCTC - 3'
5' - AAAC GAGCCAGGCCCAGCATACGC - 3'

Guide 6
5' - CACC GGTATGCTGGGCCTGGCTCC - 3'
5' - AAAC GGAGCCAGGCCCAGCATACC - 3'

2. Follow Cloning Protocol here.

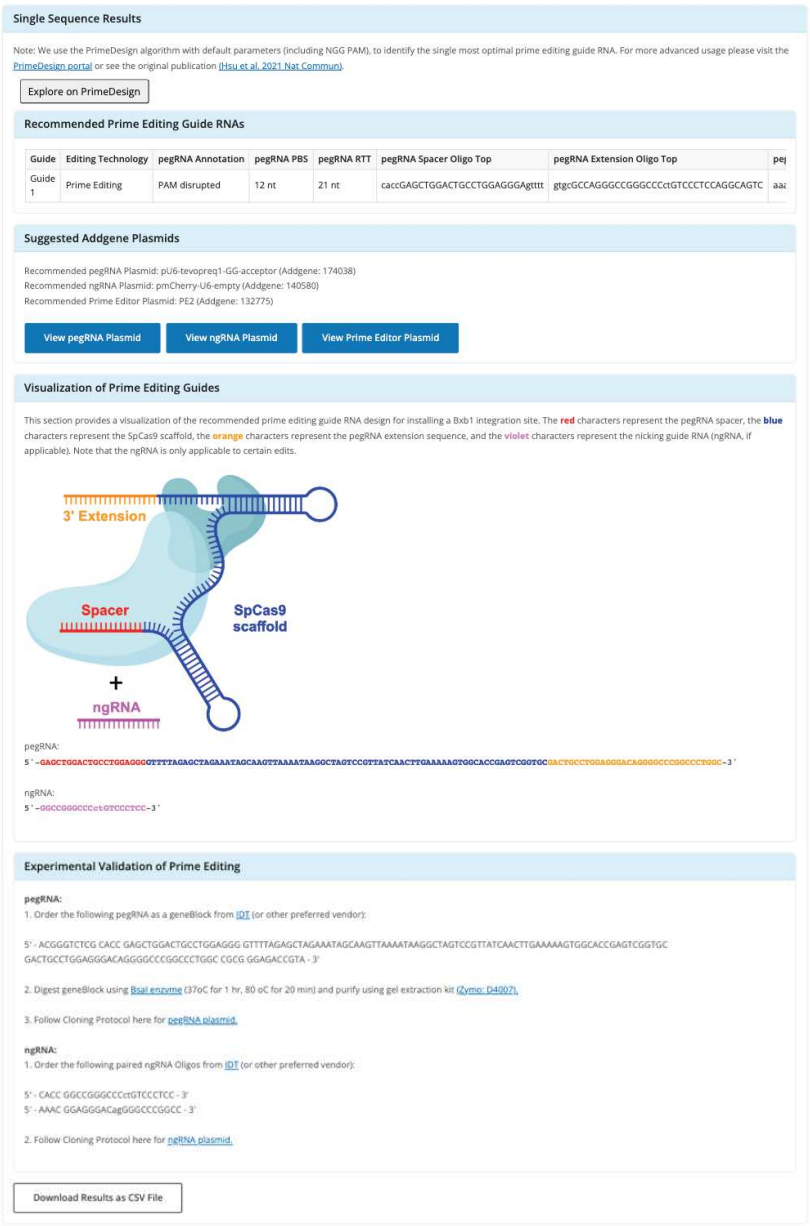
Download Results as CSV File

Example 3 – Human PKD1 Prime Editing
Sequence change: Correction of *PKD1*^{R1672fs97X} frameshift, truncating mutation. Insert AG to correct two deleted bases (39).

Original Sequence: ACCAACGTCTCCTACAGCTGGACTGCCTGGAGGGAC--GGGCCCGGCCCTGGCCGGCAGCGGCAAAGGCTTCTCGCTC

Desired Sequence: ACCAACGTCTCCTACAGCTGGACTGCCTGGAGGGACAGGGCCCGGCCCTGGCCGGCAGCGGCAAAGGCTTCTCGCTC

Expected Output:



Single Sequence Results

Note: We use the PrimeDesign algorithm with default parameters (including NGG PAM), to identify the single most optimal prime editing guide RNA. For more advanced usage please visit the PrimeDesign portal or see the original publication (Hsu et al. 2021 Nat Commun).

Explore on PrimeDesign

Recommended Prime Editing Guide RNAs

Guide	Editing Technology	pegRNA Annotation	pegRNA PBS	pegRNA RTT	pegRNA Spacer Oligo Top	pegRNA Extension Oligo Top	peg
Guide 1	Prime Editing	PAM disrupted	12 nt	21 nt	caccGAGCTGGACTGCCTGGAGGGAgttttt	gtgcGCCAGGGCCGGGCCCctGTCCCTCCAGGCAGTC	aac

Suggested Addgene Plasmids

Recommended pegRNA Plasmid: pU6-tevopreq1-GG-acceptor (Addgene: 174038)
Recommended ngRNA Plasmid: pmCherry-U6-empty (Addgene: 140580)
Recommended Prime Editor Plasmid: PE2 (Addgene: 132775)

View pegRNA Plasmid | View ngRNA Plasmid | View Prime Editor Plasmid

Visualization of Prime Editing Guides

This section provides a visualization of the recommended prime editing guide RNA design for installing a Bxb1 integration site. The red characters represent the pegRNA spacer, the blue characters represent the SpCas9 scaffold, the orange characters represent the pegRNA extension sequence, and the violet characters represent the nicking guide RNA (ngRNA, if applicable). Note that the ngRNA is only applicable to certain edits.

pegRNA:
5' -GAGCTGGACTGCCTGGAGGGGTTTTAGAGCTAGAAATAGCAAGTTAAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCGACTGCCTGGAGGGACAGGGGCCCGGCCCTGGC-3'

ngRNA:
5' -GGCCGGGCCCctGTCCCTCC-3'

Experimental Validation of Prime Editing

pegRNA:
1. Order the following pegRNA as a geneBlock from IDT (or other preferred vendor):

5' - ACGGGTCTCG CACC GAGCTGGACTGCCTGGAGGG GTTTTAGAGCTAGAAATAGCAAGTTAAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGC GACTGCCTGGAGGGACAGGGGCCCGGCCCTGGC CGCG GGAGACCGTA - 3'

2. Digest geneBlock using BsaI enzyme (37oC for 1 hr, 80 oC for 20 min) and purify using gel extraction kit (Zymo: D4007).

3. Follow Cloning Protocol here for pegRNA plasmid.

ngRNA:
1. Order the following paired ngRNA Oligos from IDT (or other preferred vendor):

5' - CACC GGCCGGGCCCctGTCCCTCC - 3'
5' - AAAC GGAGGGACagGGGCCCGGCC - 3'

2. Follow Cloning Protocol here for ngRNA plasmid.

Download Results as CSV File

Example 4 – Mouse PKD1 Base Editing
Sequence change: Correction of *Pkd1*^{T3041A} substitution, non-truncating mutation. Revert G (mutant) to A (wildtype) (40)(41).

Original Sequence:
ATGATGATGTGGAGAACTGAGGGCATTGTACCCCTGGAGGAGACCTCACCCAGCCAGG
CTGTCTGCCTCACCCGCCACCTC**GCT**GCCTTCGGTGCCAGCCTTTTTGTGCCTCCCAG
C

Desired Sequence:
ATGATGATGTGGAGAACTGAGGGCATTGTACCCCTGGAGGAGACCTCACCCAGCCAGG
CTGTCTGCCTCACCCGCCACCTC**ACT**GCCTTCGGTGCCAGCCTTTTTGTGCCTCCCAG
C

Expected Output:

Single Sequence Results

Recommended Base Editing Guide RNAs

The off-target score is calculated using the CFD score algorithm Doench et al. 2014, Nat Biotechnol., where a higher score indicates a lower likelihood of off-target activity.

The on-target score is calculated using the RuleSet3 algorithm DeWeirdt et al. 2022, Nat Commun., where a higher score indicates greater efficiency of guide RNA binding.

A higher score is better for both algorithms.

Bystander edits are other nucleotides found in the predicted editing window which could simultaneously be edited.

Guide	Editing Technology	Base Editing Guide	PAM	Off Target Score	On Target Score (RS3)	Bystander Edits?
Guide 1	Base Editing	AGGCAGCGAGGTGGCGGGTG	NGG	61.5	42.47	Yes (positions: 4)

Suggested Addgene Plasmids

Recommended Guide RNA Plasmid: pmCherry-U6-empty (Addgene: 140580)
Recommended Base Editor Plasmid: BE4max (Addgene: 112093)

View Guide RNA Plasmid | View Base Editor Plasmid

Visualization of Base Editing Guides

For each base editing guide, your input will be displayed with the guide sequence highlighted on the appropriate strand. **Green** characters represent your edited base, **blue** characters represent the PAM nucleotides, **violet** characters represent other nucleotides in the guide, **underlined** characters represent characters the base editing window, and **red** characters represent any bystander edits within the editing window. Grey characters represent nucleotides not spanned by the guide. NOTE: Only 25 bp of sequence upstream and downstream of the desired edit is shown. Also, toggling the order of the guides, the guide number above corresponds to the guide number within the visualization.

Guide 1
Forward Strand: 5'-GGCTGTCTGCCTCACCCGCCACCTCGCTGCCTTCGGTGCCAGCCTTTTTGT-3'
Reverse Strand: 3'-CCGACAGACGGAGTGGGCGGTGGAGCGACGGAAGCCACGGTCGGAAAAACA-5'

Experimental Validation of Base Editing Guide RNAs

1. Order the following paired oligos from IDT (or other preferred vendor):

Guide 1
5' - CACC GGGCAGCGAGGTGGCGGGTG - 3'
5' - AAAC CACCCGCCACCTCGCTGCCC - 3'

2. Follow Cloning Protocol here.

Download Results as CSV File

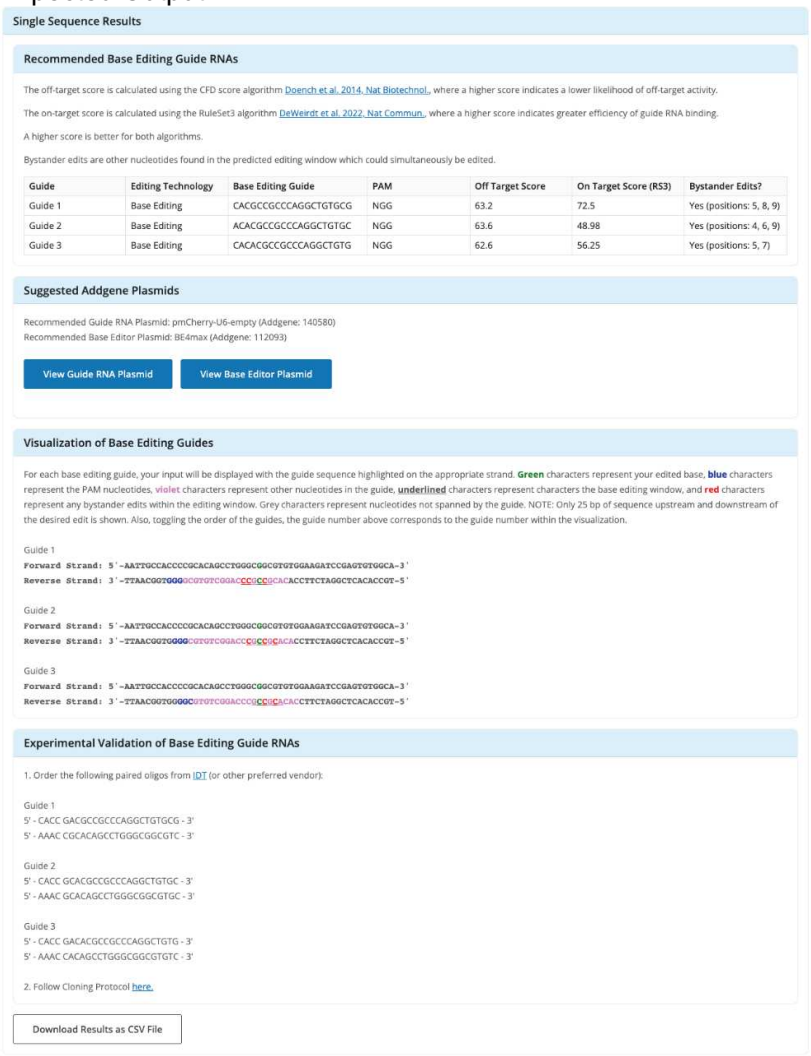
Example 5 – Mouse PKD1 Base Editing with SaCas9 PAM (NNGRRT)
Sequence change: Correction of *Pkd1*^{S3170G} substitution, non-truncating mutation. Revert G (mutant) to A (wildtype).

Original Sequence:
GAGCCTTTCATCGAAACAGCCTGGACATCTTCCAAATTGCCACCCCGCACAGCCTGGG
C**G**GCGTGTGGAAGATCCGAGTGTGGCATGATAACAAAGGTCTGGGTGATAACTGGTGT
C

Desired Sequence:
GAGCCTTTCATCGAAACAGCCTGGACATCTTCCAAATTGCCACCCCGCACAGCCTGGG
C**A**GCGTGTGGAAGATCCGAGTGTGGCATGATAACAAAGGTCTGGGTGATAACTGGTGT
C

Advanced Setting, Base Editing PAM: **NNGRRT (SaCas9)**

Expected Output:



Single Sequence Results

Recommended Base Editing Guide RNAs

The off-target score is calculated using the CFD score algorithm Doench et al. 2014, Nat Biotechnol., where a higher score indicates a lower likelihood of off-target activity.

The on-target score is calculated using the RuleSet3 algorithm DeWeirdt et al. 2022, Nat Commun., where a higher score indicates greater efficiency of guide RNA binding.

A higher score is better for both algorithms.

Bystander edits are other nucleotides found in the predicted editing window which could simultaneously be edited.

Guide	Editing Technology	Base Editing Guide	PAM	Off Target Score	On Target Score (RS3)	Bystander Edits?
Guide 1	Base Editing	CACGCCGCCCAGGCTGTGCG	NGG	63.2	72.5	Yes (positions: 5, 8, 9)
Guide 2	Base Editing	ACACGCCGCCCAGGCTGTGC	NGG	63.6	48.98	Yes (positions: 4, 6, 9)
Guide 3	Base Editing	CACACGCCGCCCAGGCTGTG	NGG	62.6	56.25	Yes (positions: 5, 7)

Suggested Addgene Plasmids

Recommended Guide RNA Plasmid: pmCherry-U6-empty (Addgene: 140580)
Recommended Base Editor Plasmid: BE4max (Addgene: 112093)

View Guide RNA Plasmid | View Base Editor Plasmid

Visualization of Base Editing Guides

For each base editing guide, your input will be displayed with the guide sequence highlighted on the appropriate strand. **Green** characters represent your edited base, **blue** characters represent the PAM nucleotides, violet characters represent other nucleotides in the guide, **underlined** characters represent characters the base editing window, and **red** characters represent any bystander edits within the editing window. Grey characters represent nucleotides not spanned by the guide. NOTE: Only 25 bp of sequence upstream and downstream of the desired edit is shown. Also, toggling the order of the guides, the guide number above corresponds to the guide number within the visualization.

Guide 1
Forward Strand: 5'–AATTGCCACCCCGCACAGCCTGGGCGGCGTGTGGAAGATCCGAGTGTGGCA–3'
Reverse Strand: 3'–TTAACGGTGGGGCGTGTCGGACCGCGCGCACACCTTCTAGGCTCACACCGT–5'

Guide 2
Forward Strand: 5'–AATTGCCACCCCGCACAGCCTGGGCGGCGTGTGGAAGATCCGAGTGTGGCA–3'
Reverse Strand: 3'–TTAACGGTGGGGCGTGTCGGACCGCGCGCACACCTTCTAGGCTCACACCGT–5'

Guide 3
Forward Strand: 5'–AATTGCCACCCCGCACAGCCTGGGCGGCGTGTGGAAGATCCGAGTGTGGCA–3'
Reverse Strand: 3'–TTAACGGTGGGGCGTGTCGGACCGCGCGCACACCTTCTAGGCTCACACCGT–5'

Experimental Validation of Base Editing Guide RNAs

1. Order the following paired oligos from IDT (or other preferred vendor):

Guide 1
5' - CACC GACGCCGCCCAGGCTGTGCG - 3'
5' - AAAC CGCACAGCCTGGGCGGCGTC - 3'

Guide 2
5' - CACC GCACGCCGCCCAGGCTGTGC - 3'
5' - AAAC GCACAGCCTGGGCGGCGTGC - 3'

Guide 3
5' - CACC GACACGCCGCCCAGGCTGTG - 3'
5' - AAAC CACAGCCTGGGCGGCGTGTC - 3'

2. Follow Cloning Protocol here.

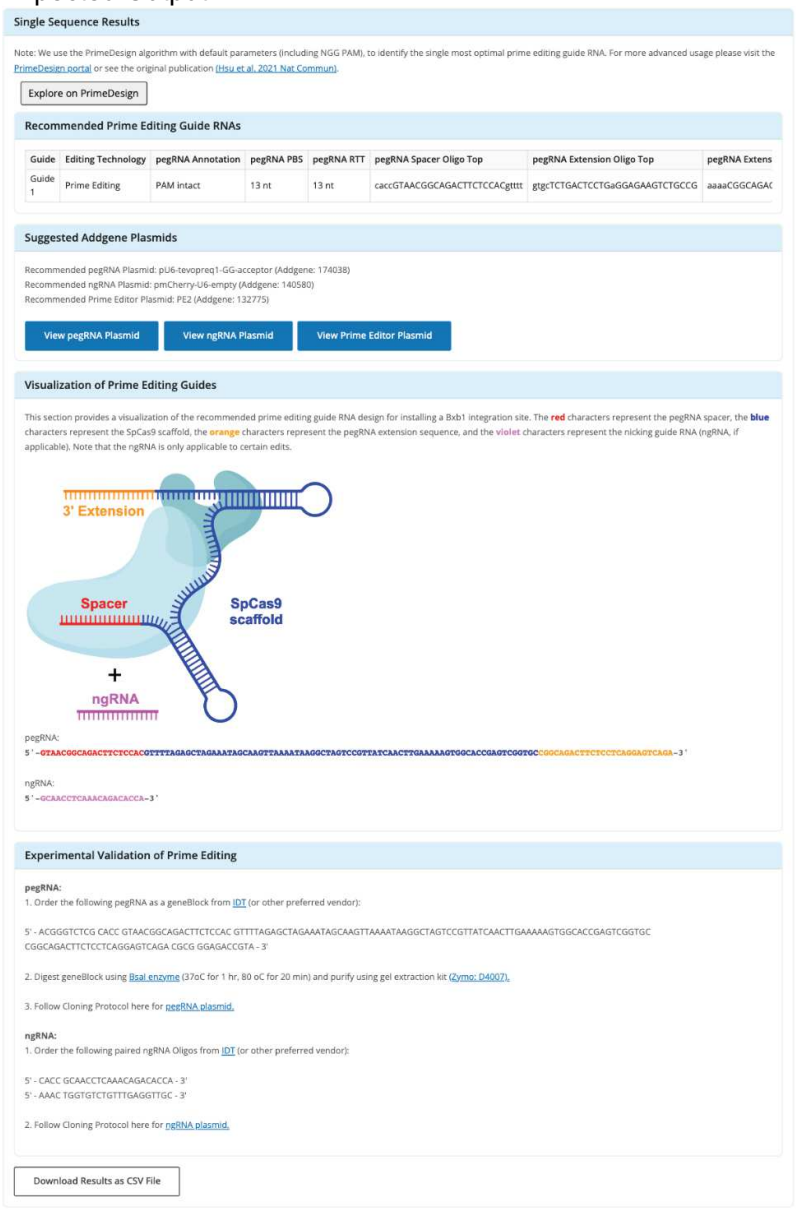
Download Results as CSV File

Example 6 – Prime Editing for Human Sickle Cell Mutation (HBB gene)
Sequence change: Correction of *HBB* p.E6V substitution mutation. Revert T (mutant) to A (wildtype) (42).

Original Sequence:
GAGCCATCTATTGCTTACATTTGCTTCTGACACAACTGTGTTCACTAGCAACCTCAAACA
GACACCATGGTGCATCTGACTCCTG**T**GGAGAAGTCTGCCGTTACTGCCCTGTGGGGC

Desired Sequence:
GAGCCATCTATTGCTTACATTTGCTTCTGACACAACTGTGTTCACTAGCAACCTCAAACA
GACACCATGGTGCATCTGACTCCTG**A**GGAGAAGTCTGCCGTTACTGCCCTGTGGGGC

Expected Output:



Example 7 – Base Editing for Hutchinson–Gilford progeria syndrome
Sequence change: Correction of human *LMNA* c.1824 C>T splicing mutation. Revert T (mutant) to C (wildtype) (43).

Original Sequence:
ACCTGCGGGCAGCCTGCCGACAAGGCATCTGCCAGCGGCTCAGGAGCCCAGGTGGGT
GGACCCATCTCCTCTGGCTCTTCTGCCTCCAGTGTC

Desired Sequence:
ACCTGCGGGCAGCCTGCCGACAAGGCATCTGCCAGCGGCTCAGGAGCCCAGGTGGG
CGGACCCATCTCCTCTGGCTCTTCTGCCTCCAGTGTC

Expected Output:

Single Sequence Results

Recommended Base Editing Guide RNAs

The off-target score is calculated using the CFD score algorithm Doench et al. 2014, Nat Biotechnol., where a higher score indicates a lower likelihood of off-target activity.

The on-target score is calculated using the RuleSet3 algorithm DeWeirdt et al. 2022, Nat Commun., where a higher score indicates greater efficiency of guide RNA binding.

A higher score is better for both algorithms.

Bystander edits are other nucleotides found in the predicted editing window which could simultaneously be edited.

Guide	Editing Technology	Base Editing Guide	PAM	Off Target Score	On Target Score (RS3)	Bystander Edits?
Guide 1	Base Editing	TCCACCCACCTGGGCTCCTG	NGN	67.19999999999999	67.66	Yes (positions: 8)
Guide 2	Base Editing	GGTCCACCCACCTGGGCTCC	NGN	63.9	34.44	No

Suggested Addgene Plasmids

Recommended Guide RNA Plasmid: pmCherry-U6-empty (Addgene: 140580)
Recommended Base Editor Plasmid: ABE8e-NG enzyme (Addgene: 138491)

View Guide RNA Plasmid | View Base Editor Plasmid

Visualization of Base Editing Guides

For each base editing guide, your input will be displayed with the guide sequence highlighted on the appropriate strand. **Green** characters represent your edited base, **blue** characters represent the PAM nucleotides, violet characters represent other nucleotides in the guide, **underlined** characters represent characters the base editing window, and **red** characters represent any bystander edits within the editing window. Grey characters represent nucleotides not spanned by the guide. NOTE: Only 25 bp of sequence upstream and downstream of the desired edit is shown. Also, toggling the order of the guides, the guide number above corresponds to the guide number within the visualization.

Guide 1
Forward Strand: 5'-CCAGCGGCTCAGGAGCCCAGGTGGGTGGACCCATCTCCTCTGGCTCTTCTG-3'
Reverse Strand: 3'-GGTCGCCGAGTCCTCGGGTCCACCCACCTGGGTAGAGGAGACCGAGAAGAC-5'

Guide 2
Forward Strand: 5'-CCAGCGGCTCAGGAGCCCAGGTGGGTGGACCCATCTCCTCTGGCTCTTCTG-3'
Reverse Strand: 3'-GGTCGCCGAGTCCTCGGGTCCACCCACCTGGGTAGAGGAGACCGAGAAGAC-5'

Experimental Validation of Base Editing Guide RNAs

1. Order the following paired oligos from IDT (or other preferred vendor):

Guide 1
5' - CACC GCCACCCACCTGGGCTCCTG - 3'
5' - AAAC CAGGAGCCCAGGTGGGTGGC - 3'

Guide 2
5' - CACC GGTCCACCCACCTGGGCTCC - 3'
5' - AAAC GGAGCCCAGGTGGGTGGACC - 3'

2. Follow Cloning Protocol here.

Download Results as CSV File

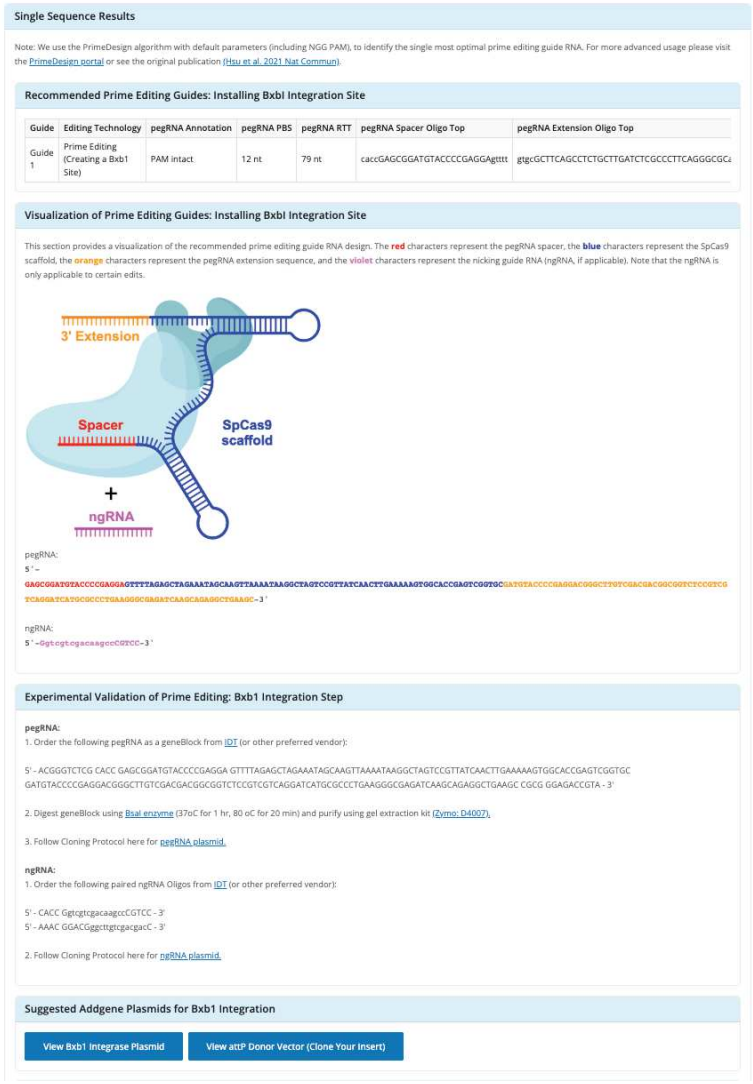
Example 8 – Integrase to insert 2×FLAG tag into Rosa26 safe harbor locus
Sequence change: Insert a dual epitope tag (2×FLAG with GS linker; 54 bp) into the mouse Rosa26 safe-harbor locus.

Original Sequence:
CAGAAGAAGACCATGGGCTGGGAGGCCTCCTCCGAGCGGATGTACCCCGAGGACG----

GCGCCCTGAAGGGCGAGATCAAGCAGAGGCTGAAGCTGAAGGACGGCGGCCACTACG
ACGCTGAGGTCAAGACCACCTACAA

Desired Sequence:
CAGAAGAAGACCATGGGCTGGGAGGCCTCCTCCGAGCGGATGTACCCCGAGGACG**GA**
CTACAAGGACGACGACGACAAGGGTAGCGACTACAAGGACGACGACGACAAGGCG
CCCTGAAGGGCGAGATCAAGCAGAGGCTGAAGCTGAAGGACGGCGGCCACTACGACG
CTGAGGTCAAGACCACCTACAA

Expected Output: Below and also see next page



Donor Plasmid Construction (Insert + attP)

Donor Plasmid Construction Protocol (Insert + attP Site):

Part 1: Prepare Your Insert

- Design your insert: include your gene or sequence of interest.
- Include regulatory elements if needed: Promoter (e.g., CMV, EF1a), Kozak sequence, and PolyA tail.
- Add flanking restriction sites (e.g., EcoRI/BamHI) or Gibson overlaps.

- Amplify your insert via high-fidelity PCR, or order it as a synthetic gBlock.
- Run the PCR product or gBlock on an agarose gel.
- Excise and purify using a gel extraction kit.

Part 2: Prepare the Vector

- Use pLenti-attP-DEST (Addgene #60561) or another attP-containing vector.
- Digest 1–2 µg of the vector using two restriction enzymes compatible with your insert (e.g., EcoRI and BamHI).
- (Optional) Treat the digested vector with alkaline phosphatase to prevent re-ligation.
- Run the digested vector on an agarose gel and purify the linearized backbone.

Part 3: Clone the Insert into the Vector

Option A: Restriction-Ligation Cloning

- Set up ligation reaction using T4 DNA ligase with a 3:1 molar ratio of insert to vector.
- Incubate at 16°C overnight or 1–2 hours at room temperature.
- Transform 5–10 µL of the ligation mix into competent E. coli (e.g., DH5α).
- Plate on LB agar with appropriate antibiotic (e.g., Ampicillin).
- Incubate overnight at 37°C.

Option B: Gibson Assembly

- Design insert and vector fragments with 20–40 bp overlaps.
- Mix insert and linearized vector with Gibson Assembly mix.
- Incubate at 50°C for 1 hour.
- Transform into competent E. coli and plate as above.

Part 4: Screen and Confirm

- Pick 4–8 colonies and perform colony PCR to screen for insert presence.
- Alternatively, miniprep and perform a diagnostic restriction digest.
- Miniprep positive clones and submit for Sanger sequencing.
- Verify correct insert sequence, orientation, and presence of attP site.

Example 9 – PRIME-Del for Fragile X Syndrome
Sequence change: Correction of FMR1 CGG repeat expansion at Xq27.3. Revert 210 CGG repeats (mutant) to 10 CGG repeats (wildtype) (44).

Original Sequence:
GGCCTCAGTCAGGCGCTCAGCTCCGTTTCGGTTTCACTTCCGGTGGAGGGCCGCCTCT
GAGCGGGCGGCGGGCCGACGGCGAGCGCGGGCGGCGGCGGTGACGGAGGCGCGCT
GCCAGGGGGCGTGCGGCAGCGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGG**CGG**
CGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGG
CGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGG
CGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGG
CGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGG
CGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGG
CGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGG
CGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGG
CGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGG
CGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGG
CGGGCTGGGCCTCGAGCGCCCGCAGCCCACCTCTCGGGGGCGGGCTCCCGGCGCTA
GCAGGGCTGAAGAGAAGATG

Desired Sequence:
GGCCTCAGTCAGGCGCTCAGCTCCGTTTCGGTTTCACTTCCGGTGGAGGGCCGCCTCT
GAGCGGGCGGCGGGCCGACGGCGAGCGCGGGCGGCGGCGGTGACGGAGGCGCGCT
GCCAGGGGGCGTGCGGCAGCGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGGCGG**-------**

GCTGGGCCTCGAGCGCCCGCAGCCCACCTCTCGGGGGCGGGCTCCCGGCGCTAGCA
GGGCTGAAGAGAAGATG

Expected Output: See next page

Single Sequence Results

Recommended PRIME-Del pegRNA sequences

Note: PRIME-Del is a genome editing method based on prime editing that facilitates the precise deletion of DNA sequences. We use the Shendure Labs's algorithm with default parameters, to identify the single most optimal prime editing guide RNAs. For more advanced usage please visit the Shendure Labs's PRIME-Del portal or see the original publication (Choi, J., et al. 2021 Nat Biotech)

Guide	Editing Technology	Prime-Del Notes	Prime-Del peg1 Spacer (RNA_1)	Prime-Del peg1 Extension (Homology+PBS_1)	Prime-Del peg1 Nick (nick_1)
Guide 3	Prime-Del	High GC at homology start; may affect priming	GTGACGGAGGCGCGCTGCCA	AGCGCCGGGAGCCCGCCCCCGAGAGGTGGGCAGCGCGCCTCCG	114
Guide 4	Prime-Del	High GC at homology start; may affect priming	GTGACGGAGGCGCGCTGCCA	CCTGCTAGCGCCGGGAGCCCGCCCCCGAGACAGCGCGCCTCCG	114
Guide 5	Prime-Del	High GC at homology start; may affect priming	GTGACGGAGGCGCGCTGCCA	CCCTGCTAGCGCCGGGAGCCCGCCCCCGAGCAGCGCGCCTCCG	114
Guide 6	Prime-Del	High GC at homology start; may affect priming	GTGACGGAGGCGCGCTGCCA	CAGCCCTGCTAGCGCCGGGAGCCCGCCCCCCAGCGCGCCTCCG	114
Guide 7	Prime-Del	High GC at homology start; may affect	TGACGGAGGCGCGCTGCCAG	GCCCGCCCCCGAGAGGTGGGCTGCGGGCGCGCAGCGCGCCTCC	115

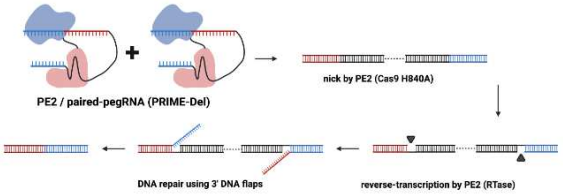
Suggested Addgene Plasmids

Recommended pegRNA Plasmid A: Prime-Del Backbone (Addgene: 172657)
Recommended pegRNA Plasmid B: Prime-Del Backbone (Addgene: 172658)
Recommended Prime Editor Plasmid: PE2 (Addgene: 199267)

View Backbone A | View Backbone B | View Prime Editor Plasmid

Visualization of PRIME-Del

This section visualizes PRIME-Del (paired prime editing). The **red** and **blue** segments simply distinguish the 3′ DNA flaps synthesized from each pegRNA's RT template—the two ends that will be joined. The dotted region marks the DNA segment programmed for deletion. Steps shown: nick by PE2 (Cas9 H840A), reverse transcription to form 3′ flaps, and DNA repair to create a precise deletion (optionally with a short, encoded insert).



PRIME-del Protocol

PRIME-del Workflow:
Components to Order:

- PE2 Prime Editor plasmid (Addgene #199267)
- pegRNA backbones (Addgene #172657, #172658)
- Custom pegRNA oligos or gene blocks (from EditABLE output)

Step 1: Clone pegRNAs into Backbone

- Gene block method (recommended for pegRNAs with PBS + RTT):
  - Order pegRNA sequences as synthetic gene blocks (IDT, Twist, etc.) with flanking BsaI sites.
  - Digest both gene block and pegRNA backbone plasmid with BsaI-HFv2 (NEB) at 37 °C for 1 hr.
  - Heat inactivate at 80 °C for 20 min.
  - Gel purify digested products (e.g., Zymo D4007 kit).
  - Ligate insert into backbone with T4 DNA ligase.
- Oligo annealing method (for short pegRNA spacers ≤24 nt):
  - Order paired oligos with EditABLE-provided CACC/AAAC overhangs.
  - Anneal oligos: 95 °C for 5 min, then cool to 25 °C.
  - Ligate into BsaI-digested backbone plasmid.
- Transformation & Screening:
  - Transform ligation into competent E. coli (DH5α).
  - Plate on LB agar + antibiotic (per plasmid resistance).
  - Screen colonies by colony PCR or direct sequencing.

Step 2: Cell Culture and Transfection

- Plate cells (e.g., HEK293T, U2OS, or target line) 24 hrs before transfection to reach ~70% confluency.
- Transfect with:
  - PE2 plasmid (Addgene #199267): 1–2 µg per well in a 6-well plate
  - Paired pegRNA plasmids: 0.5–1 µg each
- Use Lipofectamine 3000 (ThermoFisher) or electroporation for harder-to-transfect cells.
- Include a control well (PE2 + no pegRNAs).

Step 3: Expression and Editing

- Incubate cells for 72–96 hours to allow editing.
- For difficult loci, extend incubation up to 7 days.
- For stable expression, consider lentiviral delivery of pegRNAs.

Step 4: Harvest and DNA Analysis

- Extract genomic DNA from cells.
- PCR across the target locus using primers flanking the expected deletion.
- Run PCR products on agarose gel: deletion allele runs smaller than wild-type.
- Sequence PCR products (Sanger or NGS) to confirm precise deletion junction.

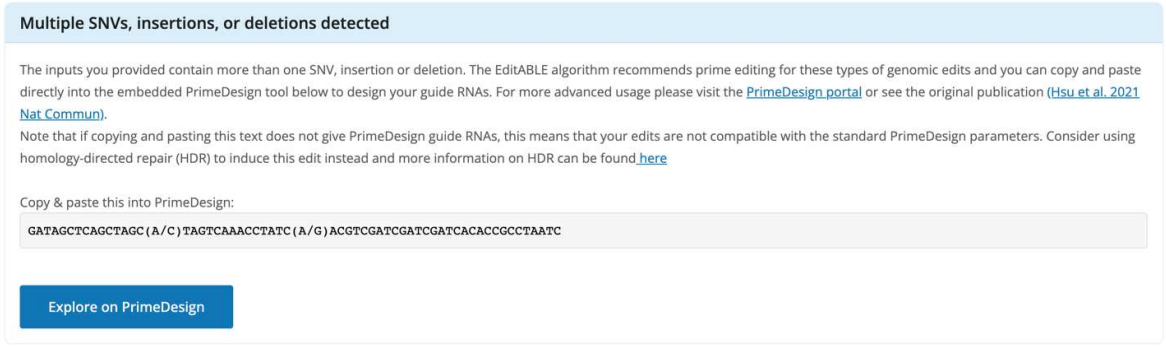
Download Results as CSV File

Example 10 – Mouse *Pkd1* Prime Editing with multiple single nucleotide variants
Sequence change: Correction of *Pkd1 R3269C* missense mutation (45)

Original Sequence:
TGGCTCTCCATATGGGACCGGCCGCCTCGTAGCCGCTTTACTTGCGTCCAGAGGGTTA
CCTGCTGTGTTCTCCTCCTCTGCCTCTTCCTGGCTGCCAATG

Desired Sequence:
TGGCTCTCCATATGGGACCGGCCGCCTCGTAGCCGCTTTACTAGAGTCCAGAGGGTTA
CCTGCTGTGTTCTCCTCCTCTGCCTCTTCCTGGCTGCCAATG

Expected Output:



NOTE S3

Full Long-Range *PKD1* PCR Protocol (Exons 13-21)

The long range protocol is based on protocols previously described with slight modifications (46),(47).

PKD1_Ex13-21_FWD Primer: TGAGTGGAGGGAGGGACGCCAAT
PKD1_Ex13-21_REV Primer: GACAGAACGGCTGAGGCTACTG

Reagent	µL/rxn
PKD1_Ex13-21_FWD primer (20 µM)	0.13
PKD1_Ex13-21_REV primer (20 µM)	0.13
10mM dNTP's (New England Biolabs: N0447S)	2.00
5X NEB LongAmp Buffer (New England Biolabs: M0323S)	5.00
DMSO (100%)	1.25
LongAmp Taq (New England Biolabs: M0323S)	1.00
Nuclease free water	14.49
TOTAL	24.00

Add 1 µL DNA template (1 – 250 ng) to make 25 µL final reaction volume.

Thermocycler Conditions:

Temp (°C)	95	95	75*	66	95	65	66	66	4
Time (min)	3:00	1:00	1:00	7:30	1:00	1:00	7:00	10:00	infinite
Cycles		x10			x25				

*Gradually decrease by 1°C per cycle.

Short-Range *PKD1* PCR Protocol on Exon 19 (*PKD1*^{Q2556X} mutation)

We dilute the long-range product 1:100 and then use 1 µL of this diluted product as the template DNA in the short-range PCR.

Reagent	µL/rxn
PKD1_Ex19_FWD primer (10 µM)	0.50
PKD1_Ex19_REV primer (10 µM)	0.50
2X OneTaq Master mix (New England Biolabs: M0482S)	12.50
Nuclease free water	10.50
TOTAL	24.00

Add 1 µL diluted long-range DNA template (1 – 250 ng) to make 25 µL final reaction volume.

Thermocycler Conditions:

Temp (°C)	95	95	49	72	72	4
Time (min)	2:00	0:15	0:15	0:30	5:00	infinite
Cycles		x35				